# Computation of Reach Sets for Dynamical Systems

Alex A. Kurzhanskiy and Pravin Varaiya*

September 28, 2008

**Abstract**

Control problems with hard bounds on the control values, restrictions on the state trajectory over a finite time horizon, and guaranteed behavior despite the disturbances, are difficult to solve using frequency based design methods. Such problems have received much attention over the past decade. In order to address them one needs to study system evolution in the time domain. The central concept that emerges in such studies is that of the *reach* set, which is the set of states that can be reached by using all possible controls. This chapter is devoted to the formulation and computation of the reach set of a linear system with disturbances.

## 1 Introduction

Traditional control theory is concerned with the design of linear feedback control with desirable asymptotic behavior, such as stability and small steady state and tracking errors, while properties of transient behavior are expressed in terms of overshoot and speed of response. External disturbances can be handled by modeling these as random processes, leading to the Linear Quadratic Gaussian (LQG) problem formulation. This theory has some limitations.

Because the feedback law is specified to be linear, it is not possible for design methods to explicitly incorporate hard bounds on the control values, e.g., the requirement that the applied force should not exceed a specified limit. Second, it is not possible to express finite time requirements, e.g., the requirement that the system state reach a pre-specified value at a pre-specified time. Third, it is not possible to demand guaranteed performance in the face of disturbances, e.g, the requirement that a certain target state be reached, no matter what the disturbance. Thus control problems with hard bounds on the control values,

*Alex A. Kurzhanskiy and Pravin Varaiya are with the Department of Electrical Engineering and Computer Sciences in the University of California, Berkeley, USA (email: akurzhan@eecs.berkeley.edu; varaiya@eecs.berkeley.edu).

restrictions on the state trajectory over a finite time horizon, and guaranteed behavior despite the disturbances, are difficult to solve using frequency based design methods.

In order to address these problems one needs to study system evolution in the time domain. The central concept that emerges in such studies is that of the *reach* set, which is the set of states that can be reached by using all possible controls. This chapter is devoted to the formulation and computation of the reach set of a linear system with disturbances. The concept of reachability was introduced in [28]; [29] shows the reach set can be computed by solving the forward Hamilton-Jacobi-Bellman-Isaacs (HJBI) partial differential equation; and the notion of backward reachability with its application to aiming at a specified target set is described in [20]. Reachability of hybrid systems is addressed in [35, 30]. Over the last decade, significant advances were made in the characterization of reach sets and their computation for linear systems. These advances are described in this chapter.

Section 2 introduces the forward and backward reach sets, the classes of open and closed loop controls, and different kinds of reach sets that are appropriate in dealing with disturbances. Although some of the discussion applies to nonlinear systems, explicit formulas for reach sets are available only for linear systems. These formulas lead to explicit algorithms for computing reach sets (of linear systems) and Section 3 critically reviews the most promising algorithms. Section 4 is devoted to a set of algorithms based on the ellipsoidal calculus. These algorithms have a lower computational complexity, greater accuracy, and can work with systems of a larger size, compared with those reviewed in Section 3. Lastly, Section 5 presents three examples to illustrate the ellipsoidal-based approach.

# 2 Basics of reachability analysis

## 2.1 Systems without disturbances

Consider a general continuous-time

$$\dot{x}(t) = f(t, x, u), \tag{2.1}$$

or discrete-time dynamical system

$$x(t+1) = f(t, x, u), \tag{2.1d}$$

wherein $t$ is time[1], $x \in \mathbf{R}^n$ is the state, $u \in \mathbf{R}^m$ is the control, and $f$ is a measurable vector function taking values in $\mathbf{R}^n$.[2] The control values $u(t, x(t))$ are restricted to a closed compact control set $\mathcal{U}(t) \subset \mathbf{R}^m$. An *open-loop* control does not depend on the state, $u = u(t)$; for a *closed-loop* control, $u = u(t, x(t))$.

[1] In discrete-time case $t$ assumes integer values.

[2] We are being general when giving the basic definitions. However, it is important to understand that for any specific *continuous-time* dynamical system it must be determined whether the solution exists and is unique, and in which class of solutions these conditions are met. Here we shall assume that function $f$ is such that the solution of the differential equation (2.1) exists and is unique in Fillipov sense. This allows the right-hand side to be discontinuous. For discrete-time systems this problem does not exist.

**Definition 2.1 (Reach set)** *The (forward) reach set* $\mathcal{X}(t, t_0, x_0)$ *at time* $t > t_0$ *from the initial position* $(t_0, x_0)$ *is the set of all states* $x(t)$ *reachable at time* $t$ *by system (2.1), or (2.1d), with* $x(t_0) = x_0$ *through all possible controls* $u(\tau, x(\tau)) \in \mathcal{U}(\tau)$*,* $t_0 \leq \tau < t$*. For a given set of initial states* $\mathcal{X}_0$*, the reach set* $\mathcal{X}(t, t_0, \mathcal{X}_0)$ *is*

$$\mathcal{X}(t, t_0, \mathcal{X}_0) = \bigcup_{x_0 \in \mathcal{X}_0} \mathcal{X}(t, t_0, x_0).$$

Here are two facts about forward reach sets.

1. $\mathcal{X}(t, t_0, \mathcal{X}_0)$ is the same for open-loop and closed-loop control.
2. $\mathcal{X}(t, t_0, \mathcal{X}_0)$ satisfies the semigroup property,

$$\mathcal{X}(t, t_0, \mathcal{X}_0) = \mathcal{X}(t, \tau, \mathcal{X}(\tau, t_0, \mathcal{X}_0)), \quad t_0 \leq \tau < t. \tag{2.2}$$

For linear systems

$$f(t, x, u) = A(t)x(t) + B(t)u, \tag{2.3}$$

with matrices $A(t)$ in $\mathbf{R}^{n \times n}$ and $B(t)$ in $\mathbf{R}^{m \times n}$. For continuous-time linear system the state transition matrix is

$$\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0), \quad \Phi(t, t) = I,$$

which for constant $A(t) \equiv A$ simplifies as

$$\Phi(t, t_0) = e^{A(t-t_0)}.$$

For discrete-time linear system the state transition matrix is

$$\Phi(t+1, t_0) = A(t)\Phi(t, t_0), \quad \Phi(t, t) = I,$$

which for constant $A(t) \equiv A$ simplifies as

$$\Phi(t, t_0) = A^{t-t_0}.$$

If the state transition matrix is invertible, $\Phi^{-1}(t, t_0) = \Phi(t_0, t)$. The transition matrix is always invertible for continuous-time and for sampled discrete-time systems. However, if for some $\tau$, $t_0 \leq \tau < t$, $A(\tau)$ is degenerate (singular), $\Phi(t, t_0) = \prod_{\tau=t_0}^{t-1} A(\tau)$, is also degenerate and cannot be inverted.

Following Cauchy's formula, the reach set $\mathcal{X}(t, t_0, \mathcal{X}_0)$ for a linear system can be expressed as

$$\mathcal{X}(t, t_0, \mathcal{X}_0) = \Phi(t, t_0)\mathcal{X}_0 \oplus \int_{t_0}^{t} \Phi(t, \tau)B(\tau)\mathcal{U}(\tau)d\tau \tag{2.4}$$

in continuous-time, and as

$$\mathcal{X}(t, t_0, \mathcal{X}_0) = \Phi(t, t_0)\mathcal{X}_0 \oplus \sum_{\tau=t_0}^{t-1} \Phi(t, \tau+1)B(\tau)\mathcal{U}(\tau) \tag{2.4d}$$

in discrete-time case.

The operation '$\oplus$' is the *geometric sum*, also known as *Minkowski sum*.[3] The geometric sum and linear (or affine) transformations preserve compactness and convexity. Hence, if the initial set $\mathcal{X}_0$ and the control sets $\mathcal{U}(\tau)$, $t_0 \leq \tau < t$, are compact and convex, so is the reach set $\mathcal{X}(t, t_0, \mathcal{X}_0)$.

**Definition 2.2 (Backward reach set)** *The backward reach set $\mathcal{Y}(t_1, t, y_1)$ for the target position $(t_1, y_1)$ is the set of all states $y(t)$ for which there exists some control $u(\tau, x(\tau)) \in \mathcal{U}(\tau)$, $t \leq \tau < t_1$, that steers system (2.1), or (2.1d) to the state $y_1$ at time $t_1$. For the target set $\mathcal{Y}_1$ at time $t_1$, the backward reach set $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ is*

$$\mathcal{Y}(t_1, t, \mathcal{Y}_1) = \bigcup_{y_1 \in \mathcal{Y}_1} \mathcal{Y}(t_1, t, y_1).$$

The backward reach set $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ is the largest *weakly invariant* set with respect to the target set $\mathcal{Y}_1$ and time values $t$ and $t_1$.[4]

**Remark.** Backward reach set can be computed for continuous-time system only if the solution of (2.1) exists for $t < t_1$; and for discrete-time system only if the right hand side of (2.1d) is invertible[5].

These two facts about the backward reach set $\mathcal{Y}$ are similar to those for forward reach sets.

1. $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ is the same for open-loop and closed-loop control.
2. $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ satisfies the semigroup property,

$$\mathcal{Y}(t_1, t, \mathcal{Y}_1) = \mathcal{Y}(\tau, t, \mathcal{Y}(t_1, \tau, \mathcal{Y}_1)), \quad t \leq \tau < t_1. \tag{2.5}$$

[3] Minkowski sum of sets $\mathcal{W}, \mathcal{Z} \subseteq \mathbf{R}^n$ is defined as $\mathcal{W} \oplus \mathcal{Z} = \{w + z \mid w \in \mathcal{W},\ z \in \mathcal{Z}\}$. Set $\mathcal{W} \oplus \mathcal{Z}$ is nonempty if and only if both, $\mathcal{W}$ and $\mathcal{Z}$ are nonempty. If $\mathcal{W}$ and $\mathcal{Z}$ are convex, set $\mathcal{W} \oplus \mathcal{Z}$ is convex.

[4] $\mathcal{M}$ is weakly invariant with respect to the target set $\mathcal{Y}_1$ and times $t_0$ and $t$, if for every state $x_0 \in \mathcal{M}$ there exists a control $u(\tau, x(\tau)) \in \mathcal{U}(\tau)$, $t_0 \leq \tau < t$, that steers the system from $x_0$ at time $t_0$ to some state in $\mathcal{Y}_1$ at time $t$. If *all* controls in $\mathcal{U}(\tau)$, $t_0 \leq \tau < t$ steer the system from every $x_0 \in \mathcal{M}$ at time $t_0$ to $\mathcal{Y}_1$ at time $t$, set $\mathcal{M}$ is said to be *strongly* invariant with respect to $\mathcal{Y}_1$, $t_0$ and $t$.

[5] There exists $f^{-1}(t, x, u)$ such that $x(t) = f^{-1}(t, x(t+1), u, v)$.

For the linear system (2.3) the backward reach set can be expressed as

$$\mathcal{Y}(t_1, t, \mathcal{Y}_1) = \Phi(t, t_1)\mathcal{Y}_1 \oplus \int_{t_1}^{t} \Phi(t, \tau)B(\tau)\mathcal{U}(\tau)d\tau \tag{2.6}$$

in the continuous-time case, and as

$$\mathcal{Y}(t_1, t, \mathcal{Y}_1) = \Phi(t, t_1)\mathcal{Y}_1 \oplus \sum_{\tau=t}^{t_1-1} -\Phi(t, \tau)B(\tau)\mathcal{U}(\tau) \tag{2.6d}$$

in discrete-time case. The last formula makes sense only for discrete-time linear systems with invertible state transition matrix. Degenerate discrete-time linear systems have unbounded backward reach sets and such sets cannot be computed with available software tools.

Just as in the case of forward reach set, the backward reach set of a linear system $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ is compact and convex if the target set $\mathcal{Y}_1$ and the control sets $\mathcal{U}(\tau)$, $t \leq \tau < t_1$, are compact and convex.

**Remark.** In the computer science literature the reach set is said to be the result of operator *post*, and the backward reach set is the result of operator *pre*. In the control literature the backward reach set is also called the *solvability set*.

## 2.2 Systems with disturbances

Consider the continuous-time dynamical system with disturbance

$$\dot{x}(t) = f(t, x, u, v), \tag{2.7}$$

or the discrete-time dynamical system with disturbance

$$x(t+1) = f(t, x, u, v), \tag{2.7d}$$

in which we also have the disturbance input $v \in \mathbf{R}^d$ with values $v(t)$ restricted to a closed compact set $\mathcal{V}(t) \subset \mathbf{R}^d$.

In the presence of disturbances the open-loop reach set (OLRS) is different from the closed-loop reach set (CLRS).

Given the initial time $t_0$, the set of initial states $\mathcal{X}_0$, and terminal time $t$, there are two types of OLRS.

**Definition 2.3 (OLRS of maxmin type)** *The maxmin open-loop reach set* $\overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0)$ *is the set of all states* $x$*, such that for any disturbance* $v(\tau) \in \mathcal{V}(\tau)$*, there exist an initial state* $x_0 \in \mathcal{X}_0$ *and a control* $u(\tau) \in \mathcal{U}(\tau)$*,* $t_0 \leq \tau < t$*, that steers system (2.7) or (2.7d) from* $x(t_0) = x_0$ *to* $x(t) = x$*.*

**Definition 2.4 (OLRS of minmax type)** *The minmax open-loop reach set* $\underline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0)$ *is the set of all states* $x$*, such that there exists a control* $u(\tau) \in \mathcal{U}(\tau)$ *that for all disturbances* $v(\tau) \in \mathcal{V}(\tau)$*,* $t_0 \leq \tau < t$*, assigns an initial state* $x_0 \in \mathcal{X}_0$ *and steers system (2.7), or (2.7d), from* $x(t_0) = x_0$ *to* $x(t) = x$*.*

In the maxmin case the control is chosen *after* knowing the disturbance over the entire time interval $[t_0, t]$, whereas in the minmax case the control is chosen *before* any knowledge of the disturbance. Consequently, the OLRS do not satisfy the semigroup property.

