Basics about Sets

REPLAN team

February 25, 2025

Outline

Outline

Why sets?

Sets are and have often been used in control and motion planning topics:

- for computing reachable sets (e.g., obstacle avoidance in a “zero-sum game” context, formal validation, solving first order partial differential Hamilton-Jacobi equations)
- invariance analysis for linear dynamics under uncertainties (unknown but bounded)
- dynamical programming
- characterizations of uncertain dynamics

Outline

Set operations

- We all know how to sum, substract, project or compute the distance between numbers in $\mathbb{R}^n, \mathbb{C}^n$;
- But what happens when the numbers become sets?
- We need to introduce new operations, especially tailored for sets!

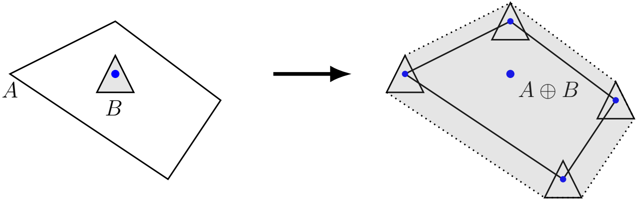
Standard operations and their set correspondent:

- Minkowski sum: $x + y \quad \rightarrow \quad X \oplus Y$
- Pontryagin difference: $x - y \quad \rightarrow \quad X \ominus Y$
- Hausdorff distance: $d(x, y) \quad \rightarrow \quad d(X, Y)$
- Projection and Cartesian product

Minkowski addition

$$A \oplus B = \{x + y : \forall x \in A,\ y \in B\}$$

- used to “enlarge” sets
- and to characterize reachability



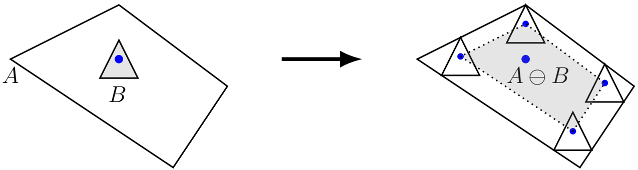
Problem

In general, the complexity quickly increases!

Pontryagin difference

$$A \ominus B = \{x :\in A : \; x + y \in A, \forall y \in B\}$$

- used to "tighten" sets
- useful to discard noise and take into account disturbances



Problem

- The result may easily become the empty set (whenever $\nexists x \in A$ s.t. $x \oplus A \subseteq B$)
- In general $(A \ominus B) \oplus B \subset A$

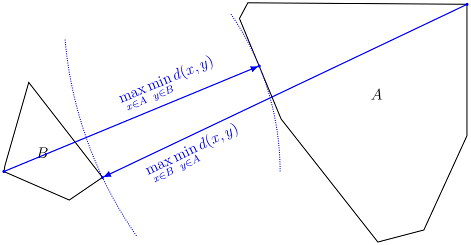
Hausdorff distance

$$d(A,B) = \max\left\{ \max_{x\in A}\min_{y\in B} d(x,y),\ \max_{x\in B}\min_{y\in A} d(x,y) \right\}$$

- a measure for the distance between two sets
- vanishes if and only if $A = B$
- alternative definition:

  $d(A,B) = \max\{\epsilon_1, \epsilon_2\}$ with

  $$\epsilon_1 = \min_{\epsilon}\{\epsilon :\ A \subseteq B \oplus \epsilon\mathcal{B}_2^n\},\ \epsilon_2 = \min_{\epsilon}\{\epsilon :\ B \subseteq A \oplus \epsilon\mathcal{B}_2^n\}$$



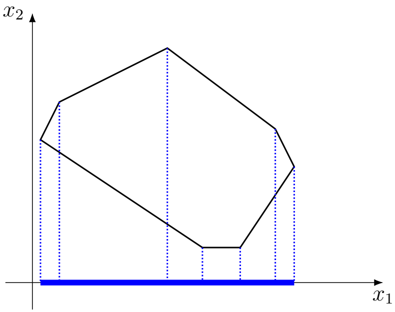
Problem

We need to look in both directions, otherwise $A \subseteq B$ implies $d(A,B) = 0$!

Projection and Cartesian product

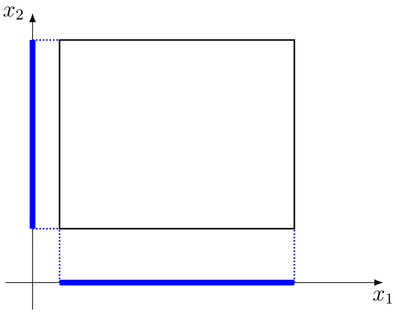
For a set $A \subset \mathbb{R}^{n+m}$, its orthogonal projection onto $\mathbb{R}^n$ is the set given by:

$$\pi_x(A) = \left\{ x \in \mathbb{R}^n : \exists y \text{ s.t. } \begin{bmatrix} x \\ y \end{bmatrix} \in A \right\}$$



For two sets $A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^m$, their cartesian product is given by:

$$A \otimes B = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : \forall x \in A, \forall y \in B \right\}$$

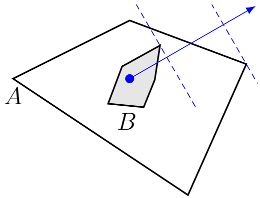


Support function idea

The support function

$$h_A(\eta) = \max_{x \in A} \eta^\top x$$

is useful, for example in checking set inclusion!



$$A \subseteq B \quad \Leftrightarrow \quad h_A(\eta) \leq h_B(\eta), \quad \forall \eta \in \mathbb{R}^n \setminus \{0\}$$

It allows to reformulate the set operations:

- scalar/matrix multiplication:

$$h_{MA}(\eta) = M^\top h_A(\eta)$$

- Minkowski addition:

$$h_{A \oplus B}(\eta) = h_A(\eta) + h_B(\eta)$$

- Pontryagin difference (if A convex and $A \ominus B \neq \emptyset$):

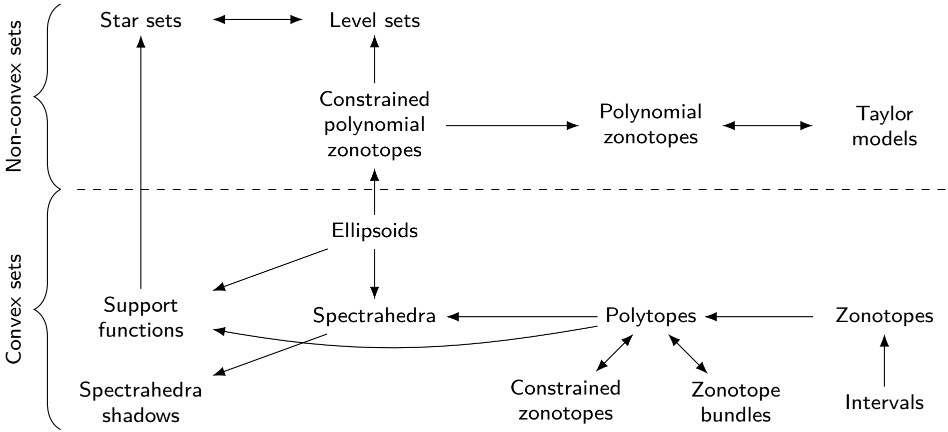
$$h_{A \ominus B}(\eta) \leq h_A(\eta) - h_B(\eta)$$

- Hausdorff distance:

$$d(A, B) = \max_{\|\eta\| \leq 1} |h_A(\eta) - h_B(\eta)|$$

Outline

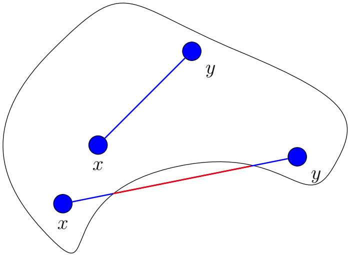
Set types (an incomplete phylogenetic tree)



Set types (some remarks)

- the clearest demarcation is between convex and non-convex sets
- better representational capability means higher algorithm complexity
- some operations are not closed for a given family (e.g., the intersection of ellipsoids is not an ellipsoid)
- implementations are often platform and language-specific

What is convexity actually?

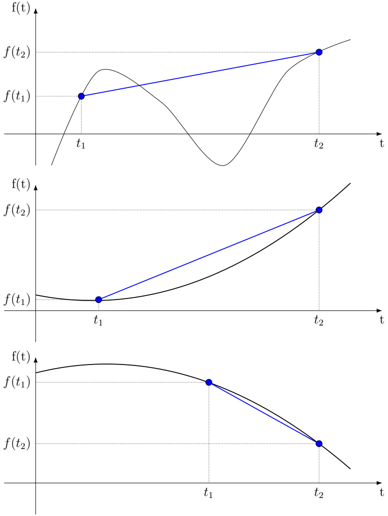


- set convexity:

$$\forall x, y \in S, \text{ we have } \lambda x + (1-\lambda)y \in S, \forall \lambda \in [0,1]$$

- function convexity:

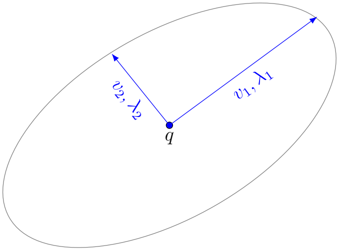
$$f(\lambda t_1 + (1-\lambda)t_2) \leq \lambda f(t_1) + (1-\lambda)f(t_2), \forall t_1, t_2 \text{ and } \lambda \in [0,1]$$



Ellipsoids

Defining characteristics:

- defined by a center $q \in \mathbb{R}^n$ and shape matrix $Q \succeq 0$ from $\mathbb{R}^{n\times n}$
- traditionally used in control (Lyapunov, Riccati equations, etc.)
- strong/resilient algorithms almost impervious to problem size
- not closed for sum, intersection, difference



An example (the continuous Lyapunov eq.):

$$(A+BK)X + X(A+BK)^\top + Q \preceq 0$$

$$\mathcal{E} = \{x \in \mathbb{R}^n : \ \ell^\top x \le \ell^\top q + \sqrt{\ell^\top Q \ell}, \quad \forall \ell \in \mathbb{R}^n\}$$

or, equivalently (if $Q \succ 0$)

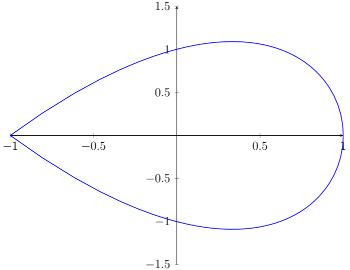
$$\mathcal{E} = \{x \in \mathbb{R}^n : \ (x-q)^\top Q^{-1}(x-q) \le 1\}$$

Spectrahedra

A spectrahedron may be expressed as the collection of points verifying a linear matrix inequality (LMI):

$$A(x) = \{x \in \mathbb{R}^n : \; A_0 + \sum_{i=1}^{n} A_i x_i \succeq 0\}$$

with $A_i \succeq 0$ for $i = 0, \dots, n$.