The terms 'maxmin' and 'minmax' come from the fact that $\overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0)$ is the subzero level set of the value function

$$\underline{V}(t, x) = \max_v \min_u \{\mathbf{dist}(x(t_0), \mathcal{X}_0) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t_0 \leq \tau < t\}, \tag{2.8}$$

i.e., $\overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0) = \{x \mid \underline{V}(t, x) \leq 0\}$, and $\underline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0)$ is the subzero level set of the value function

$$\overline{V}(t, x) = \min_u \max_v \{\mathbf{dist}(x(t_0), \mathcal{X}_0) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t_0 \leq \tau < t\}, \tag{2.9}$$

in which $\mathbf{dist}(\cdot, \cdot)$ denotes Hausdorff semidistance.[6] Since $\underline{V}(t, x) \leq \overline{V}(t, x)$, $\underline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0) \subseteq \overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0)$.

Note that maxmin and minmax OLRS imply *guarantees*: these are states that can be reached no matter what the disturbance is, whether it is known in advance (maxmin case) or not (minmax case). The OLRS may be empty.

Fixing time instant $\tau_1$, $t_0 < \tau_1 < t$, define the *piecewise maxmin open-loop reach set with one correction*,

$$\overline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0) = \overline{\mathcal{X}}_{OL}(t, \tau_1, \overline{\mathcal{X}}_{OL}(\tau_1, t_0, \mathcal{X}_0)), \tag{2.10}$$

and the *piecewise minmax open-loop reach set with one correction*,

$$\underline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0) = \underline{\mathcal{X}}_{OL}(t, \tau_1, \underline{\mathcal{X}}_{OL}(\tau_1, t_0, \mathcal{X}_0)). \tag{2.11}$$

The piecewise maxmin OLRS $\overline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0)$ is the subzero level set of the value function

$$\underline{V}^1(t, x) = \max_v \min_u \{\underline{V}(\tau_1, x(\tau_1)) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ \tau_1 \leq \tau < t\}, \tag{2.12}$$

with $V(\tau_1, x(\tau_1))$ given by (2.8), which yields

$$\underline{V}^1(t, x) \geq \underline{V}(t, x),$$

[6] Hausdorff semidistance between compact sets $\mathcal{W}, \mathcal{Z} \subseteq \mathbf{R}^n$ is defined as

$$\mathbf{dist}(\mathcal{W}, \mathcal{Z}) = \min\{\langle w - z, w - z\rangle^{1/2} \mid w \in \mathcal{W},\ z \in \mathcal{Z}\},$$

where $\langle \cdot, \cdot \rangle$ denotes inner product.

and thus,

$$\overline{\mathcal{X}}^1_{OL}(t, t_0\mathcal{X}_0) \subseteq \overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0).$$

On the other hand, the piecewise minmax OLRS $\underline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0)$ is the subzero level set of the value function

$$\overline{V}^1(t, x) = \min_u \max_v \{\overline{V}(\tau_1, x(\tau_1)) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ \tau_1 \leq \tau < t\},\ \ (2.13)$$

with $V(\tau_1, x(\tau_1))$ given by (2.9), which yields

$$\overline{V}(t, x) \geq \overline{V}^1(t, x),$$

and thus,

$$\underline{\mathcal{X}}_{OL}(t, t_0\mathcal{X}_0) \subseteq \underline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0).$$

We can now recursively define piecewise maxmin and minmax OLRS with $k$ corrections for $t_0 < \tau_1 < \cdots < \tau_k < t$. The maxmin piecewise OLRS with $k$ corrections is

$$\overline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0) = \overline{\mathcal{X}}_{OL}(t, \tau_k, \overline{\mathcal{X}}^{k-1}_{OL}(\tau_k, t_0, \mathcal{X}_0)), \tag{2.14}$$

which is the subzero level set of the corresponding value function

$$\underline{V}^k(t, x) =$$
$$\max_v \min_u \{\underline{V}^{k-1}(\tau_k, x(\tau_k)) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ \tau_k \leq \tau < t\}. \tag{2.15}$$

The minmax piecewise OLRS with $k$ corrections is

$$\underline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0) = \underline{\mathcal{X}}_{OL}(t, \tau_k, \underline{\mathcal{X}}^{k-1}_{OL}(\tau_k, t_0, \mathcal{X}_0)), \tag{2.16}$$

which is the subzero level set of the corresponding value function

$$\overline{V}^k(t, x) =$$
$$\min_u \max_v \{\overline{V}^{k-1}(\tau_k, x(\tau_k)) \mid x(t) = x,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ \tau_k \leq \tau < t\}. \tag{2.17}$$

From (2.12), (2.13), (2.15) and (2.17) it follows that

$$\underline{V}(t, x) \leq \underline{V}^1(t, x) \leq \cdots \leq \underline{V}^k(t, x) \leq \overline{V}^k(t, x) \leq \cdots \leq \overline{V}^1(t, x) \leq \overline{V}(t, x).$$

Hence,

$$\begin{aligned} &\underline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0) \subseteq \underline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0) \subseteq \cdots \subseteq \underline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0) \subseteq \\ &\overline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0) \subseteq \cdots \subseteq \overline{\mathcal{X}}^1_{OL}(t, t_0, \mathcal{X}_0) \subseteq \overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0). \end{aligned} \tag{2.18}$$

We call

$$\overline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0) = \overline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0), \quad k = \begin{cases} \infty & \text{for continuous-time system} \\ t - t_0 - 1 & \text{for discrete-time system} \end{cases} \tag{2.19}$$

the *maxmin closed-loop reach set* of system (2.7) or (2.7d) at time $t$, and we call

$$\underline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0) = \underline{\mathcal{X}}^k_{OL}(t, t_0, \mathcal{X}_0), \quad k = \begin{cases} \infty & \text{for continuous-time system} \\ t - t_0 - 1 & \text{for discrete-time system} \end{cases} \tag{2.20}$$

the *minmax closed-loop reach set* of system (2.7) or (2.7d) at time $t$.

**Definition 2.5 (CLRS of maxmin type)** *Given initial time $t_0$ and the set of initial states $\mathcal{X}_0$, the maxmin CLRS $\overline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0)$ of system (2.7) or (2.7d) at time $t > t_0$, is the set of all states $x$, for each of which and for every disturbance $v(\tau) \in \mathcal{V}(\tau)$, there exist an initial state $x_0 \in \mathcal{X}_0$ and a control $u(\tau, x(\tau)) \in \mathcal{U}(\tau)$, such that the trajectory $x(\tau|v(\tau), u(\tau, x(\tau)))$ satisfying $x(t_0) = x_0$ and*

$$\dot{x}(\tau|v(\tau), u(\tau, x(\tau))) \in f(\tau, x(\tau), u(\tau, x(\tau)), v(\tau))$$

*in the continuous-time case, or*

$$x(\tau + 1|v(\tau), u(\tau, x(\tau))) \in f(\tau, x(\tau), u(\tau, x(\tau)), v(\tau))$$

*in the discrete-time case, with $t_0 \leq \tau < t$, is such that $x(t) = x$.*

**Definition 2.6 (CLRS of minmax type)** *Given initial time $t_0$ and the set of initial states $\mathcal{X}_0$, the maxmin CLRS $\underline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0)$ of system (2.7) or (2.7d), at time $t > t_0$, is the set of all states $x$, for each of which there exists a control $u(\tau, x(\tau)) \in \mathcal{U}(\tau)$, and for every disturbance $v(\tau) \in \mathcal{V}(\tau)$ there exists an initial state $x_0 \in \mathcal{X}_0$, such that the trajectory $x(\tau, v(\tau)|u(\tau, x(\tau)))$ satisfying $x(t_0) = x_0$ and*

$$\dot{x}(\tau, v(\tau)|u(\tau, x(\tau))) \in f(\tau, x(\tau), u(\tau, x(\tau)), v(\tau))$$

*in the continuous-time case, or*

$$x(\tau + 1, v(\tau)|u(\tau, x(\tau))) \in f(\tau, x(\tau), u(\tau, x(\tau)), v(\tau))$$

*in the discrete-time case, with $t_0 \leq \tau < t$, is such that $x(t) = x$.*

By construction, both maxmin and minmax CLRS satisfy the semigroup property (2.2).

For some classes of dynamical systems and some types of constraints on initial conditions, controls and disturbances, the maxmin and minmax CLRS may coincide. This is the case for continuous-time linear systems with convex compact bounds on the initial set, controls and disturbances under the condition that the initial set $\mathcal{X}_0$ is large enough to ensure that $\mathcal{X}(t_0 + \epsilon, t_0, \mathcal{X}_0)$ is nonempty for some small $\epsilon > 0$.

Consider the linear system case,

$$f(t, x, u) = A(t)x(t) + B(t)u + G(t)v, \tag{2.21}$$

where $A(t)$ and $B(t)$ are as in (2.3), and $G(t)$ takes its values in $\mathbf{R}^d$.

The maxmin OLRS for the continuous-time linear system can be expressed through set valued integrals,

$$\begin{aligned}&\overline{\mathcal{X}}_{OL}(t, t_0, \mathcal{X}_0) = \\ &\left(\Phi(t, t_0)\mathcal{X}_0 \oplus \textstyle\int_{t_0}^t \Phi(t, \tau)B(\tau)\mathcal{U}(\tau)d\tau\right) \dot{-} \\ &\textstyle\int_{t_0}^t \Phi(t, \tau)(-G(\tau))\mathcal{V}(\tau)d\tau,\end{aligned} \tag{2.22}$$

and for discrete-time linear system through set-valued sums,

$$\overline{\mathcal{X}}_{OL}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,t_0)\mathcal{X}_0 \oplus \textstyle\sum_{\tau=t_0}^{t-1} \Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau)\Big) \dot{-} \\ \textstyle\sum_{\tau=t_0}^{t-1} \Phi(t,\tau+1)(-G(\tau))\mathcal{V}(\tau). \tag{2.22d}$$

Similarly, the minmax OLRS for the continuous-time linear system is

$$\underline{\mathcal{X}}_{OL}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,t_0)\mathcal{X}_0 \dot{-} \textstyle\int_{t_0}^{t} \Phi(t,\tau)(-G(\tau))\mathcal{V}(\tau)d\tau\Big) \oplus \\ \textstyle\int_{t_0}^{t} \Phi(t,\tau)B(\tau)\mathcal{U}(\tau)d\tau, \tag{2.23}$$

and for the discrete-time linear system it is

$$\underline{\mathcal{X}}_{OL}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,t_0)\mathcal{X}_0 \dot{-} \textstyle\sum_{\tau=t_0}^{t-1} \Phi(t,\tau+1)(-G(\tau))\mathcal{V}(\tau)\Big) \oplus \\ \textstyle\sum_{\tau=t_0}^{t-1} \Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau). \tag{2.23d}$$

The operation '$\dot{-}$' is *geometric difference*, also known as *Minkowski difference*.[7]

Now consider the piecewise OLRS with $k$ corrections. Expression (2.14) translates into

$$\overline{\mathcal{X}}_{OL}^{k}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,\tau_k)\overline{\mathcal{X}}_{OL}^{k-1}(\tau_k,t_0,\mathcal{X}_0) \oplus \textstyle\int_{\tau_k}^{t} \Phi(t,\tau)B(\tau)\mathcal{U}(\tau)d\tau\Big) \dot{-} \\ \textstyle\int_{\tau_k}^{t} \Phi(t,\tau)(-G(\tau))\mathcal{V}(\tau)d\tau, \tag{2.24}$$

in the continuous-time case, and for the discrete-time case into

$$\overline{\mathcal{X}}_{OL}^{k}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,\tau_k)\overline{\mathcal{X}}_{OL}^{k-1}(\tau_k,t_0,\mathcal{X}_0) \oplus \textstyle\sum_{\tau=\tau_k}^{t-1} \Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau)\Big) \dot{-} \\ \textstyle\sum_{\tau=\tau_k}^{t-1} \Phi(t,\tau+1)(-G(\tau))\mathcal{V}(\tau). \tag{2.24d}$$

Expression (2.16) translates into

$$\underline{\mathcal{X}}_{OL}^{k}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,\tau_k)\underline{\mathcal{X}}_{OL}^{k-1}(t,t_0,\mathcal{X}_0) \dot{-} \textstyle\int_{\tau_k}^{t} \Phi(t,\tau)(-G(\tau))\mathcal{V}(\tau)d\tau\Big) \oplus \\ \textstyle\int_{\tau_k}^{t} \Phi(t,\tau)B(\tau)\mathcal{U}(\tau)d\tau, \tag{2.25}$$

in the continuous-time case, and for the discrete-time case into

$$\underline{\mathcal{X}}_{OL}^{k}(t,t_0,\mathcal{X}_0) = \\ \Big(\Phi(t,\tau_k)\underline{\mathcal{X}}_{OL}^{k-1}(\tau_k,t_0,\mathcal{X}_0) \dot{-} \textstyle\sum_{\tau=\tau_k}^{t-1} \Phi(t,\tau+1)(-G(\tau))\mathcal{V}(\tau)\Big) \oplus \\ \textstyle\sum_{\tau=\tau_k}^{t-1} \Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau). \tag{2.25d}$$

[7] The Minkowski difference of sets $\mathcal{W},\mathcal{Z} \in \mathbf{R}^n$ is defined as $\mathcal{W}\dot{-}\mathcal{Z} = \{\xi \in \mathbf{R}^n \mid \xi \oplus \mathcal{Z} \subseteq \mathcal{W}\}$. If $\mathcal{W}$ and $\mathcal{Z}$ are convex, $\mathcal{W}\dot{-}\mathcal{Z}$ is convex if it is nonempty.

Since for any $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3 \subseteq \mathbf{R}^n$ it is true that

$$(\mathcal{W}_1 \dot{-} \mathcal{W}_2) \oplus \mathcal{W}_3 = (\mathcal{W}_1 \oplus \mathcal{W}_3) \dot{-} (\mathcal{W}_2 \oplus \mathcal{W}_3) \subseteq (\mathcal{W}_1 \oplus \mathcal{W}_3) \dot{-} \mathcal{W}_2,$$

from (2.24), (2.25) and from (2.24d), (2.25d), it is clear that (2.18) is true.

For linear systems, if the initial set $\mathcal{X}_0$, control bounds $\mathcal{U}(\tau)$ and disturbance bounds $\mathcal{V}(\tau)$, $t_0 \leq \tau < t$, are compact and convex, the CLRS $\overline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0)$ and $\underline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0)$ are compact and convex, provided they are nonempty. For continuous-time linear systems, $\overline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0) = \underline{\mathcal{X}}_{CL}(t, t_0, \mathcal{X}_0) = \mathcal{X}_{CL}(t, t_0, \mathcal{X}_0)$.