Defining characteristics:

- always convex
- they may actually be represented as a sum of squares!
- more difficult to handle (complexity $\approx n^{6.5}$)
- but powerful approximants (the Lasserre moment hierarchy)

$$A(x) = \begin{bmatrix} 2-2x_1 & x_2 & -1+x_1 \\ x_2 & 1+x_1 & 0 \\ -1+x_1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_2$$

Spectrahedra shadows

A spectrahedral shadow is the projection of a spectrahedron:

$$A(x) = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m, \text{ s.t. } A_0 + \sum_{i=1}^{n} A_i x_i + \sum_{j=1}^{m} B_j y_j \succeq 0\}$$

with $A_i, B_j \succeq 0$ for $i = 0, \ldots, n$ and $j = 0, \ldots, n$. Equivalently:

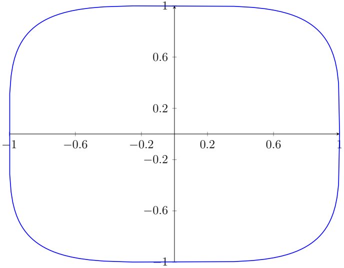
$$A(x) = \{x \in \mathbb{R}^n : x = c + G\beta, \text{ where } A_0 + \sum_{i=1}^{n} A_i \beta_i \succeq 0\}$$

Defining characteristics:

- always convex
- more general than spectrahedra
- can approximate any basic semialgebraic set!

The "TV screen" $A(x) = \{x \in \mathbb{R}^2 : 1 - x_1^4 - x_2^4 \geq 0\}$ is not a spectrahedron but it may be represented as

$$A(x) = \{\exists y \in \mathbb{R}^2 \text{ s.t. } : \begin{bmatrix} 1+y_1 & y_2 & 0 & 0 & 0 & 0 \\ y_2 & 1-y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x_1 & 0 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 \end{bmatrix} \succeq 0\}$$



Hyperplane and half-space

Let's introduce the basic notions of

- **hyperplane**: the set of form

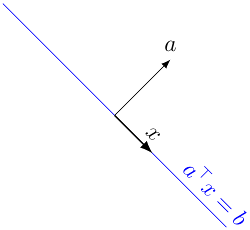
$$\left\{x \in \mathbb{R}^n : \ a^\top x = b\right\}$$

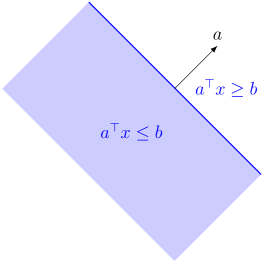
- **halfspace**: the set of form

$$\left\{x \in \mathbb{R}^n : \ a^\top x \le b\right\}$$

for $a \neq 0$ and $(a, b) \in \mathbb{R}^n \times \mathbb{R}$

A pair (a, b) will determine three sets, the halfspaces $\mathcal{H}^-, \mathcal{H}^+$ and the hyperplane $\mathcal{H}^0$ which is their separating boundary.





Polyhedral sets

Dual representation:

- half-space representation

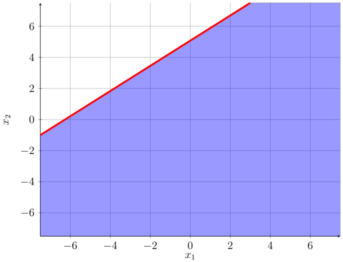
$$X = \{x \in \mathbb{R}^d : F_i^\top x \leq \theta_i, i = 1 \dots n_h\},$$

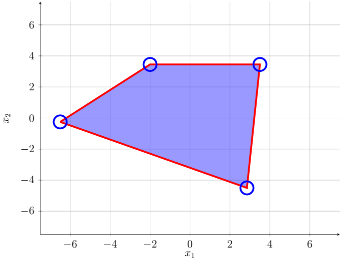
- Vertex representation

$$X = \left\{x \in \mathbb{R}^d : x = \sum_{j=1}^{n_v} \alpha_j v_j,\ \sum_{j=1}^{n_v} \alpha_j = 1,\ \alpha_j \geq 0\right\}.$$

Defining characteristics:

- can approximate arbitrarily-well convex sets
- robust to small and medium-sized problem sizes
- can be embedded in large-scale LP/QP optimization problems





Polyhedral sets (about their complexity)

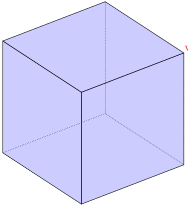
Theorem (McMullen's upper bound)

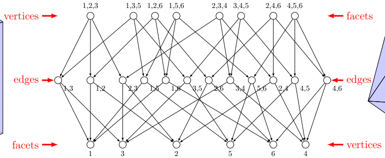
The maximum number $f_k(P)$ of k-faces of a polytope $P \in \mathbb{R}^n$ with N_h facets is attained by the cyclic polytope $c(n, N_h)$:

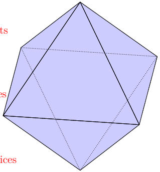
$$f_k(P) \leq f_{n-k-1}(c(n, N_h)), \quad \forall k = 0, 1, \dots, n-2,$$

$$\textit{where:} \qquad f_k(c(n, N_h)) = \sum_{r=0}^{\lfloor n/2 \rfloor} \binom{r}{n-k-1} \binom{N_h - n + r - 1}{r} + \sum_{r=\lfloor n/2 \rfloor + 1}^{n} \binom{r}{n-k-1} \binom{N_h - r - 1}{n-r}.$$

- hypercube:
  $B^n_\infty = \left\{ x \in \mathbb{R}^n : \ |x_i| \leq 1, \ \forall i = 1, \dots, n \right\}$
- cross-polytope:
  $\Delta^n = \left\{ x \in \mathbb{R}^n : \ \sum_{i=1}^{n} |x_i| \leq 1 \right\}$