Just as for forward reach sets, the backward reach sets can be open-loop (OLBRS) or closed-loop (CLBRS).

**Definition 2.7 (OLBRS of maxmin type)** *Given the terminal time $t_1$ and target set $\mathcal{Y}_1$, the maxmin open-loop backward reach set $\overline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1)$ of system (2.7) or (2.7d) at time $t < t_1$, is the set of all $y$, such that for any disturbance $v(\tau) \in \mathcal{V}(\tau)$ there exists a terminal state $y_1 \in \mathcal{Y}_1$ and control $u(\tau) \in \mathcal{U}(\tau)$, $t \leq \tau < t_1$, which steers the system from $y(t) = y$ to $y(t_1) = y_1$.*

$\overline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1)$ is the subzero level set of the value function

$$\underline{V}_b(t, y) = \max_v \min_u \{\mathbf{dist}(y(t_1), \mathcal{Y}_1) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \leq \tau < t_1\}, \quad (2.26)$$

**Definition 2.8 (OLBRS of minmax type)** *Given the terminal time $t_1$ and target set $\mathcal{Y}_1$, the minmax open-loop backward reach set $\underline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1)$ of system (2.7) or (2.7d) at time $t < t_1$, is the set of all $y$, such that there exists a control $u(\tau) \in \mathcal{U}(\tau)$ that for all disturbances $v(\tau \in \mathcal{V}(\tau)$, $t \leq \tau < t_1$, assigns a terminal state $y_1 \in \mathcal{Y}_1$ and steers the system from $y(t) = y$ to $y(t_1) = y_1$.*

$\underline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1)$ is the subzero level set of the value function

$$\overline{V}_b(t, y) = \min_u \max_v \{\mathbf{dist}(y(t_1), \mathcal{Y}_1) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \leq \tau < t_1\}, \quad (2.27)$$

**Remark.** The backward reach set can be computed for a continuous-time system only if the solution of (2.7) exists for $t < t_1$, and for a discrete-time system only if the right hand side of (2.7d) is invertible.

Similarly to the forward reachability case, we construct piecewise OLBRS with one correction at time $\tau_1$, $t < \tau_1 < t_1$. The piecewise maxmin OLBRS with one correction is

$$\overline{\mathcal{Y}}^1_{OL}(t_1, t, \mathcal{Y}_1) = \overline{\mathcal{Y}}_{OL}(\tau_1, t, \overline{\mathcal{Y}}_{OL}(t_1, \tau_1, \mathcal{Y}_1)), \quad (2.28)$$

and it is the subzero level set of the function

$$\underline{V}_b^1(t,y) = \max_v \min_u \{\underline{V}_b(\tau_1, y(\tau_1)) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \le \tau < \tau_1\}. \quad (2.29)$$

The piecewise minmax OLBRS with one correction is

$$\underline{\mathcal{Y}}_{OL}^1(t_1, t, \mathcal{Y}_1) = \underline{\mathcal{Y}}_{OL}(\tau_1, t, \underline{\mathcal{Y}}_{OL}(t_1, \tau_1, \mathcal{Y}_1)), \quad (2.30)$$

and it is the subzero level set of the function

$$\overline{V}_b^1(t,y) = \min_u \max_v \{\overline{V}_b(\tau_1, y(\tau_1)) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \le \tau < \tau_1\}, \quad (2.31)$$

Recursively define maxmin and minmax OLBRS with $k$ corrections for $t < \tau_k < \cdots < \tau_1 < t_1$. The maxmin OLBRS with $k$ corrections is

$$\overline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1) = \overline{\mathcal{Y}}_{OL}(\tau_k, t, \overline{\mathcal{Y}}_{OL}^{k-1}(t_1, \tau_k, \mathcal{Y}_1)), \quad (2.32)$$

which is the subzero level set of function

$$\underline{V}_b^k(t,y) = \max_v \min_u \{\underline{V}_b^{k-1}(\tau_k, y(\tau_k)) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \le \tau < \tau_k\}. \quad (2.33)$$

The minmax OLBRS with $k$ corrections is

$$\underline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1) = \underline{\mathcal{Y}}_{OL}(\tau_k, t, \underline{\mathcal{Y}}_{OL}^{k-1}(t_1, \tau_k, \mathcal{Y}_1)), \quad (2.34)$$

which is the subzero level set of the function

$$\overline{V}_b^k(t,y) = \min_u \max_v \{\overline{V}_b^{k-1}(\tau_k, y(\tau_k)) \mid y(t) = y,\ u(\tau) \in \mathcal{U}(\tau),\ v(\tau) \in \mathcal{V}(\tau),\ t \le \tau < \tau_k\}, \quad (2.35)$$

From (2.29), (2.31), (2.33) and (2.35) it follows that

$$\underline{V}_b(t,y) \le \underline{V}_b^1(t,y) \le \cdots \le \underline{V}_b^k(t,y) \le \overline{V}_b^k(t,y) \le \cdots \le \overline{V}_b^1(t,y) \le \overline{V}_b(t,y).$$

Hence,

$$\underline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1) \subseteq \underline{\mathcal{Y}}_{OL}^1(t_1, t, \mathcal{Y}_1) \subseteq \cdots \subseteq \underline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1) \subseteq \overline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1) \subseteq \cdots \subseteq \overline{\mathcal{Y}}_{OL}^1(t_1, t, \mathcal{Y}_1) \subseteq \overline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1). \quad (2.36)$$

We say that

$$\overline{\mathcal{Y}}_{CL}(t_1, t, \mathcal{Y}_1) = \overline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1), \quad k = \begin{cases} \infty & \text{for continuous-time system} \\ t_1 - t - 1 & \text{for discrete-time system} \end{cases} \quad (2.37)$$

is the *maxmin closed-loop backward reach set* of system (2.7) or (2.7d) at time $t$.

We say that

$$\underline{\mathcal{Y}}_{CL}(t_1, t, \mathcal{Y}_1) = \underline{\mathcal{Y}}_{OL}^k(t_1, t, \mathcal{Y}_1), \quad k = \begin{cases} \infty & \text{for continuous-time system} \\ t_1 - t - 1 & \text{for discrete-time system} \end{cases} \quad (2.38)$$

is the *minmax closed-loop backward reach set* of system (2.7) or (2.7d) at time $t$.

**Definition 2.9 (CLBRS of maxmin type)** *Given the terminal time $t_1$ and target set $\mathcal{Y}_1$, the maxmin CLBRS $\overline{\mathcal{Y}}_{CL}(t_1, t, \mathcal{Y}_1)$ of system (2.7) or (2.7d) at time $t < t_1$, is the set of all states $y$, for each of which for every disturbance $v(\tau) \in \mathcal{V}(\tau)$ there exists terminal state $y_1 \in \mathcal{Y}_1$ and control $u(\tau, y(\tau)) \in \mathcal{U}(\tau)$ that assigns trajectory $y(\tau, |v(\tau), u(\tau, y(\tau)))$ satisfying*

$$\dot{y}(\tau|v(\tau), u(\tau, y(\tau))) \in f(\tau, y(\tau), u(\tau, y(\tau)), v(\tau))$$

*in continuous-time case, or*

$$y(\tau + 1|v(\tau), u(\tau, y(\tau))) \in f(\tau, y(\tau), u(\tau, y(\tau)), v(\tau))$$

*in discrete-time case, with $t \leq \tau < t_1$, such that $y(t) = y$ and $y(t_1) = y_1$.*

**Definition 2.10 (CLBRS of minmax type)** *Given the terminal time $t_1$ and target set $\mathcal{Y}_1$, the minmax CLBRS $\underline{\mathcal{Y}}_{CL}(t_1, t, \mathcal{Y}_1)$ of system (2.7) or (2.7d) at time $t < t_1$, is the set of all states $y$, for each of which there exists control $u(\tau, y(\tau)) \in \mathcal{U}(\tau)$ that for every disturbance $v(\tau) \in \mathcal{V}(\tau)$ assigns terminal state $y_1 \in \mathcal{Y}_1$ and trajectory $y(\tau, v(\tau)|u(\tau, y(\tau)))$ satisfying*

$$\dot{y}(\tau, v(\tau)|u(\tau, y(\tau))) \in f(\tau, y(\tau), u(\tau, y(\tau)), v(\tau))$$

*in the continuous-time case, or*

$$y(\tau + 1, v(\tau)|u(\tau, y(\tau))) \in f(\tau, y(\tau), u(\tau, y(\tau)), v(\tau))$$

*in the discrete-time case, with $t \leq \tau < t_1$, such that $y(t) = y$ and $y(t_1) = y_1$.*

Both maxmin and minmax CLBRS satisfy the semigroup property (2.5).

The maxmin OLBRS for the continuous-time linear system can be expressed through set valued integrals,

$$\begin{aligned} &\overline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1) = \\ &\left(\Phi(t, t_1)\mathcal{Y}_1 \oplus \textstyle\int_{t_1}^{t} \Phi(t, \tau)B(\tau)\mathcal{U}(\tau)d\tau\right) \dot{-} \\ &\textstyle\int_{t}^{t_1} \Phi(t, \tau)G(\tau)\mathcal{V}(\tau)d\tau, \end{aligned} \tag{2.39}$$

and for the discrete-time linear system through set-valued sums,

$$\begin{aligned} &\overline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1) = \\ &\left(\Phi(t, t_1)\mathcal{Y}_1 \oplus \textstyle\sum_{\tau=t}^{t_1-1} -\Phi(t, \tau + 1)B(\tau)\mathcal{U}(\tau)\right) \dot{-} \\ &\textstyle\sum_{\tau=t}^{t_1-1} \Phi(t, \tau + 1)G(\tau)\mathcal{V}(\tau). \end{aligned} \tag{2.39d}$$

Similarly, the minmax OLBRS for the continuous-time linear system is

$$\begin{aligned} &\underline{\mathcal{Y}}_{OL}(t_1, t, \mathcal{Y}_1) = \\ &\left(\Phi(t, t_1)\mathcal{Y}_1 \dot{-} \textstyle\int_{t}^{t_1} \Phi(t, \tau)G(\tau)\mathcal{V}(\tau)d\tau\right) \oplus \\ &\textstyle\int_{t_1}^{t} \Phi(t, \tau)B(\tau)\mathcal{U}(\tau)d\tau, \end{aligned} \tag{2.40}$$

and for the discrete-time linear system it is

$$\begin{aligned}&\underline{\mathcal{Y}}_{OL}(t_1,t,\mathcal{Y}_1) = \\ &\left(\Phi(t,t_1)\mathcal{Y}_1 \dot{-} \sum_{\tau=t}^{t_1-1}\Phi(t,\tau+1)G(\tau)\mathcal{V}(\tau)\right) \oplus \\ &\sum_{\tau=t}^{t_1-1} -\Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau).\end{aligned} \tag{2.40d}$$

Now consider piecewise OLBRS with $k$ corrections. Expression (2.32) translates into

$$\begin{aligned}&\overline{\mathcal{Y}}_{OL}^{k}(t_1,t,\mathcal{Y}_1) = \\ &\left(\Phi(t,\tau_k)\overline{\mathcal{Y}}_{OL}^{k-1}(t_1,\tau_k,\mathcal{Y}_1) \oplus \int_{\tau_k}^{t}\Phi(t,\tau)B(\tau)\mathcal{U}(\tau)d\tau\right) \dot{-} \\ &\int_{t}^{\tau_k}\Phi(t,\tau)G(\tau)\mathcal{V}(\tau)d\tau,\end{aligned} \tag{2.41}$$

in the continuous-time case, and for the discrete-time case into

$$\begin{aligned}&\overline{\mathcal{Y}}_{OL}^{k}(t_1,t,\mathcal{Y}_1) = \\ &\left(\Phi(t,\tau_k)\overline{\mathcal{Y}}_{OL}^{k-1}(t_1,\tau_k,\mathcal{Y}_1) \oplus \sum_{\tau=t}^{\tau_k-1} -\Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau)\right) \dot{-} \\ &\sum_{\tau=t}^{\tau_k-1}\Phi(t,\tau+1)G(\tau)\mathcal{V}(\tau).\end{aligned} \tag{2.41d}$$

Expression (2.34) translates into

$$\begin{aligned}&\underline{\mathcal{Y}}_{OL}^{k}(t_1,t,\mathcal{Y}_1) = \\ &\left(\Phi(t,\tau_k)\overline{\mathcal{Y}}_{OL}^{k-1}(t_1,\tau_k,\mathcal{Y}_1) \dot{-} \int_{t}^{\tau_k}\Phi(t,\tau)G(\tau)\mathcal{V}(\tau)d\tau\right) \oplus \\ &\int_{\tau_k}^{t}\Phi(t,\tau)B(\tau)\mathcal{U}(\tau)d\tau,\end{aligned} \tag{2.42}$$

in the continuous-time case, and for the discrete-time case into

$$\begin{aligned}&\underline{\mathcal{Y}}_{OL}^{k}(t_1,t,\mathcal{Y}_1) = \\ &\left(\Phi(t,\tau_k)\overline{\mathcal{Y}}_{OL}^{k-1}(t_1,\tau_k,\mathcal{Y}_1) \dot{-} \sum_{\tau=t}^{\tau_k-1}\Phi(t,\tau+1)G(\tau)\mathcal{V}(\tau)\right) \oplus \\ &\sum_{\tau=t}^{\tau_k-1} -\Phi(t,\tau+1)B(\tau)\mathcal{U}(\tau).\end{aligned} \tag{2.25d}$$

For continuous-time linear systems $\overline{\mathcal{Y}}_{CL}(t_1,t,\mathcal{Y}_1) = \underline{\mathcal{Y}}_{CL}(t_1,t,\mathcal{Y}_1) = \mathcal{Y}_{CL}(t_1,t,\mathcal{Y}_1)$ under the condition that the target set $\mathcal{Y}_1$ is large enough to ensure that $\underline{\mathcal{Y}}_{CL}(t_1,t_1-\epsilon,\mathcal{Y}_1)$ is nonempty for some small $\epsilon > 0$.

Computation of backward reach sets for discrete-time linear systems makes sense only if the state transition matrix $\Phi(t_1,t)$ is invertible.

If the target set $\mathcal{Y}_1$, control sets $\mathcal{U}(\tau)$ and disturbance sets $\mathcal{V}(\tau)$, $t \leq \tau < t_1$, are compact and convex, then CLBRS $\overline{\mathcal{Y}}_{CL}(t_1,t,\mathcal{Y}_1)$ and $\underline{\mathcal{Y}}_{CL}(t_1,t,\mathcal{Y}_1)$ are compact and convex, if they are nonempty.

## 2.3 Reachability problem

Reachability analysis is concerned with the computation of the forward $\mathcal{X}(t, t_0, \mathcal{X}_0)$ and backward $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ reach sets (the reach sets may be maxmin or minmax) in a way that can effectively meet requests like the following:

1. For the given time interval $[t_0, t]$, determine whether the system can be steered into the given target set $\mathcal{Y}_1$. In other words, is the set $\mathcal{Y}_1 \cap \bigcup_{t_0 \leq \tau \leq t} \mathcal{X}(\tau, t_0, \mathcal{X}_0)$ nonempty? And if the answer is 'yes', find a control that steers the system to the target set (or avoids the target set).[8]

2. If the target set $\mathcal{Y}_1$ is reachable from the given initial condition $\{t_0, \mathcal{X}_0\}$ in the time interval $[t_0, t]$, find the shortest time to reach $\mathcal{Y}_1$,

$$\arg\min_{\tau} \{\mathcal{X}(\tau, t_0, \mathcal{X}_0) \cap \mathcal{Y}_1 \neq \emptyset \mid t_0 \leq \tau \leq t\}.$$

3. Given the terminal time $t_1$, target set $\mathcal{Y}_1$ and time $t < t_1$ find the set of states starting at time $t$ from which the system can reach $\mathcal{Y}_1$ within time interval $[t, t_1]$. In other words, find $\bigcup_{t \leq \tau < t_1} \mathcal{Y}(t_1, \tau, \mathcal{Y}_1)$.

4. Find a closed-loop control that steers a system with disturbances to the given target set in given time.

5. Graphically display the projection of the reach set along any specified two- or three-dimensional subspace.

For linear systems, if the initial set $\mathcal{X}_0$, target set $\mathcal{Y}_1$, control bounds $\mathcal{U}(\cdot)$ and disturbance bounds $\mathcal{V}(\cdot)$ are compact and convex, so are the forward $\mathcal{X}(t, t_0, \mathcal{X}_0)$ and backward $\mathcal{Y}(t_1, t, \mathcal{Y}_1)$ reach sets. Hence reachability analysis requires the computationally effective manipulation of convex sets, and performing the set-valued operations of unions, intersections, geometric sums and differences.

Existing reach set computation tools can deal reliably only with linear systems with convex constraints. A claim that certain tool or method can be used *effectively* for nonlinear systems must be treated with caution, and the first question to ask is for what class of nonlinear systems and with what limit on the state space dimension does this tool work? Some "reachability methods for nonlinear systems" reduce to the local linearization of a system followed by the use of well-tested techniques for linear system reach set computation. Thus these approaches in fact use reachability methods for linear systems.

[8] So-called verification problems often consist in ensuring that the system is unable to reach an 'unsafe' target set within a given time interval.

# 3 Overview of computational methods and tools

Before choosing a method and a tool for reachability analysis, one should answer the following questions to specify the requirements.

1. Do you really need to compute reach sets, or it is enough to perform a safety check, e.g., to ensure that trajectories of a system never enter a given target set, or never leave a given initial set? Barrier functions or invariant sets, described in the end of this section, may be sufficient for safety checking.
2. Do you need to compute reach sets exactly, or will approximations, external and internal, be enough? Except for very specific classes of systems, exact reach set computation is not possible, and approximation techniques are required. Unless a reach set has simple structure, its exact representation is possible only for low state space dimension. Hence, the next question.
3. What is the dimension of your system? The higher is the system dimension, the rougher is the reach set approximation.

Another important quality of a computational method for reach sets is the preservation of the semigroup property. It is highly desirable that the semigroup property is maintained by the algorithm as well as by its software implementation.

## 3.1 Level set method

We start the overview of computational techniques for reach sets with the *level set* method as it points out the essence of the reachability problem, and has been used in practice for specific nonlinear systems. The idea is to solve the Hamilton-Jacobi-Bellman-Isaacs (HJBI) partial differential equation

$$\frac{\partial V}{\partial t} + \max_{u} \left\langle \frac{\partial V}{\partial x}, f(t, x, u) \right\rangle = 0,$$

with initial condition

$$V(t_0, x) = \mathbf{dist}(x, \mathcal{X}_0),$$

for $t > t_0$, and then to find the reach set $\mathcal{X}(t, t_0, \mathcal{X}_0)$ as the subzero level set of the solution $V(t, x)$,

$$\mathcal{X}(t, t_0, \mathcal{X}_0) = \{x \in \mathbf{R}^n \mid V(t, x) \leq 0\}.$$

This *forward* HJBI equation was introduced in [29]. For systems with disturbances, the HJBI equation is

$$\frac{\partial V}{\partial t} + \min_{v} \max_{u} \left\langle \frac{\partial V}{\partial x}, f(t, x, u, v) \right\rangle = 0$$

in the maxmin case, and

$$\frac{\partial \overline{V}}{\partial t} + \max_u \min_v \left\langle \frac{\partial \overline{V}}{\partial x}, f(t,x,u,v) \right\rangle = 0$$

in the minmax case.

For backward reach sets, the HJBI equation is solved in backward time,

$$\frac{\partial V_b}{\partial t} + \min_u \left\langle \frac{\partial V_b}{\partial x}, f(t,x,u) \right\rangle,$$

with boundary condition

$$V_b(t_1,x) = \mathbf{dist}(x, \mathcal{Y}_1),$$

for $t < t_1$. For systems with disturbances, the backward HJBI is

$$\frac{\partial \underline{V}_b}{\partial t} + \max_v \min_u \left\langle \frac{\partial \underline{V}_b}{\partial x}, f(t,x,u,v) \right\rangle = 0$$

in the maxmin case, and

$$\frac{\partial \overline{V}_b}{\partial t} + \min_u \max_v \left\langle \frac{\partial \overline{V}_b}{\partial x}, f(t,x,u,v) \right\rangle = 0$$

in the minmax case.

Computation of reach sets as level sets of HJBI solutions was introduced in [22, 21, 24] with special emphasis on linear systems. In [31] the authors applied the level set method to reachability analysis of hybrid systems. The level set method is implemented in the *Level Set Toolbox* [5], which uses numerical algorithms for time-dependent HJBI equations and structured grids. Work is under way to implement fast marching methods. These are effective numerical schemes that work for time-independent HJBI, but whose major restriction is the need for the control to have the same dimension as the state. Level Set Toolbox tries to compute the surface of the reach set exactly with accuracy dependent on the choice of the grid. This, plus the exponential growth of computational complexity with the system dimension makes the level set method impractical for systems with dimension larger than three.

Level Set Toolbox deals with continuous-time systems. To use the level set method in discrete-time case, one has to solve Bellman equation under the condition that the right hand side of system (2.1d) is invertible,

$$V(t+1,x) = \min_u (V(t, f^{-1}(t,x,u)))$$

with initial condition

$$V(t_0,x) = \mathbf{dist}(x, \mathcal{X}_0)$$

for $t > t_0$, and then find the forward reach set

$$\mathcal{X}(t,t_0,\mathcal{X}_0) = \{x \in \mathbf{R}^n \mid V(t,x) \leq 0\}.$$

The backward reach set is the subzero level set of the value function $V_b(t, x)$ obtained from the backward Bellman equation,

$$V_b(t-1, x) = \min_u (V_b(t, f(t-1, x, u))),$$

with boundary condition

$$V_b(t_1, x) = \mathbf{dist}(x, \mathcal{Y}_1),$$

for $t < t_1$. For systems with disturbances $\min_u$ is substituted with $\max_v \min_u$ or with $\min_u \max_v$ in both forward and backward Bellman equations, and the functions $f$ and $f^{-1}$, whose existence is required, depend on an additional parameter $v$.

Even though in discrete-time case the computation of the value function does not involve solving a PDE, it is still very burdensome, especially for nonlinear systems whose reach sets are nonconvex. Computing the distance function for such sets and minimizing it over $u$ may be difficult. Even more difficult it is to search for maxmin or minmax.

The conclusion is that although the level set method handles nonlinear systems, it is computationally costly, and the need to maintain a grid with the value function values, which must be rather dense to ensure proper accuracy, makes it practical only for low dimensional dynamical systems.

## 3.2 Quantifier elimination

For some classes of systems the reach sets can be computed symbolically using *quantifier elimination*. Quantifier elimination is the removal of all quantifiers (the universal quantifier $\forall$ and the existential quantifier $\exists$) from a quantified system. Each quantified formula is substituted with quantifier-free expression with operations $+$, $\times$, $=$ and $<$. For example, consider the discrete-time linear system (2.1d), (2.3), with $A(t) = A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B(t) = B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. For initial conditions $x_0 \in \{x \in \mathbf{R}^2 \mid \|x\|_\infty \leq 1\}$ and controls $u(t) \in \{u \in \mathbf{R} \mid -1 \leq u \leq 1\}$, the reach set for $t \geq 0$ is given by the quantified formula

$$\Big\{x \in \mathbf{R}^2 \mid \exists x_0, \;\; \exists t \geq 0, \;\; \exists u(\tau), \; 0 \leq \tau < t : \\ x = A^t x_0 + \sum_{\tau=0}^{t-1} A^{t-\tau-1} B u(\tau) \Big\},$$

which is equivalent to the quantifier-free expression

$$-1 \leq [1 \;\; 0]x \leq 1 \;\wedge\; -1 \leq [0 \;\; 1]x \leq 1.$$

It is proved in [27] that if $A$ is constant and nilpotent or is diagonalizable with rational real or purely imaginary eigenvalues, the quantifier elimination package returns a quantifier free formula describing the reach set. This class of systems is evidently rather limited.

Requiem [10] is a Mathematica notebook which, given a linear system, the set of initial conditions and control bounds, symbolically computes the exact reach set, using the experimental quantifier elimination package.

## 3.3 Polytope method

**Definition 3.1 (Hyperplane)** *The hyperplane $H(c,\gamma)$ in $\mathbf{R}^n$ is the set*

$$H = \{x \in \mathbf{R}^n \mid \langle c, x\rangle = \gamma\},$$

*with fixed $c \in \mathbf{R}^n$ and $\gamma \in \mathbf{R}$.*

A hyperplane defines two (closed) *halfspaces*,

$$\mathbf{S}_1 = \{x \in \mathbf{R}^n \mid \langle c, x\rangle \leq \gamma\},$$

and

$$\mathbf{S}_2 = \{x \in \mathbf{R}^n \mid \langle c, x\rangle \geq \gamma\}.$$

**Definition 3.2 (Polytope)** *The polytope $P(C,g)$ is the intersection of a finite number of closed halfspaces:*

$$P = \{x \in \mathbf{R}^n \mid Cx \leq g\},$$

*with fixed $C = [c_1 \ \cdots \ c_m]^T \in \mathbf{R}^{m\times n}$ and $g = [\gamma_1 \ \cdots \ \gamma_m]^T \in \mathbf{R}^m$.*

For linear discrete-time systems (2.7d), (2.21), with $\mathcal{X}_0$, $\mathcal{U}(t)$ and $\mathcal{V}(t)$, $t \geq t_0$, being polytopes in $\mathbf{R}^n$, $\mathbf{R}^m$ and $\mathbf{R}^d$ respectively, the reach sets $\overline{\mathcal{X}}_{CL}(t,t_0,\mathcal{X}_0)$ and $\underline{\mathcal{X}}_{CL}(t,t_0,\mathcal{X}_0)$ are also polytopes, because the polytope structure is closed under the operations of affine transformation, geometric sum and geometric difference. (For continuous-time systems the reach sets need not be polytopes.) Starting with initial condition

$$\overline{\mathcal{X}}_{CL}(t_0,t_0,\mathcal{X}_0) = \underline{\mathcal{X}}_{CL}(t_0,t_0,\mathcal{X}_0) = \mathcal{X}_0,$$

for time step $t > t_0$ these reach sets are computed *exactly* as

$$\overline{\mathcal{X}}_{CL}(t,t_0,\mathcal{X}_0) = A(t-1)\mathcal{X}(t-1,t_0,\mathcal{X}_0) \oplus B(t-1)\mathcal{U}(t-1) \dot{-} G(t-1)\mathcal{V}(t-1),$$

and

$$\underline{\mathcal{X}}_{CL}(t,t_0,\mathcal{X}_0) = A(t-1)\mathcal{X}(t-1,t_0,\mathcal{X}_0) \dot{-} G(t-1)\mathcal{V}(t-1) \oplus B(t-1)\mathcal{U}(t-1).$$

A similar procedure works for backward reach sets if the matrices $A(t)$, $t < t_1$ are nondegenerate.

As we see, the polytope method consists in sequential computation of affine transformations of polytopes, geometric sum of two polytopes, and geometric difference of two polytopes (or, geometric difference first, then geometric sum for minmax CLRS). In the absence of disturbances the term $G(t-1)\mathcal{V}(t-1)$ vanishes and no geometric difference operation is performed. Each operation of geometric sum or geometric difference for two polytopes consists in finding the vertices of the resulting polytope and calculating its convex hull.

This method is implemented in the *Multi-Parametric Toolbox* (MPT) for MATLAB [26, 7]. Among its advantages are its simplicity, the fact that the reach sets are computed exactly, and that it is easy to compute the distance between two polytopes and to check whether two or more polytopes intersect, or whether a polytope intersects a hyperplane or a halfspace.

However, the polytope method has its limitations. The convex hull algorithm employed by MPT is based on the Double Description method [32] and implemented in the *CDD/CDD+* package [1]. Its complexity is $K^n$, where $K$ is the number of vertices and $n$ is the state space dimension. Hence, the use of MPT in general, is practical only for low dimensional systems, or for systems with very special structure of matrices $A$, $B$ and $G$ that ensure that the number of polytope vertices does not grow too much with each time step. But even in low dimensional systems the number of vertices in the reach set polytope can grow very large with the number of time steps. For example, consider the discrete-time linear time-invariant system with $A(t) = A = \begin{bmatrix} \cos 1 & -\sin 1 \\ \sin 1 & \cos 1 \end{bmatrix}$, $B(t) = I$, $G(t) = 0$ $u_k \in \{u \in \mathbf{R}^2 \mid \|u\|_\infty \leq 1\}$, and $x_0 \in \{x \in \mathbf{R}^2 \mid \|x\|_\infty \leq 1\}$. Starting with a rectangular initial set, the number of vertices of the reach set polytope is $4t + 4$ at the $t$-th step.