Intervals

- interval $\mathbf{x} \in \mathbb{IR}$:

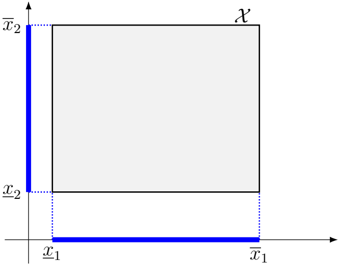
$$\mathbf{x} = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq xc \leq \overline{x}\}$$

- box or interval vector $\in \mathbb{IR}^n$:

$$\mathcal{X} \triangleq \mathbf{x_1} \times \ldots \times \mathbf{x_n}$$

$$\text{or, alternatively,}$$

$$\mathcal{X} \triangleq \begin{bmatrix} \mathbf{x_1} & \ldots & \mathbf{x_n} \end{bmatrix}^\top$$



Characteristics:

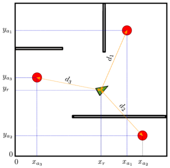
- width $w(\mathbf{x}) \triangleq \overline{x} - \underline{x}$
- center $mid(\mathbf{x}) \triangleq \frac{1}{2}(\overline{x} - \underline{x})$

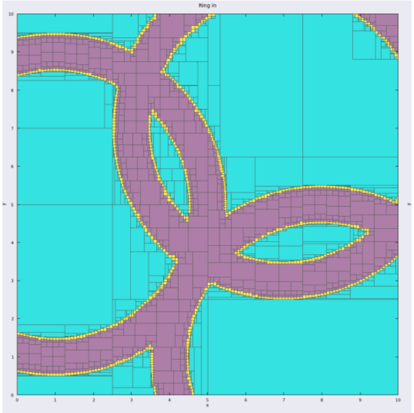
Main advantage

(Almost) any unary or binary operator can be extended to intervals, and, implicitly, to boxes. $\rightarrow$ Interval Arithmetic (i.e., the result is the smallest interval containing the set):

- for $\mathbf{x} \in \mathbb{IR}^n$ and $u \in \{\sqrt{}, \log, \exp, \ldots\}$: $u : \mathbb{R}^n \mapsto \mathbb{R}^n, \quad u(\mathbf{x}) \triangleq \{u(x), x \in \mathbf{x}\} \subseteq \mathbb{R}^n$
- for $\mathbf{x}, \mathbf{y} \in \mathbb{IR}^n$ and $\diamond \in \{+, -, \cdot, /, \ldots\}$ $\diamond : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n, \quad \mathbf{x} \diamond \mathbf{y} \triangleq \{x \diamond y, x \in \mathbf{x}, y \in \mathbf{y}\} \subseteq \mathbb{R}^n$

Intervals (Example - SLAM)





- localization of a robot navigating in a field of landmarks
- Goal: estimate (x_r, y_r)
- 3 landmarks $\rightarrow$ 3 solution sets:
  $\mathbb{S}_i = \{x \in \mathbb{R} \mid (x_r - x_{ai})^2 + (y_r - y_{ai})^2 \in \mathbf{d}_i^2\}$
- SIVIA algorithm for $\mathbb{S}_i$ divides the space in:
  - green boxes - $\notin \mathbb{S}_i$
  - red boxes - $\in \mathbb{S}_i$
  - yellow boxes - uncertain

Zonotopes

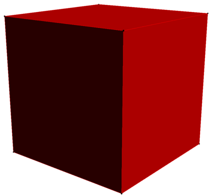
Are a particular case of centrally-symmetric polytopes with two interpretations:

- the projection of a higher-dimension hyper-cube
- Minkowski sum of segments

$$\langle c, G\rangle = \left\{ c + \sum_{i=1}^{D} G_i \lambda_i : \ |\lambda_i| \leq 1, i = 1 \dots n_g \right\}$$

Defining characteristics:

- closed for Minkowski sum and matrix multiplication
- have an equivalent half-space representation
- more compact representation $\implies$ more efficient operations
- not closed under set intersection



Constrained zonotopes

$Z \subset \mathbb{R}^n$ is a constrained zonotope if exist (c, G, F, θ) such that:

$$Z = \langle c, G, F, \theta \rangle = \{x \in \mathbb{R}^n : \; x = c + G\lambda, \; \|\lambda\|_\infty \leq 1, F\lambda = \theta\} .$$

Defining characteristics:

- closed under affine transformation:

$$r + RZ_1 = \langle r + Rc_1, RG_1, F_1, \theta_1 \rangle$$

- closed under Minkowski sum:

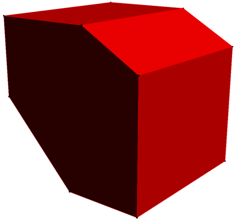
$$Z_1 \oplus Z_2 = \left\langle c_1 + c_2, \begin{bmatrix} G_1 & G_2 \end{bmatrix}, \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right\rangle$$

- is equivalent with a polyhedral set

Closed under set intersection:

$$Z_1 \cap Z_2 = \left\langle c_1, \begin{bmatrix} G_1 & 0 \end{bmatrix}, \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \\ G_1 & -G_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \\ c_2 - c_1 \end{bmatrix} \right\rangle,$$

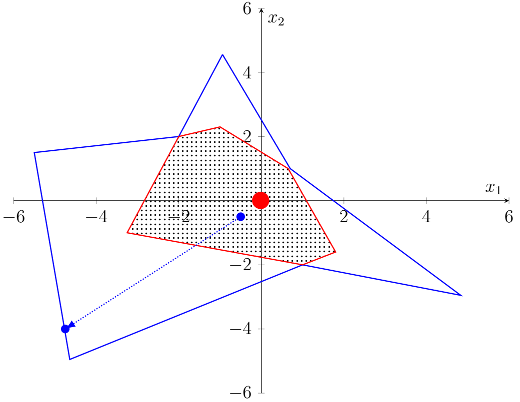
where $Z_1 = \langle c_1, G_1, F_1, \theta_1 \rangle$ and $Z_2 = \langle c_2, G_2, F_2, \theta_2 \rangle$.