### 3.4 d/dt

In *d/dt* [3], the reach set of a discrete-time linear system is approximated by unions of rectangular polytopes [13]. The algorithm works as follows. First, given the set of initial conditions $\mathcal{X}_0$ defined as a polytope at time $t_0$, the evolution in time of the polytope's extreme points is computed $\mathcal{X}(\tau_1, t_0, \mathcal{X}_0)$. Second, the algorithm computes the convex hull of vertices of both, the initial polytope $\mathcal{X}_0$ and $\mathcal{X}(\tau_1, t_0, \mathcal{X}_0)$. The resulting polytope is then bloated (magnified) to include $\bigcup_{t_0 \leq \tau \leq \tau_1} \mathcal{X}(\tau, t_0, \mathcal{X}_0)$. Finally, this overapproximating polytope is in its turn overapproximated by the union of rectangles. The same procedure is repeated for the next time interval $[\tau_1, \tau_2]$, and the union of both rectangular approximations is taken, and so on.

Rectangular polytopes are easy to represent and the number of facets grows linearly with dimension, but a large number of rectangles must be used to assure the approximation is not overly conservative. Besides, the important part of this method is again the convex hull calculation whose implementation relies on the same CDD/CDD+ library. This limits the dimension of the system and time interval for which it is feasible to calculate the reach set.

*d/dt* is implemented in C.

### 3.5 Zonotope method

Polytopes can give arbitrarily close approximations to any convex set, but the number of vertices can grow prohibitively large and, as shown in [14], the computation of a polytope by its convex hull becomes intractable for large number of vertices in high dimensions. Symmetric polytopes, called zonotopes [11], could be a solution.

**Definition 3.3 (Zonotope)** *A zonotope is a special class of polytopes of the form,*

$$Z = \left\{ x \in \mathbf{R}^n \mid x = c + \sum_{i=1}^{k} \alpha_i g_i, \quad -1 \leq \alpha_i \leq 1 \right\},$$

*wherein $c$ and $g_1, ..., g_k$ are vectors in $\mathbf{R}^n$.*

Thus, a zonotope $Z$ is compactly represented by its center $c$ and generator vectors $g_1, ..., g_k$. The value $k/n$ is called the order of the zonotope.

The zonotope method for external approximation of reach sets of discrete-time linear systems was introduced in [17], implemented in the *MATISSE* package for MATLAB [6], and further discussed in [18]. In [18] the authors introduce computational tricks that work only for *time-invariant* linear systems. The advantage of zonotopes is that they are closed under the operations of affine transformation and geometric sum, hence, the reach set of a discrete-time linear system (2.1d), (2.3), with $\mathcal{X}_0$ and $\mathcal{U}(t)$, $t \geq t_0$, being zonotopes, is also zonotope. Similar properties hold for the backward reach set.

The problem with using zonotopes is that with every time step the order of the approximating zonotope increases by $k/n$. This difficulty can be averted by limiting the number of generator vectors, and overapproximating zonotopes whose number of generator vectors exceeds this limit by lower order zonotopes. This may affect the accuracy of the reach set approximation and potentially destroy the semigroup property that is inherently present in the zonotope method.

Further limitations of zonotopes are that geometric difference of two zonotopes, intersections of zonotopes or zonotopes with hyperplanes or halfspaces, are not zonotopes. That presents a difficulty for the computation of reach sets for systems with disturbances and hybrid systems. Effective zonotope approximation algorithms for the geometric difference and intersections are needed. Currently, MATISSE does not provide a zonotope library in which these operations are implemented.

### 3.6 CheckMate

CheckMate [2] is a MATLAB toolbox that can evaluate specifications for trajectories starting from the set of initial (continuous) states corresponding to the parameter values at the vertices of the parameter set. This provides preliminary insight into whether the specifications will be true for all parameter values. The method of oriented rectangluar polytopes for external approximation of reach sets is introduced in [36]. The basic idea is to construct an oriented rectangular hull of the reach set for every time step, whose orientation is determined by the singular value decomposition of the sample covariance matrix for the states reachable from the vertices of the initial polytope. The limitation of CheckMate and the method of oriented rectangles is that only autonomous (i.e., there is no control) systems are allowed, and only an external approximation of the reach set is provided.

Currently, the development of CheckMate is discontinued. Therefore, we refer the reader to *PHAVer* [8], the newly developed verification tool that uses *Parma Polyhedra Library* (PPL) [9] for its polyhedral computations.

### 3.7 Ellipsoidal method

All the geometric methods for reach set computation described above, namely polytopes, zonotopes, rectangular hulls and oriented rectangles employ the notion of time step. At every time step a certain algorithm runs producing a new reach set for that time step. This can work only for discrete-time systems. The ellipsoidal method offers a different approach that works for continuous- and discrete-time linear systems with disturbances, with ellipsoidal constraints on the initial or target set, controls and disturbances.

**Definition 3.4 (Ellipsoid)** *The ellipsoid $\mathcal{E}(q, Q)$ in $\mathbf{R}^n$ with center $q$ and shape matrix $Q$ is the set*

$$\mathcal{E}(q, Q) = \left\{ x \in \mathbf{R}^n \mid \langle (x-q), Q^{-1}(x-q) \rangle \leq 1 \right\},$$

*wherein $Q$ is positive definite ($Q = Q^T$ and $\langle x, Qx \rangle > 0$ for all nonzero $x \in \mathbf{R}^n$).*

**Definition 3.5 (Support function)** *The support function of a set $\mathcal{X} \subseteq \mathbf{R}^n$ is*

$$\rho(l \mid \mathcal{X}) = \sup_{x \in \mathcal{X}} \langle l, x \rangle.$$

In particular, the support function of an ellipsoid is

$$\rho(l \mid \mathcal{E}(q, Q)) = \langle l, q \rangle + \langle l, Ql \rangle^{1/2}. \tag{3.1}$$

We say that the ellipsoid $\mathcal{E}$ *tightly overapproximates* a given convex set $\mathcal{X}$ if there exist $l \in \mathbf{R}^n$ such that

$$\rho(\pm l \mid \mathcal{E}) = \rho(\pm l \mid \mathcal{X}) \quad \text{and} \quad \mathcal{X} \subseteq \mathcal{E}.$$

We say that ellipsoid $\mathcal{E}$ *tightly underapproximates* given convex set $\mathcal{X}$ if there exist $l \in \mathbf{R}^n$ such that

$$\rho(\pm l \mid \mathcal{E}) = \rho(\pm l \mid \mathcal{X}) \quad \text{and} \quad \mathcal{E} \subseteq \mathcal{X}.$$

The equality $\rho(\pm l \mid \mathcal{E}) = \rho(\pm l \mid \mathcal{X})$ means that the boundaries of $\mathcal{E}$ and $\mathcal{X}$ touch in directions $l$ and $-l$.

In [23] the authors introduce parametrized families of external and internal ellipsoids that tightly overapproximate and underapproximate the reach set and derive the differential equations that govern the evolution in time of the center and the shape matrices of these ellipsoids. The reach set is represented as the intersection of tight external and as the union of tight internal ellipsoids. In [25] this result is extended to the discrete-time case with special emphasis on systems with degenerate matrices $A(t)$. In the next Section we present the equations that describe ellipsoidal overapproximation and underapproximation of reach sets.

The ellipsoidal method provides the following benefits:

- Approximating the reach set of an $n$-dimensional discrete-time linear system by $L$ ellipsoids over $t$ time steps requires $t[L(8n^3 + 4n^2 + 2n) + 2n^2]$ scalar multiplications. The computational complexity grows polynomially with the system dimension, in contrast with the exponential growth of the polytope method complexity.
- It is possible to refine the reach set approximation as much as needed by adding more ellipsoids to the parameterized family. Theoretically, it is possible to exactly represent the reach set of linear system through both external and internal ellipsoids.
- It is possible to single out individual external and internal approximating ellipsoids that are optimal for a given criterion (e.g., trace, volume, diameter), or a combination of such criteria.
- For systems with no disturbance, there are simple analytical expressions for control sequences that steer the state to a desired target.

*Ellipsoidal Toolbox* (ET) for MATLAB [4] implements the reach set computations described here.

### 3.8 Parallelotope method

The parallelotope[9] method [19] employs the idea of the ellipsoidal method to compute the reach sets of linear systems. The reach set is represented as the intersection of a parametrized family of tight external, and the union of a parametrized family of tight internal parallelotopes. The evolution equations for the centers and orientation matrices of both external and internal parallelotopes are provided. This method also finds controls that can drive the system to the boundary points of the reach set, similarly to [37] and [23]. The computation to solve the evolution equations for tight approximating parallelotopes, however, is more involved than the one for ellipsoids, and in the case of discrete-time systems this method does not deal with singular state transition matrices.

### 3.9 Other methods

As was mentioned above, for certain verification problems computation of reach sets can be avoided. For example, it may be enough ensure that for given set of initial conditions $\mathcal{X}_0$,

[9] Parallelotope is a zonotope with $n$ generator vectors in $\mathbf{R}^n$.

the trajectories of system (2.1) never enter a given target set $\mathcal{Y}_1$. In this case, the method of *barrier certificates* [34] may help. The idea as well as the main difficulty is to find a Liapunov-like function $C(x)$ such that

1. $C(x) > 0$ in $\mathcal{Y}_1$;
2. $C(x) \leq 0$ in $\mathcal{X}_0$; and
3. $\langle D_x C(x), f(t,x,u)\rangle \leq 0$ where $C(x) = 0$.

If such a function exists, system (2.1) is 'safe' with respect to the initial set $\mathcal{X}_0$ and the target set $\mathcal{Y}_1$, i.e., system trajectories emanating from $\mathcal{X}_0$ never reach $\mathcal{Y}_1$.

Another example for which reach sets need not be computed exactly occurs when it is possible to ensure that for given initial set $\mathcal{X}_0$ there exist system trajectories that never leave $\mathcal{X}_0$. The set $\mathcal{X}_0$ is said to be *invariant* with respect to those trajectories.

In [12] the authors show that for certain classes of discrete-time dynamical systems with disturbances (2.7d) and certain initial sets $\mathcal{X}_0$, convex constraints on controls and disturbances, for every disturbance there exist closed-loop control strategies that keep the state of the system inside $\mathcal{X}_0$.

For more information about invariant sets, we refer the reader to the survey paper [15] and references therein.

# 4 Ellipsoidal method

Consider a continuous-time linear system

$$\dot{x}(t) = A(t)x(t) + B(t)u + G(t)v, \tag{4.1}$$

in which $x \in \mathbf{R}^n$ is the state, $u \in \mathbf{R}^m$ is the control and $v \in \mathbf{R}^d$ is the disturbance. $A(t)$, $B(t)$ and $G(t)$ are continuous and take their values in $\mathbf{R}^{n\times n}$, $\mathbf{R}^{n\times m}$ and $\mathbf{R}^{n\times d}$ respectively. Control $u(t,x(t))$ and disturbance $v(t)$ are measurable functions restricted by ellipsoidal constraints: $u(t,x(t)) \in \mathcal{E}(p(t),P(t))$ and $v(t) \in \mathcal{E}(q(t),Q(t))$. The set of initial states at initial time $t_0$ is assumed to be the ellipsoid $\mathcal{E}(x_0,X_0)$.

The reach sets for systems with disturbances computed by the Ellipsoidal Toolbox are CLRS. Henceforth, when describing backward reachability, reach sets refer to CLRS or CLBRS. Recall that for continuous-time linear systems maxmin and minmax CLRS coincide, and the same is true for maxmin and minmax CLBRS.

If the matrix $Q(\cdot) = 0$, the system (4.1) becomes an ordinary affine system with known $v(\cdot) = q(\cdot)$. If $G(\cdot) = 0$, the system becomes linear. For these two cases ($Q(\cdot) = 0$ or

$G(\cdot) = 0$) the reach set is as given in Definition 2.1, and so the reach set will be denoted as $\mathcal{X}_{CL}(t, t_0, \mathcal{E}(x_0, X_0)) = \mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0))$.

The reach set $\mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0))$ is a symmetric compact convex set, whose center evolves in time according to

$$\dot{x}_c(t) = A(t)x_c(t) + B(t)p(t) + G(t)q(t), \quad x_c(t_0) = x_0. \tag{4.2}$$

Fix a vector $l_0 \in \mathbf{R}^n$, and consider the solution $l(t)$ of the adjoint equation

$$\dot{l}(t) = -A^T(t)l(t), \quad l(t_0) = l_0, \tag{4.3}$$

which is equivalent to

$$l(t) = \Phi^T(t_0, t)l_0.$$

If the reach set $\mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0))$ is nonempty, there exist tight external and tight internal approximating ellipsoids $\mathcal{E}(x_c(t), X_l^+(t))$ and $\mathcal{E}(x_c(t), X_l^-(t))$, respectively, such that

$$\mathcal{E}(x_c(t), X_l^-(t)) \subseteq \mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0)) \subseteq \mathcal{E}(x_c(t), X_l^+(t)), \tag{4.4}$$

and

$$\rho(l(t) \mid \mathcal{E}(x_c(t), X_l^-(t))) = \rho(l(t) \mid \mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0))) = \rho(l(t) \mid \mathcal{E}(x_c(t), X_l^+(t))). \tag{4.5}$$

The equation for the shape matrix of the external ellipsoid is

$$\begin{aligned} \dot{X}_l^+(t) &= A(t)X_l^+(t) + X_l^+(t)A^T(t) + \\ &\quad \pi_l(t)X_l^+(t) + \frac{1}{\pi_l(t)}B(t)P(t)B^T(t) - \\ &\quad (X_l^+(t))^{1/2}S_l(t)(G(t)Q(t)G^T(t))^{1/2} - \\ &\quad (G(t)Q(t)G^T(t))^{1/2}S_l^T(t)(X_l^+(t))^{1/2}, \qquad (4.6) \\ X_l^+(t_0) &= X_0, \qquad (4.7) \end{aligned}$$

in which

$$\pi_l(t) = \frac{\langle l(t), B(t)P(t)B^T(t)l(t)\rangle^{1/2}}{\langle l(t), X_l^+(t)l(t)\rangle^{1/2}},$$

and the orthogonal matrix $S_l(t)$ ($S_l(t)S_l^T(t) = I$) is determined by the equation

$$S_l(t)(G(t)Q(t)G^T(t))^{1/2}l(t) = \frac{\langle l(t), G(t)Q(t)G^T(t)l(t)\rangle^{1/2}}{\langle l(t), X_l^+(t)l(t)\rangle^{1/2}}(X_l^+(t))^{1/2}l(t).$$

In the presence of disturbance, if the reach set is empty, the matrix $X_l^+(t)$ becomes sign indefinite. For a system without disturbance, the terms containing $G(t)$ and $Q(t)$ vanish from the equation (4.6).