Star-shaped sets

Defining characteristics:

- there exists a kernel from which any segment finishing on the boundary, remains inside
- can be used as a pseudo-norm
- closed under linear transformations
- closed under Minkowski sum and Cartesian product
- not closed under intersection or union



Outline

Set propagation

A typical dynamical equation (continuous time):

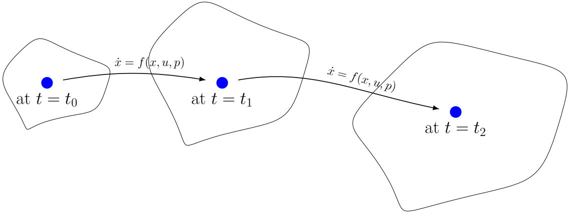
$$\dot{x}(t) = f(\underbrace{x(t)}_{\text{states}}, \quad \underbrace{u(t)}_{\text{inputs}}, \quad \underbrace{p}_{\text{parameters / disturbances}})$$

Simpler cases are possible:

- affine in the input: $\dot{x} = f(x, p) + g(x, p)u$
- linear time invariant: $\dot{x} = Ax + Bu + Ep$

In conjunction with sets, we may consider:

- forward and backward propagation
- admissibility
- invariance



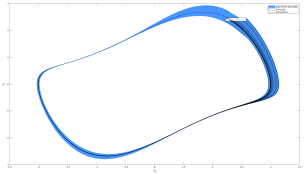
Forward reachability

The forward evolution of a dynamical system from initial time t_0 to final time t_f,

- starting from the initial set $x_0 \in \mathcal{X}_0$
- with a set of input values $u(t) \in \mathcal{U}(t)$
- and a set of parameter values $p \in \mathcal{P}$

The Van der Pol oscillator

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \left(1 - x_1^2\right) x_2 - x_1 + u$$



is given by the forward reachable set

$$\mathcal{R}(t_f) = \{\chi(t_f; x_0, u(\cdot), p) \in \mathbb{R}^n : \ x_0 \in \mathcal{X}_0, \forall t_0 \leq t \leq t_f : \ u(t) \in \mathcal{U}(t), p \in \mathcal{P}\}$$

where $\chi(\cdot)$ denotes the system's trajectory under a particular combination of initial state, input sequence and parameters:

$$\chi(t_f; x_0, u(\cdot), p) = \int_{t_0}^{t_f} f(x(t), u(t), p)\, dt, \qquad \text{where } x(t_0) = x_0$$

Trivia:

- much easier to compute when the sets $\mathcal{X}_0$, $\mathcal{U}$ and $\mathcal{P}$ are time-invariant and convex;
- it can be computed over an interval: $\mathcal{R}([t_0, t_f]) = \bigcup_{t \in [t_0, t_f]} \mathcal{R}(t)$

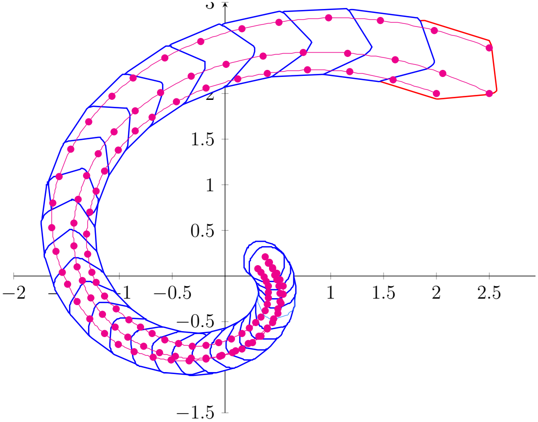
Forward reachability (an example)

Computing the sets is so much easier in discrete time and for linear dynamics!

- linear dynamics:

$$\dot{x} = \begin{bmatrix} -0.7 & -2 \\ 2 & -0.7 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta$$

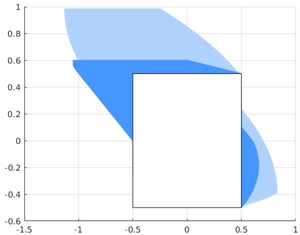
- simulation time, $t_f = 3$, 25 steps
- initial set, $x_0 \in \mathcal{X}_0 = \left\langle \begin{bmatrix} 2 \\ 2 \end{bmatrix}, 2.5 \cdot I_2 \right\rangle$
- disturbance set, $\delta \in \Delta = \langle 0, 0.1 \rangle$



Backward reachability

The backward evolution of a dynamical system from time t_0 over a time interval τ,

- given a target set $x_0 \in \mathcal{X}_0$
- with a set of input values $u(t) \in \mathcal{U}(t)$
- and a set of disturbance values $w(t) \in \mathcal{W}(t)$



is given by the backward reachable set (in two forms)

$$\mathcal{R}_{AE}(-\tau) = \{x_0 \in \mathbb{R}^n : \forall u(t) \in \mathcal{U}(t), \exists w(t) \in \mathcal{W}(t), \exists t \in \tau : \chi(t; x_0, u(\cdot), w(\cdot)) \in \mathcal{X}_0\}$$
$$\mathcal{R}_{EA}(-\tau) = \{x_0 \in \mathbb{R}^n : \exists w(t) \in \mathcal{W}(t), \forall u(t) \in \mathcal{U}(t), \exists t \in \tau : \chi(t; x_0, u(\cdot), w(\cdot)) \in \mathcal{X}_0\},$$

depending on the order in which the information is received (is the control action knowing the value of the disturbance?)

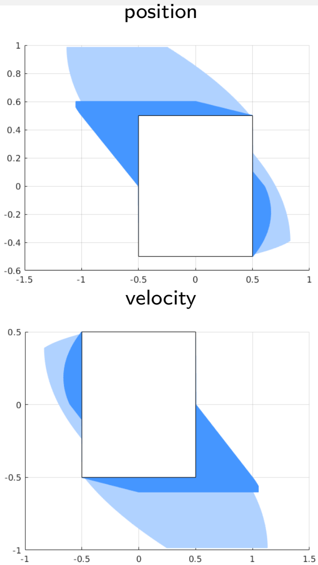
Trivia:

- much easier to compute when the sets $\mathcal{X}_0$, $\mathcal{U}$ and $\mathcal{W}$ are time-invariant and convex;
- it can be computed over an interval or at a certain moment of time (depending on how we choose τ)

Backward reachability (an example)

The pursuit-evasion game:

- linear dynamics:
$$\ddot{x}(t) = u(t) - w(t)$$
with u – the acceleration of the pursuer and δ – the acceleration of the evader
- potential collision time at $t_0 = 0$, go backward in time $t_f = 1$
- to avoid the target set, $x(0) \in \mathcal{X}_0 = \langle 0, 0.5 \cdot I_4 \rangle$
- allowable control actions in $u(t) \in \mathcal{U}(t) = \left\langle \begin{bmatrix} -0.5 \\ -0.1 \end{bmatrix}, \mathrm{diag}(0.1, 0.5) \right\rangle$
- possible disturbance values in $w(t) \in \mathcal{W}(t) = \left\langle \begin{bmatrix} 0 \\ -0.2 \end{bmatrix}, \mathrm{diag}(0.2, 0) \right\rangle$
- AE (light blue) and EA (dark blue) variants



Set truncation

Many approaches aim to

- control the complexity (number of inequalities, of vertices, of generators, ...)
- but avoid major wrapping effects (where the errors due to the over-approximation accumulate)

Some ideas (particularly for zonotopes):

- cluster together generators with small magnitude using an outer box approximation
- re-compute the generators to minimize some error cost (volume change, PCA, etc.)

Longer simulation horizons lead to increased errors. For set-based estimation one idea is to correct with output information (prior and a posteriori update)

Fixed complexity idea

- Consider the discrete-time linear dynamics

$$x^+ = Ax + \delta, \quad \delta \in \Delta$$

- and polytopic sets, given in half-space formulation:

$$S(\theta) = \{x :\in \mathbb{R}^n : \ Fx \leq \theta\}, \quad \Delta = \{x :\in \mathbb{R}^m : \ F_\delta \delta \leq \theta_\delta\}$$

- The one-step forward invariant set is given by

$$x \in \Omega \xrightarrow{apply \ \ dynamics} x^+ \in \Omega^+ = A\Omega \oplus \Delta$$

The idea: keep the complexity constant!

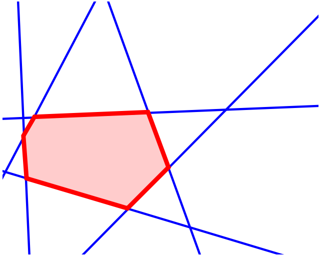
If $\Omega = S(\theta)$, find θ^+ such that $\Omega^+ \subseteq S(\theta^+)$

- Consider matrix $H \geq 0$ such that $HF = FA$ holds.
- Then, the set inclusion condition may be written as

$$AS(\theta) \oplus \Delta \subseteq S(\theta^+)$$

- which translates into the sufficient condition

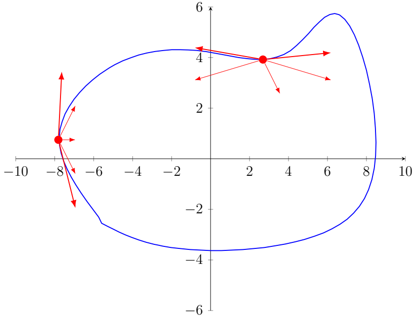
$$H\theta + \max_{\delta \in \Delta} F\delta \leq \theta^+$$



Set invariance

Why re-compute the set at each dynamic update? Better to find one whose shape is invariant!

Set invariance = All system's trajectories remain inside the set once they have entered it



- **Nagumo's theorem**: For dynamics $\dot{x}(t) = f(x(t))$, set S is positively invariant iff

$$f(x) \in \mathcal{T}_S(x), \quad \forall x \in S$$

  with tangent cone $\mathcal{T}_S(x) = \left\{ z : \lim_{\tau \to 0} \inf \frac{\mathrm{dist}(x+\tau z, S)}{\tau} = 0 \right\}$

- in the discrete case, we only need to ensure that at the next step, the trajectory is still inside

In both cases, the geometrical condition is the same:

Discrete case: $x^+ = f(x, u, \delta)$ Continuous case: $\dot{x} = f(x, u, \delta)$

$$f(S, U, \Delta) \subseteq S$$

Set invariance (variations)

Depending on the nature of the dynamics and on the controllable terms, there are many variations[1]:

- For dynamics $x^+ = f(x, \delta)$, the set S is positive robust invariant iff

$$f(S, \Delta) \subseteq S$$

- For dynamics $x^+ = f(x, u, \delta)$, the set S is controlled robust invariant iff

$$\exists u \in U \text{ s.t. } f(S, u, \Delta) \subseteq S$$

- it becomes simpler in the linear case:

$$AS \oplus \Delta \subseteq S, \quad \exists u \text{ s.t. } AS \oplus \{u\} \oplus \Delta \subseteq S$$

Some shorthands:

- PI - positive invariance
- RPI - robust positive invariance
- CI - control invariance
- RCI - robust control invariance

[1] The set inclusions conditions are the same so we stay in the discrete case.