The equation for the shape matrix of the internal ellipsoid is

$$
\begin{aligned}
\dot{X}_l^-(t) &= A(t)X_l^-(t) + X_l^-(t)A^T(t) + \\
&\quad (X_l^-(t))^{1/2}T_l(t)(B(t)P(t)B^T(t))^{1/2} + \\
&\quad (B(t)P(t)B^T(t))^{1/2}T_l^T(t)(X_l^-(t))^{1/2} - \\
&\quad \eta_l(t)X_l^-(t) - \frac{1}{\eta_l(t)}G(t)Q(t)G^T(t), \qquad (4.8)\\
X_l^-(t_0) &= X_0, \qquad (4.9)
\end{aligned}
$$

in which

$$\eta_l(t) = \frac{\langle l(t), G(t)Q(t)G^T(t)l(t)\rangle^{1/2}}{\langle l(t), X_l^+(t)l(t)\rangle^{1/2}},$$

and the orthogonal matrix $T_l(t)$ is determined by the equation

$$T_l(t)(B(t)P(t)B^T(t))^{1/2}l(t) = \frac{\langle l(t), B(t)P(t)B^T(t)l(t)\rangle^{1/2}}{\langle l(t), X_l^-(t)l(t)\rangle^{1/2}}(X_l^-(t))^{1/2}l(t).$$

Similarly to the external case, the terms containing $G(t)$ and $Q(t)$ vanish from the equation (4.8) for a system without disturbance.

The point where the external and internal ellipsoids touch the boundary of the reach set is given by

$$x_l^*(t) = x_c(t) + \frac{X_l^+(t)l(t)}{\langle l(t), X_l^+(t)l(t)\rangle^{1/2}}.$$

The boundary points $x_l^*(t)$ form trajectories, which we call *extremal trajectories*. Due to the nonsingular nature of the state transition matrix $\Phi(t, t_0)$, every boundary point of the reach set belongs to an extremal trajectory. To follow an extremal trajectory specified by parameter $l_0$, the system has to start at time $t_0$ at initial state

$$x_l^0 = x_0 + \frac{X_0 l_0}{\langle l_0, X_0 l_0\rangle^{1/2}}. \qquad (4.10)$$

In the absence of disturbances, the open-loop control

$$u_l(t) = p(t) + \frac{P(t)B^T(t)l(t)}{\langle l(t), B(t)P(t)B^T(t)l(t)\rangle^{1/2}}. \qquad (4.11)$$

steers the system along the extremal trajectory defined by the vector $l_0$. When a disturbance is present, this control keeps the system on an extremal trajectory if and only if the disturbance plays against the control always taking its extreme values.

Expressions (4.4) and (4.5) lead to the following fact,

$$\bigcup_{\langle l_0, l_0\rangle = 1} \mathcal{E}(x_c(t), X_l^-(t)) = \mathcal{X}(t, t_0, \mathcal{E}(x_0, X_0)) = \bigcap_{\langle l_0, l_0\rangle = 1} \mathcal{E}(x_c(t), X_l^+(t)).$$

In practice this means that the more values of $l_0$ we use to compute $X_l^+(t)$ and $X_l^-(t)$, the better will be our approximation.

**Remark about discrete-time systems.** For discrete-time linear system

$$x(t+1) = A(t)x(t) + B(t)u(t, x(t)) + G(t)v(t), \tag{4.1d}$$

the equivalent of (4.3) is

$$l(t+1) = \left(A^T\right)^{-1}(t)l(t), \quad l(t_0) = l_0, \tag{4.3d}$$

which implies nonsingular $A(t)$[10].

For discrete-time systems maxmin and minmax CLRS do not coincide and are computed separately. For maxmin CLRS, the ellipsoidal approximation (external or internal) $\mathcal{E}(x_c(t+1), \overline{X}_l(t+1))$ defined by parameter $l_0 \in \mathbf{R}^n$ is computed as tight external or internal approximating ellipsoid of

$$\mathcal{E}(A(t)x_c(t), A(t)\overline{X}_l(t)A^T(t)) \oplus \mathcal{E}(B(t)p(t), B(t)P(t)B^T(t)) \dot{-} \mathcal{E}(G(t)q(t), G(t)Q(t)G^T(t)),$$

and for minmax CLRS, the ellipsoidal approximation $\mathcal{E}(x_c(t+1), \underline{X}_l(t+1))$ is computed as tight external or internal approximating ellipsoid of

$$\mathcal{E}(A(t)x_c(t), A(t)\underline{X}_l(t)A^T(t)) \dot{-} \mathcal{E}(G(t)q(t), G(t)Q(t)G^T(t)) \oplus \mathcal{E}(B(t)p(t), B(t)P(t)B^T(t))$$

specified by direction $l(t)$ that satisfies (4.3d).

For details and equations related to the discrete-time case, we refer the reader to the manual of the Ellipsoidal Toolbox [4].

Analogous results hold for the backward reach set.

Given the terminal time $t_1$ and ellipsoidal target set $\mathcal{E}_(y_1, Y_1)$, the CLBRS $\mathcal{Y}_{CL}(t_1, t, \mathcal{Y}_1) = \mathcal{Y}(t_1, t, \mathcal{Y}_1)$, $t < t_1$, if it is nonempty, is a symmetric compact convex set whose center is governed by

$$y_c(t) = Ay_c(t) + B(t)p(t) + G(t)q(t), \quad y_c(t_1) = y_1. \tag{4.12}$$

[10]The case when $A(t)$ is singular is described in [25]. The idea is to substitute $A(t)$ with the nonsingular $A_\delta(t) = A(t) + \delta U(t)W(t)$, in which $U(t)$ and $W(t)$ are obtained from the singular value decomposition

$$A(t) = U(t)\Sigma(t)V(t).$$

The parameter $\delta$ can be chosen based on the number of time steps for which the reach set must be computed and the required accuracy. The issue of inverting ill-conditioned matrices is also addressed in [25].

Fix a vector $l_1 \in \mathbf{R}^n$, and consider

$$l(t) = \Phi(t_1, t)^T l_1. \tag{4.13}$$

If the backward reach set $\mathcal{Y}(t_1, t, \mathcal{E}(y_1, Y_1))$ is nonempty, there exist tight external and tight internal approximating ellipsoids $\mathcal{E}(y_c(t), Y_l^+(t))$ and $\mathcal{E}(y_c(t), Y_l^-(t))$ respectively, such that

$$\mathcal{E}(y_c(t), Y_l^-(t)) \subseteq \mathcal{Y}(t_1, t, \mathcal{E}(y_1, Y_1)) \subseteq \mathcal{E}(y_c(t), Y_l^+(t)), \tag{4.14}$$

and

$$\rho(l(t) \mid \mathcal{E}(y_c(t), Y_l^-(t))) = \rho(l(t) \mid \mathcal{Y}(t_1, t, \mathcal{E}(y_0, Y_0))) = \rho(l(t) \mid \mathcal{E}(y_c(t), Y_l^+(t))). \tag{4.15}$$

The equation for the shape matrix of the external ellipsoid is

$$\begin{aligned} \dot{Y}_l^+(t) &= A(t)Y_l^+(t) + Y_l^+(t)A^T(t) - \\ &\quad \pi_l(t)Y_l^+(t) - \frac{1}{\pi_l(t)}B(t)P(t)B^T(t) + \\ &\quad (Y_l^+(t))^{1/2}S_l(t)(G(t)Q(t)G^T(t))^{1/2} + \\ &\quad (G(t)Q(t)G^T(t))^{1/2}S_l^T(t)(Y_l^+(t))^{1/2}, \qquad (4.16) \\ Y_l^+(t_1) &= Y_1, \qquad (4.17) \end{aligned}$$

in which

$$\pi_l(t) = \frac{\langle l(t), B(t)P(t)B^T(t)l(t)\rangle^{1/2}}{\langle l(t), Y_l^+(t)l(t)\rangle^{1/2}},$$

and the orthogonal matrix $S_l(t)$ satisfies the equation

$$S_l(t)(G(t)Q(t)G^T(t))^{1/2}l(t) = \frac{\langle l(t), G(t)Q(t)G^T(t)l(t)\rangle^{1/2}}{\langle l(t), Y_l^+(t)l(t)\rangle^{1/2}}(Y_l^+(t))^{1/2}l(t).$$

The equation for the shape matrix of the internal ellipsoid is

$$\begin{aligned} \dot{Y}_l^-(t) &= A(t)Y_l^-(t) + Y_l^-(t)A^T(t) - \\ &\quad (Y_l^-(t))^{1/2}T_l(t)(B(t)P(t)B^T(t))^{1/2} - \\ &\quad (B(t)P(t)B^T(t))^{1/2}T_l^T(t)(Y_l^-(t))^{1/2} + \\ &\quad \eta_l(t)Y_l^-(t) + \frac{1}{\eta_l(t)}G(t)Q(t)G^T(t), \qquad (4.18) \\ Y_l^-(t_1) &= Y_1, \qquad (4.19) \end{aligned}$$

in which

$$\eta_l(t) = \frac{\langle l(t), G(t)Q(t)G^T(t)l(t)\rangle^{1/2}}{\langle l(t), Y_l^+(t)l(t)\rangle^{1/2}},$$

and the orthogonal matrix $T_l(t)$ is determined by the equation

$$T_l(t)(B(t)P(t)B^T(t))^{1/2}l(t) = \frac{\langle l(t), B(t)P(t)B^T(t)l(t)\rangle^{1/2}}{\langle l(t), Y_l^-(t)l(t)\rangle^{1/2}}(Y_l^-(t))^{1/2}l(t).$$

Just as in the forward reachability case, the terms containing $G(t)$ and $Q(t)$ vanish from equations (4.16) and (4.18) in the absence of disturbances. The boundary value problems (4.12), (4.16) and (4.18) are converted to the initial value problems by the change of variables $s = -t$.

Due to (4.14) and (4.15),

$$\bigcup_{\langle l_1, l_1 \rangle = 1} \mathcal{E}(y_c(t), Y_l^-(t)) = \mathcal{Y}(t_1, t, \mathcal{E}(y_1, Y_1)) = \bigcap_{\langle l_1, l_1 \rangle = 1} \mathcal{E}(y_c(t), Y_l^+(t)).$$

**Remark.** In expressions (4.6), (4.8), (4.16) and (4.18) the terms $\frac{1}{\pi_l(t)}$ and $\frac{1}{\eta_l(t)}$ may not be well defined for some vectors $l$, because matrices $B(t)P(t)B^T(t)$ and $G(t)Q(t)G^T(t)$ may be singular. In such cases, we set these entire expressions to zero.

# 5 Applications

We illustrate the ellipsoidal approach with three applications.

## 5.1 Steering the system to a target

Given system (4.1), target set defined by ellipsoid $\mathcal{E}(y_1, Y_1)$ and terminal time $t_1$, we want to find a closed-loop control that steers the the system from some state $y_0$ at time $t_0 < t_1$ to $\mathcal{E}(y_1, Y_1)$ at $t_1$.

First we compute external and internal ellipsoidal approximations $\mathcal{E}(y_c(t), Y_l^+(t))$ using (4.16), (4.17), and $\mathcal{E}(y_c(t), Y_l^-(t))$ using (4.18), (4.19), $t_0 \leq t < t_1$, for different values of the parameter $l_1 \in \mathbf{R}^n$. If there exists an external ellipsoid $\mathcal{E}(y_c(t_0), Y_l^+(t_0))$ such that $y_0 \notin \mathcal{E}(y_c(t_0), Y_l^+(t_0))$, there is no closed-loop control that can guarantee taking the system from $y_0$ at $t_0$ to a state within $\mathcal{E}(y_1, Y_1)$ at $t_1$. On the other hand, if there exists an internal ellipsoid $\mathcal{E}(y_c(t_0), Y_l^-(t_0))$ defined by the choice of $l_1$, such that $y_0 \in \mathcal{E}(y_c(t_0), Y_l^-(t_0))$, such a control does exist.

We build the closed-loop control $u(t, y(t))$ so as to keep the system state $y(t)$ inside, if possible, or as close as we can, if not, to the internal approximating ellipsoid $\mathcal{E}(y_c(t), Y_l^-(t))$ for $t_0 \leq t < t_1$. The steps below describe control synthesis at time $t$.

1. Compute
$$\gamma(t) = \langle y(t) - y_c(t), (Y_l^-(t))^{-1}(y(t) - y_c(t)) \rangle.$$
If $\gamma(t) \leq 1$, then $y(t) \in \mathcal{E}(y_c(t), Y_l^-(t))$, and the control $u(t, y(t))$ can be chosen arbitrarily in $\mathcal{E}(p(t), P(t))$. For example, set $u(t, y(t)) = p(t)$.

2. Otherwise, if $\gamma(t) > 1$, $y(t)$ is a boundary point of ellipsoid $\mathcal{E}(y_c(t), \gamma(t)Y_l^-(t))$ corresponding to the direction $m(t) \in \mathbf{R}^n$,

$$m(t) = (Y_l^-(t))^{-1}(y(t) - y_c(t)).$$

   In order to steer the system closer to the internal approximating ellipsoid, control $u(t, y(t))$ must act in the direction $-m(t)$.

3. Choose $u(t, y(t))$ so that the vector $B(t)u(t, y(t))$ is a boundary point of the ellipsoid $\mathcal{E}(B(t)p(t), B(t)P(t)B^T(t)) \subset \mathbf{R}^n$ in the direction $-m(t)$,

$$u(t, y(t)) = p(t) - \frac{P(t)B^T(t)m(t)}{\langle m(t), B(t)P(t)B^T(t)m(t)\rangle^{1/2}}.$$

To summarize,

$$u(t, y(t)) = \begin{cases} p(t), \text{ if } \langle y(t) - y_c(t), (Y_l^-(t))^{-1}(y(t) - y_c(t))\rangle \leq 1, \\ p(t) - \frac{P(t)B^T(t)(Y_l^-(t))^{-1}(y(t)-y_c(t))}{\langle (Y_l^-(t))^{-1}(y(t)-y_c(t)), B(t)P(t)B^T(t)(Y_l^-(t))^{-1}(y(t)-y_c(t))\rangle^{1/2}}, \text{ otherwise.} \end{cases} \tag{5.1}$$

The rigorous proof that this closed-loop control works can be found in [22]. In [16] the authors apply this technique to stop a high-dimensional oscillating system using the Ellipsoidal Toolbox for backward reach set computation.