Controlled invariance for ellipsoidal sets

Assumptions:

- linear dynamics
  $$\dot{x} = Ax + Bu$$
  with the pair A, B controllable
- consider an ellipsoidal set
  $$\mathcal{E} = \{x : \; x^\top P x \leq 1\}$$

Find K such that $u = Kx$ makes the closed dynamics stable!

Algorithm:

- the tangent cone is
  $$\mathcal{T}_{\mathcal{E}}(x) = \{z : \; x^\top P z \leq 0\}$$
- leading to the controlled invariance condition
  $$x^\top P(A - BK)x \leq 0$$
- equivalent with the Lyapunov condition
  $$(A - BK)^\top P + P(A - BK) \preceq 0$$
- note $Q = P^{-1}$, $L = KQ$ and see where it goes...

Controlled invariance for ellipsoidal sets (an example)

- linear dynamics

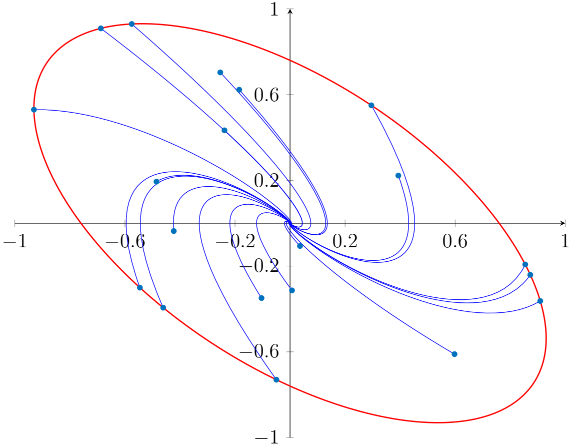
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

- resulted ellipsoid shape matrix is

$$P = \begin{bmatrix} 1.7321 & 1.0000 \\ 1.0000 & 1.7321 \end{bmatrix}$$

- and the static feedback gain is

$$K = \begin{bmatrix} 1 & 1.7321 \end{bmatrix}$$



The mPRI set

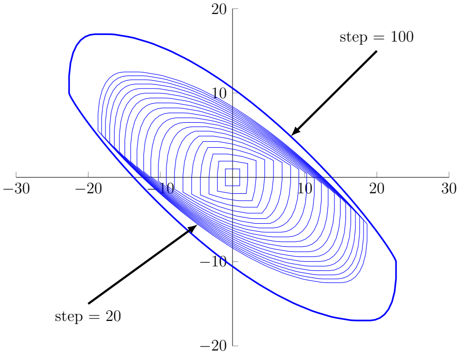
The minimal RPI set has two equivalent definitions:

- the set contained in all other RPI sets
- the fix point of iterating the dynamics

$$\Omega_\infty = \underbrace{f(f(\ldots,\Delta),\Delta)}_{\infty \text{ iterations}} = \lim_{k\to\infty} f^{(k)}(0,\Delta)$$

$$= \bigoplus_{i=0}^{\infty} A^i B\Delta.$$

- it is useful to analyze "worst-case" behavior and provide safety bounds
- in general, there is no explicit solution



The iterative idea:

- construct $\Omega_s = \bigoplus_{i=0}^{s} A^i\Delta$
- find $\alpha < 1$ s.t. $A^s\Delta \subseteq \alpha\Delta$

- then, we have

$$\Omega_s \subset \Omega_\infty \subset (1-\alpha)^{-1}\,\Omega_s$$

- with $(1-\alpha)^{-1}\,\Omega_s$ an RPI set

Ultimate bounds idea

Consider:

- the stable, discrete-time[a] system

$$x^{+} = Ax + E\delta$$

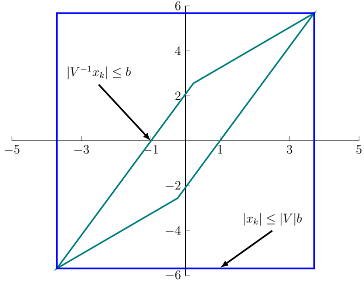
- with Jordan decomposition $A = V\Lambda V^{-1}$
- and bound $|\delta_k| \leq \bar{\delta}, \quad \forall k \geq 0$

Then there exists $\ell(\epsilon)$ such that for all $k \geq \ell$:

$$x_k \in \Omega_{UB}(\epsilon) = \left\{ x : \mathbb{R}^n : |V^{-1}x| \leq (I - |\Lambda|)^{-1} |V^{-1}E|\bar{\delta} + \epsilon \right\}$$

$$x_k \in B_{UB}(\epsilon) = \left\{ x : \mathbb{R}^n : |x| \leq |V| (I - |\Lambda|)^{-1} |V^{-1}E|\bar{\delta} + |V|\epsilon \right\}$$

[a] A continuous-time counterpart exists.