Formula (5.1) holds for discrete-time linear systems, except that instead of $Y_l^-(t)$, shape matrices of internal approximating ellipsoids for maxmin or minmax CLBRS must be used.

## 5.2 Switching system

A *switching system* is a system whose dynamics changes at known times. Consider the RLC circuit shown in figure 1. It has two inputs, the voltage $v$ and current $i$. Define

- $x_1$, the voltage across capacitor $C_1$, so $C_1\dot{x}_1$ is the corresponding current,
- $x_2$, the voltage across capacitor $C_2$, so the corresponding current is $C_2\dot{x}_2$, and
- $x_3$, the current through the inductor $L$, so the voltage across the inductor is $L\dot{x}_3$.

Applying Kirchoff's laws we arrive at the linear system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1C_1} & 0 & -\frac{1}{C_1} \\ 0 & 0 & \frac{1}{C_2} \\ \frac{1}{L} & -\frac{1}{L} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1C_1} & \frac{1}{C_1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix}. \tag{5.2}$$

The parameters $R_1$, $R_2$, $C_1$, $C_2$ and $L$, as well as the inputs, may depend on time. Suppose, for time $0 \leq t < 2$, $R_1 = 2$ Ohm, $R_2 = 1$ Ohm, $C_1 = 3$ F, $C_2 = 7$ F, $L = 2$ H, and

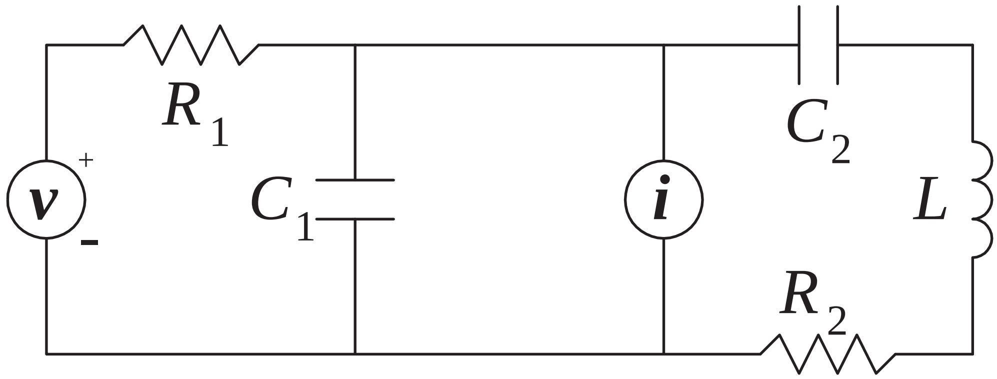


Figure 1: RLC circuit with two inputs.

both inputs, $v$ and $i$ are present and bounded by ellipsoid $\mathcal{E}(0, I)$; and for time $t \geq 2$, $R_1 = R_2 = 2$ Ohm, $C_1 = C_2 = 3$ F, $L = 6$ H, the current source is turned off, and $|v| \leq 1$. Then, system (5.2) can be rewritten as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{cases} \begin{bmatrix} -\frac{1}{6} & 0 & -\frac{1}{3} \\ 0 & 0 & \frac{1}{7} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{6} & \frac{1}{3} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix}, & 0 \leq t < 2, \\ \begin{bmatrix} -\frac{1}{6} & 0 & -\frac{1}{3} \\ 0 & 0 & \frac{1}{3} \\ \frac{1}{6} & -\frac{1}{6} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{6} \\ 0 \\ 0 \end{bmatrix} v, & 2 \leq t. \end{cases} \tag{5.3}$$

We can use the Ellipsoidal Toolbox to compute the reach set of (5.3) for some time $t > 2$, say, $t = 3$.

```
>> % define system 1:
>> A1 = [-1/6 0 -1/3; 0 0 1/7; 1/2 -1/2 -1/2];
>> B1 = [1/6 1/3; 0 0; 0 0];
>> U1 = ellipsoid(eye(2));
>> s1 = linsys(A1, B1, U1);
>>
>> % define system 2:
>> A2 = [-1/6 0 -1/3; 0 0 1/3; 1/6 -1/6 -1/3];
>> B2 = [1/6; 0; 0];
>> U2 = ellipsoid(1);
>> s2 = linsys(A2, B2, U2);
>>
>> X0 = ellipsoid(0.01*eye(3));  % set of initial states
>> L0 = eye(3);  % 3 initial directions
>> TS = 2;  % time of switch
>> T = 3;  % terminal time
>>
>> % compute the reach set:
```

```
>> rs1 = reach(s1, X0, L0, TS);  % reach set of the first system
>> % computation of the second reach set starts
>> % where the first left off
>> rs2 = evolve(rs1, T, s2);
>>
>> % obtain projections onto (x1, x2) subspace:
>> BB = [1 0 0; 0 1 0]';  % (x1, x2) subspace basis
>> ps1 = projection(rs1, BB);
>> ps2 = projection(rs2, BB);
>>
>> % plot the results:
>> subplot(2, 2, 1);
>> plot_ea(ps1, 'r');  % external apprx. of reach set 1 (red)
>> hold on;
>> plot_ia(ps1, 'g');  % internal apprx. of reach set 1 (green)
>> plot_ea(ps2, 'y');  % external apprx. of reach set 2 (yellow)
>> plot_ia(ps2, 'b');  % internal apprx. of reach set 2 (blue)
>>
>> % plot the 3-dimensional reach set at time t = 3:
>> subplot(2, 2, 2);
>> plot_ea(cut(rs2, 3), 'y');
>> hold on;
>> plot_ia(cut(rs2, 3), 'b');
```

Figure 2(a) shows how the reach set of system (5.3) projected onto the $(x_1, x_2)$ plane evolves in time from $t = 0$ to $t = 3$. The external reach set approximation for the first dynamics is in red, the internal approximation is in green. The dynamics switches at $t = 2$. The external reach set approximation for the second dynamics is in yellow, its internal approximation is in blue. The full three-dimensional external (yellow) and internal (blue) approximations of the reach set at $t = 3$ are shown in figure 2(b).

To find out where the system should start at time $t = 0$ in order to reach a neighborhood `M` of the origin at time $t = 3$, we compute the backward reach set from $t = 3$ to $t = 0$.

```
>> M  = ellipsoid(0.01*eye(3));  % terminal set
>> TT = 3;  % terminal time
>>
>> % compute backward reach set:
>> % compute the reach set:
>> brs2 = reach(s2, M, L0, [TT TS]);  % second system comes first
>> brs1 = evolve(brs2, 0, s1);  % then the first system
>>
>> % obtain projections onto (x1, x2) subspace:
>> bps1 = projection(brs1, BB);
```

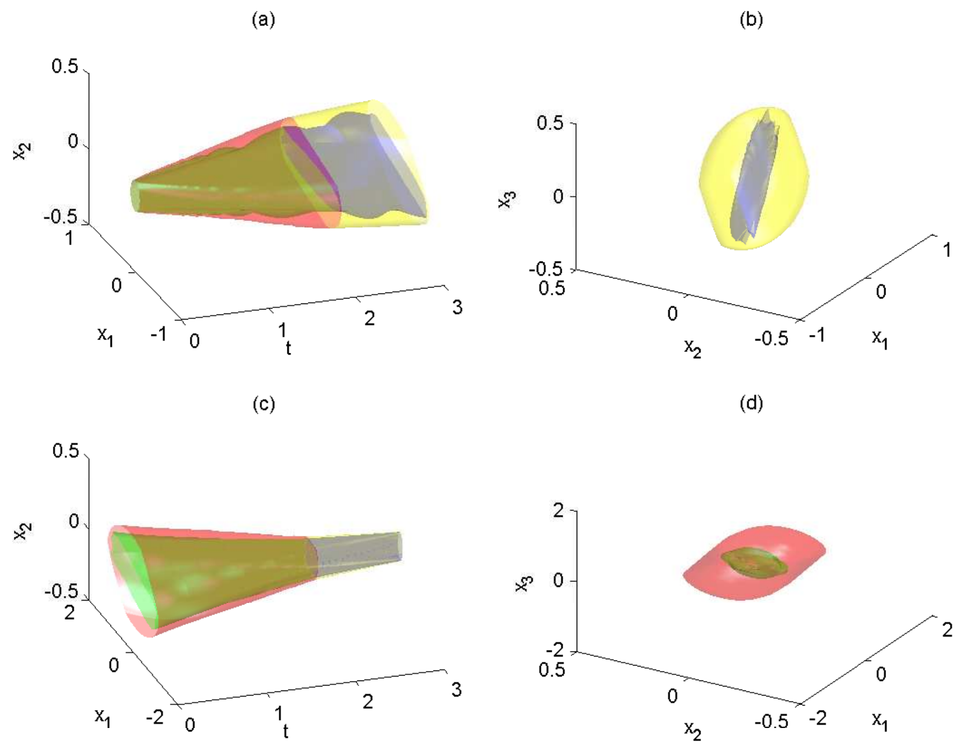


Figure 2: Forward and backward reach sets of the switched system (external and internal approximations). The dynamics switches at $t = 2$.
(a) Forward reach set for the time interval $0 \leq t \leq 3$ projected onto $(x_1, x_2)$ subspace.
(b) Forward reach set at $t = 3$ in $\mathbf{R}^3$.
(c) Backward reach set evolving from $t = 3$ to $t = 0$ projected onto $(x_1, x_2)$ subspace.
(d) Backward reach set at $t = 0$ in $\mathbf{R}^3$.

```
>> bps2 = projection(brs2, BB);
>>
>> % plot the results:
>> subplot(2, 2, 3);
>> plot_ea(bps1, 'r');  % external apprx. of backward reach set 1 (red)
>> hold on;
>> plot_ia(bps1, 'g');  % internal apprx. of backward reach set 1 (green)
>> plot_ea(bps2, 'y');  % external apprx. of backward reach set 2 (yellow)
>> plot_ia(bps2, 'b');  % internal apprx. of backward reach set 2 (blue)
>>
>> % plot the 3-dimensional backward reach set at time t = 0:
>> subplot(2, 2, 4);
>> plot_ea(cut(brs1, 0), 'r');
>> hold on;
>> plot_ia(cut(brs1, 0), 'g');
```

Figure 2(c) presents the evolution of the reach set projected onto the $(x_1, x_2)$ plane in backward time. Again, external and internal approximations corresponding to the first dynamics are shown in red and green, and to the second dynamics in yellow and blue. The full dimensional backward reach set external and internal approximations of system (5.3) at time $t = 0$ is shown in figure 2(d).

### 5.3 Hybrid system

There is no explicit implementation of the reachability analysis for hybrid systems in the Ellipsoidal Toolbox. Nonetheless, the operations of intersection available in the toolbox allow us to work with certain class of hybrid systems, namely, hybrid systems with affine continuous dynamics whose guards are ellipsoids, hyperplanes, halfspaces or polytopes.

We consider the *switching-mode model* of highway traffic presented in [33]. The highway segment is divided into $N$ cells as shown in figure 3. In this particular case, $N = 4$. The traffic density in cell $i$ is $x_i$ vehicles per mile, $i = 1, 2, 3, 4$.

Define

- $v_i$, the average speed in mph, in the $i$-th cell, $i = 1, 2, 3, 4$;
- $w_i$, the backward congestion wave propagation speed in mph, in the $i$-th highway cell, $i = 1, 2, 3, 4$;
- $x_{Mi}$, the maximum allowed density in the $i$-th cell; when this velue is reached, there is a traffic jam, $i = 1, 2, 3, 4$;
- $d_i$, the length of $i$-th cell in miles, $i = 1, 2, 3, 4$;

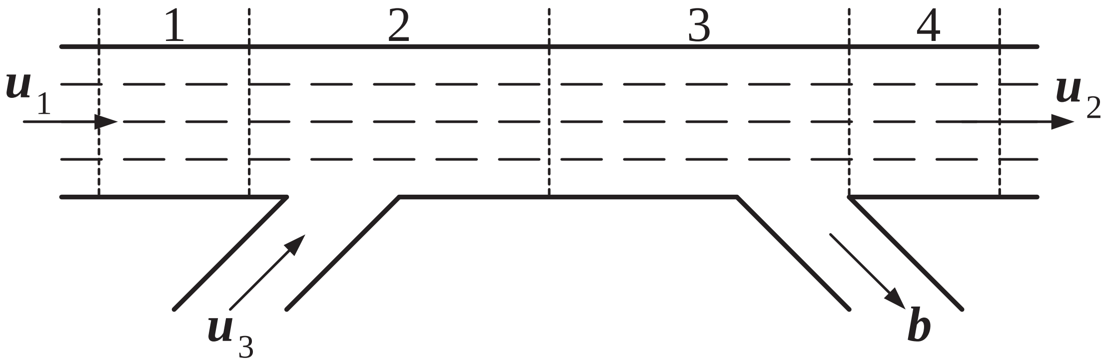


Figure 3: Highway model. Adapted from [33].

- $T_s$, the sampling time in hours;
- $b$, the split ratio for the off-ramp;
- $u_1$, the traffic flow coming into the highway segment, in vehicles per hour (vph);
- $u_2$, the traffic flow coming out of the highway segment (vph);
- $u_3$, the on-ramp traffic flow (vph).

Highway traffic operates in two modes: *free-flow* in normal operation; and *congested* mode, when there is a jam. Traffic flow in free-flow mode is described by

$$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \\ x_3[t+1] \\ x_4[t+1] \end{bmatrix} = \begin{bmatrix} 1-\frac{v_1 T_s}{d_1} & 0 & 0 & 0 \\ \frac{v_1 T_s}{d_2} & 1-\frac{v_2 T_s}{d_2} & 0 & 0 \\ 0 & \frac{v_2 T_s}{d_3} & 1-\frac{v_3 T_s}{d_3} & 0 \\ 0 & 0 & (1-b)\frac{v_3 T_s}{d_4} & 1-\frac{v_4 T_s}{d_4} \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \\ x_3[t] \\ x_4[t] \end{bmatrix} + \begin{bmatrix} \frac{v_1 T_s}{d_1} & 0 & 0 \\ 0 & 0 & \frac{v_2 T_s}{d_2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{5.4}$$

The equation for the congested mode is

$$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \\ x_3[t+1] \\ x_4[t+1] \end{bmatrix} = \begin{bmatrix} 1-\frac{w_1T_s}{d_1} & \frac{w_2T_s}{d_1} & 0 & 0 \\ 0 & 1-\frac{w_2T_s}{d_2} & \frac{w_3T_s}{d_2} & 0 \\ 0 & 0 & 1-\frac{w_3T_s}{d_3} & \frac{1}{1-b}\frac{w_4T_s}{d_3} \\ 0 & 0 & 0 & 1-\frac{w_4T_s}{d_4} \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \\ x_3[t] \\ x_4[t] \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 & \frac{w_1T_s}{d_1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{w_4T_s}{d_4} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$
$$+ \begin{bmatrix} \frac{w_1T_s}{d_1} & -\frac{w_2T_s}{d_1} & 0 & 0 \\ 0 & \frac{w_2T_s}{d_2} & -\frac{w_3T_s}{d_2} & 0 \\ 0 & 0 & \frac{w_3T_s}{d_3} & -\frac{1}{1-b}\frac{w_4T_s}{d_3} \\ 0 & 0 & 0 & \frac{w_4T_s}{d_4} \end{bmatrix} \begin{bmatrix} x_{M1} \\ x_{M2} \\ x_{M3} \\ x_{M4} \end{bmatrix}. \qquad (5.5)$$

The switch from the free-flow to the congested mode occurs when the density $x_2$ reaches $x_{M2}$. In other words, the hyperplane $H([0\ 1\ 0\ 0]^T, x_{M2})$ is the guard. (When the state enters the guard, the system equation switches.)

We indicate how to implement the reach set computation of this hybrid system using the Ellipsoidal Toolbox. We first define the two linear systems and the guard.

```
>> % assign parameter values:
>> v1 = 65; v2 = 60; v3 = 63; v4 = 65;  % mph
>> w1 = 10; w2 = 10; w3 = 10; w4 = 10;  % mph
>> d1 = 2; d2 = 3; d3 = 4; d4 = 2;  % miles
>> Ts = 2/3600;  % sampling time in hours
>> xM1 = 200; xM2 = 200; xM3 = 200; xM4 = 200;  % vehicles per lane
>> b = 0.4;
>>
>> A1 = [(1-(v1*Ts/d1)) 0 0 0
         (v1*Ts/d2) (1-(v2*Ts/d2)) 0 0
         0 (v2*Ts/d3) (1-(v3*Ts/d3)) 0
         0 0 ((1-b)*(v3*Ts/d4)) (1-(v4*Ts/d4))];
>> B1 = [v1*Ts/d1 0 0; 0 0 v2*Ts/d2; 0 0 0; 0 0 0];
>> U1 = ellipsoid([180; 150; 50], [100 0 0; 0 100 0; 0 0 25]);
>>
>> A2 = [(1-(w1*Ts/d1)) (w2*Ts/d1) 0 0
         0 (1-(w2*Ts/d2)) (w3*Ts/d2) 0
         0 0 (1-(w3*Ts/d3)) ((1/(1-b))*(w4*Ts/d3))
         0 0 0 (1-(w4*Ts/d4))];
>> B2 = [0 0 w1*Ts/d1; 0 0 0; 0 0 0; 0 -w4*Ts/d4 0];
>> U2 = U1;
>> G2 = [(w1*Ts/d1) (-w2*Ts/d1) 0 0
```

```
         0 (w2*Ts/d2) (-w3*Ts/d2) 0
         0 0 (w3*Ts/d3) ((-1/(1-b))*(w4*Ts/d3))
         0 0 0 (w4*Ts/d4)];
>> V2 = [xM1; xM2; xM3; xM4];
>>
>> % define linear systems:
>> s1 = linsys(A1, B1, U1, [], [], [], [], 'd');  % free-flow mode
>> s2 = linsys(A2, B2, U2, G2, V2, [], [], 'd');  % congestion mode
>>
>> % define guard:
>> GRD = hyperplane([0; 1; 0; 0], xM2);
```

We assume that initially the system is in free-flow mode. Given a set of initial conditions, we compute the reach set according to dynamics (5.4) for certain number of time steps. We consider an external approximation of the reach set by a single ellipsoid.

```
>> initial conditions:
>> X0 = [170; 180; 175; 170] + 10*ell_unitball(4);
>
>> L0 = [1; 0; 0; 0];  % single initial direction
>> N = 100;  % number of time steps
>>
>> ffrs = reach(s1, X0, L0, N);  % free-flow reach set
>> EA = get_ea(ffrs);  % 101x1 array of external ellipsoids
```

Having obtained the ellipsoidal array EA representing the reach set evolving in time, we determine the ellipsoids in the array that intersect the guard.

```
>> I = hpintersection(EA, GRD);  % some of the intersections are empty
>> D = find(~isempty(I));  % determine nonempty intersections
>> min(D)

ans =

      19

>> max(D)

ans =

      69
```

Analyzing the values in array D, we conclude that the free-flow reach set has nonempty intersection with hyperplane GRD at $t = 18$ for the first time, and at $t = 68$ for the last time.

Between $t = 18$ and $t = 68$ the reach set crosses the guard. Figure 4(a) shows the free-flow reach set projected onto the $(x_1, x_2, x_3)$ subspace for $t = 10$, before the guard crossing; figure 4(b) for $t = 50$, during the guard crossing; and figure 4(c) for $t = 80$, after the guard was crossed.

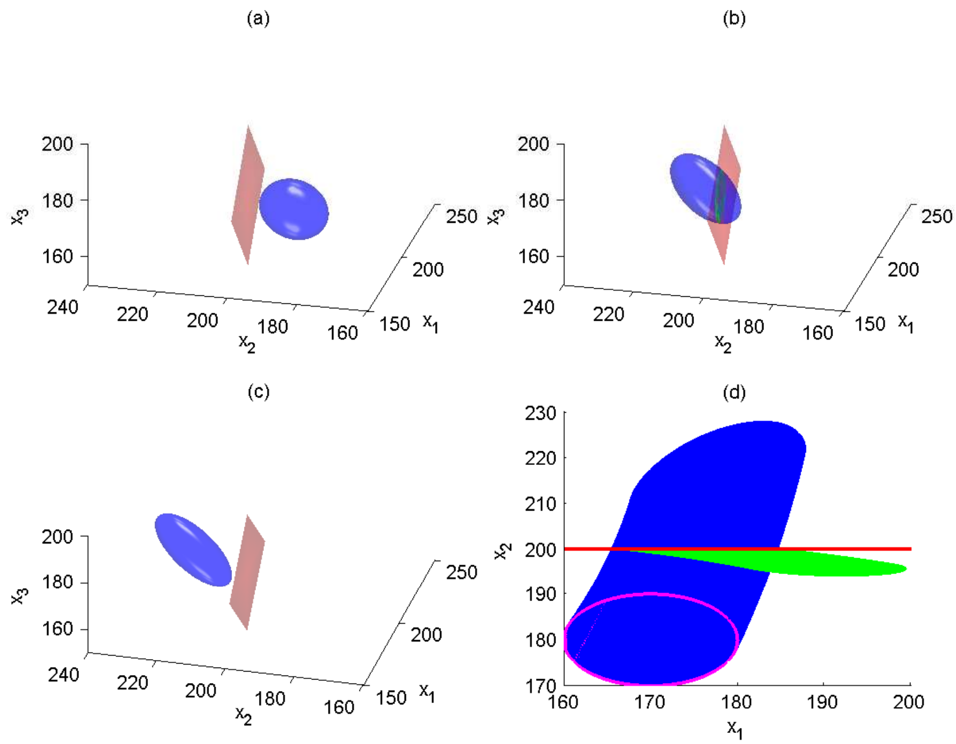


Figure 4: Reach set of the free-flow system is blue, reach set of the congested system is green, the guard is red.
(a) Reach set of the free-flow system at $t = 10$, before reaching the guard (projection onto $(x_1, x_2, x_3)$).
(b) Reach set of the free-flow system at $t = 50$, crossing the guard. (projection onto $(x_1, x_2, x_3)$).
(c) Reach set of the free-flow system at $t = 80$, after the guard is crossed. (projection onto $(x_1, x_2, x_3)$).
(d) Reach set trace from $t = 0$ to $t = 100$, free-flow system in blue, congested system in green; bounds of initial conditions are marked with magenta (projection onto $(x_1, x_2)$).

For each time step that the intersection of the free-flow reach set and the guard is nonempty, we establish a new initial time and a set of initial conditions for the reach set computation according to dynamics (5.5). The initial time is the array index minus one, and the set of initial conditions is the intersection of the free-flow reach set with the guard.

```
>> crs = [];
```

```
>> for i = 1:size(D, 2)
     rs = reach(s2, I(D(i)), L0, [(D(i)-1) N]);
     crs = [crs rs];
   end
```

The union of reach sets in array `crs` forms the reach set for the congested dynamics.

A summary of the reach set computation of the linear hybrid system (5.4-5.5) for $N = 100$ time steps with one guard crossing is given in figure 4(d), which shows the projection of the reach set trace onto the $(x_1, x_2)$ subspace. The system starts evolving in time in free-flow mode from a set of initial conditions at $t = 0$, whose boundary is shown in magenta. The free-flow reach set evolving from $t = 0$ to $t = 100$ is shown in blue. Between $t = 18$ and $t = 68$ the free-flow reach set crosses the guard. The guard is shown in red. For each nonempty intersection of the free-flow reach set and the guard, the congested mode reach set starts evolving in time until $t = 100$. All the congested mode reach sets are shown in green. Observe that in the congested mode, the density $x_2$ in the congested part decreases slightly, while the density $x_1$ upstream of the congested part increases. The blue set above the guard is not actually reached, because the state evolves according to the green region.

## 6 Conclusion

Control problems with hard bounds on the control set and for which finite time behavior has to meet guarantees (reaching or avoiding a target set) despite disturbances, cannot be addressed by traditional methods of design. Central to recent approaches to solving these design problems are the concept and calculation of the open-loop or closed-loop reach set. Effective reach set computational tools have been developed over the past decade. These tools have been described. Among these tools, the ellipsoidal approach is the most promising. That approach, embodied in the Ellipsoidal Toolbox, is illustrated by three examples.

## References

[1] CDD/CDD+ homepage. www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html.

[2] CheckMate homepage. www.ece.cmu.edu/~webk/checkmate.

[3] *d/dt* homepage. www-verimag.imag.fr/~tdang/ddt.html.

[4] Ellipsoidal Toolbox homepage. code.google.com/p/ellipsoids.

[5] Level Set Toolbox homepage. www.cs.ubc.ca/~mitchell/ToolboxLS.

[6] MATISSE homepage. www.seas.upenn.edu/~agirard/Software/MATISSE.

[7] MPT homepage. control.ee.ethz.ch/˜mpt.

[8] PHAVer homepage. www-verimag.imag.fr/˜frehse/phaver_web.

[9] PPL homepage. www.cs.unipr.it/ppl.

[10] Requiem homepage. www.seas.upenn.edu/˜hybrid/requiem/requiem.html.

[11] Zonotope methods on Wolfgang Kühn homepage. www.decatur.de.

[12] Z. Artstein and S. Raković. Feedback and invariance under uncertainty via set-iterates. *Automatica*, 44(2):520–525, 2008.

[13] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise linear dynamical systems. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 21–31. Springer, 2000.

[14] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry: Theory and Applications*, 7:265–301, 1997.

[15] F. Blanchini. Set-invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[16] A. N. Daryin, A. B. Kurzhanskiy, and I. V. Vostrikov. Reachability approaches and ellipsoidal techniques for closed-loop control of oscillating systems under uncertainty. In *IEEE Conference on Decision and Control*, Seville, Spain, 2006.

[17] A. Girard. Reachability of uncertain linear systems using zonotopes. In M. Morari, L. Thiele, and F. Rossi, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2005.

[18] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2006.

[19] E. K. Kostousova. Control synthesis via parallelotopes: optimization and parallel computations. *Optimization Methods and Software*, 14(4):267–310, 2001.

[20] N. N. Krasovski and A. I. Subbotin. *Positional Differential Games*. Springer Verlag, New York, 1988.

[21] A. B. Kurzhanski. Set-valued Calculus and Dynamic Programming in Problems of Feedback Control. volume 124 of *Series of Numerical Mathematics*, pages 163–174. Birkhäuser Verlag, 1998.

[22] A. B. Kurzhanski and I. Vályi. *Ellipsoidal Calculus for Estimation and Control*. ser. SCFA. Birkhäuser, 1997.

[23] A. B. Kurzhanski and P. Varaiya. On ellipsoidal techniques for reachability analysis. *Optimization Methods and Software*, 17:177–237, 2000.

[24] A. B. Kurzhanski and P. Varaiya. Dynamic optimization for reachability problems. *Optimization Theory and Applications*, 108(2):227–251, 2001.

[25] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 2007.

[26] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-Parametric Toolbox (MPT). In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 448–462. Springer, 2004.

[27] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32:231–253, 2001.

[28] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. Wiley, New York, 1961.

[29] G. Leitmann. Optimality and reachability with feedback controls. In A. Blaquiere and G. Leitmann, editors, *Dynamical Systems and Microphysics*, volume 1790, pages 119–141. Academic Press, 1982.

[30] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370, 1999.

[31] I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 21–31. Springer, 2000.

[32] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. The double description method. In H. W. Kuhn and A. W. Tucker, editors, *Conttributions to Theory of Games*, volume 2. Princeton University Press, 1953.

[33] L. Muñoz, X. Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *American Control Conference*, pages 3750–3755, 2003.

[34] S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

[35] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusion. In *Proceedings of the 6th International Conference on Computer Aided Verification*, volume 818 of *Lecture Notes in Computer Science*, pages 95–104. Springer, 1994.

[36] O. Stursberg and B. H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, volume 2623 of *Lecture Notes in Computer Science*, pages 482–497. Springer, 2003.

[37] P. Varaiya. Reach set computation using optimal control. Proc. of KIT Workshop on Verification on Hybrid Systems. Verimag, Grenoble., 1998.