Relevant details:

- conservative, but any iteration of an RPI is also an RPI
- applies to stable systems having real eigenvalues
- complex eigenvalues describe an intersection of ellipsoids, no longer a parallelogram

Ultimate bounds RPI set (upper bound for the convergence time)

Any trajectory starting in x° will reach $\Omega_{UB}(\epsilon)$ in at most

$$k\left(x^{\circ}, \Omega_{UB}\left(\epsilon\right)\right) = \max\left\{\lceil \ell_1 \rceil, \ldots, \lceil \ell_n \rceil\right\}$$

- where

$$\ell_i = \begin{cases} 0, & \text{if } \zeta_i^{\circ} = r_i^{\circ} \\ \max\left\{0, log_{|\lambda_i|}\left(\frac{\epsilon_i}{|\zeta_i^{\circ} - r_i^{\circ}|}\right)\right\}, & \text{if } \zeta_i^{\circ} \neq r_i^{\circ} \end{cases}$$

- with

$$\begin{cases} \zeta^{\circ} = V^{-1} x^{\circ} \\ r^{\circ} = \arg\min_{r} \left\{|\zeta^{\circ} - r| : \; |r| \leq \left(I - |\Lambda|\right)^{-1} |V^{-1} E| \bar{\delta}\right\} \end{cases}$$

Maximal positive invariant set

- The control action switches to a fixed feedback law $u_k = Kx_k$ which ensures closed-loop stability and admissibility ($x_k \in \mathcal{X}, u_k \in \mathcal{U}$):

$$\begin{cases} x_{k+1} & = Ax_k + Bu_k \\ x_k & \in \mathcal{X} \\ u_k & \in \mathcal{U} \end{cases} \xRightarrow{u_k = Kx_k} \begin{cases} x_{k+1} & = (A+BK)x_k \\ x_k & \in \mathcal{X} \\ Kx_k & \in \mathcal{U} \end{cases} \xLeftrightarrow[\overline{\mathcal{X}} \leftarrow \mathcal{X} \cap \{x:\, Kx \in \mathcal{U}\}]{A_\circ \leftarrow A+BK} \begin{cases} x_{k+1} & = A_\circ x_k \\ x_k & \in \overline{\mathcal{X}} \end{cases}$$

- The standard recurrence for MPI (maximal positive invariant) set construction is

$$\Omega_0 = \overline{\mathcal{X}}, \quad \Omega_{k+1} = A_\circ^{-1}\Omega_k \bigcap \overline{\mathcal{X}} = \bigcap_{j=0}^{k+1} A_\circ^{-j}\overline{\mathcal{X}}$$

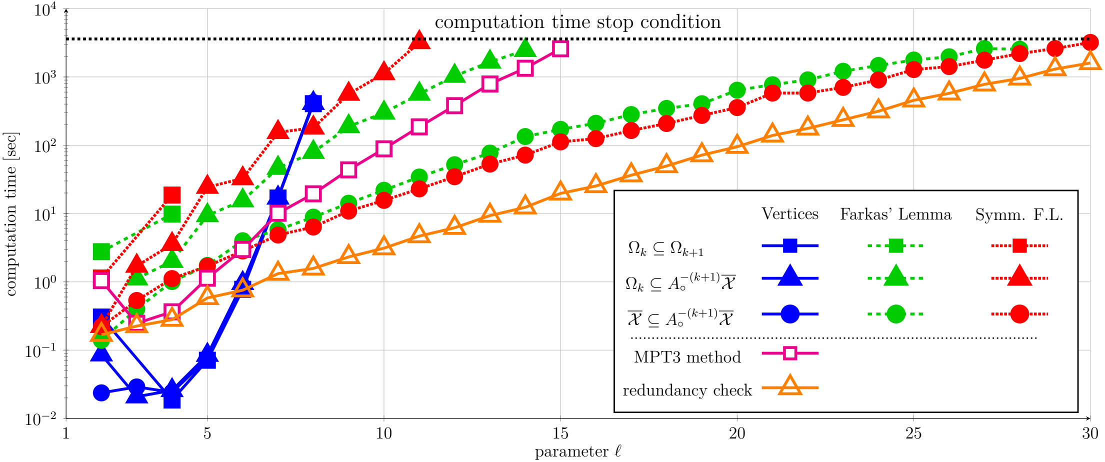
- Under mild and reasonable assumptions, the recurrence is guaranteed to stop by arriving at a fix point $\Omega_{\bar{k}} = \Omega_{\bar{k}+1}$, for some finite index $\bar{k}$.

Stop conditions

Since by construction we have that $\Omega_{k+1} \subseteq \Omega_k$ for all k, it suffices to check that

$$\Omega_k \subseteq \Omega_{k+1} \qquad \Leftrightarrow \Omega_k \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}} \qquad \overline{\mathcal{X}} \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}}$$

Stop conditions (a comparative analysis)



Exploiting symmetry: comparing S8) and S9) we observe that $\dfrac{t_{S8} - t_{S9}}{t_{S8}} \in (13, 46)\%$, with an average of 32%;

Outline

Cell decompositions

- Cell decomposition is a widely used method in robotic motion planning that simplifies complex environments into manageable regions called cells.
- Decompose the robot's configuration space into a collection of non-overlapping, simple regions where planning a path becomes more straightforward.
- The robot can then navigate from one cell to another through adjacent cells (the "lawn-mower" algorithm, the BCD algorithm, etc.)
- Decompositions can be exact (trapezoidal, triangular, cylindrical algebraic decompositions) or approximate (grid-based, octree, BCD).

Voronoi cell decomposition

A cell complex is:

- a set of convex polyhedra
- any two adjacent polyhedra intersect along a common face of both

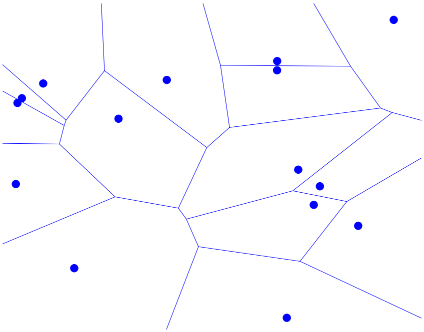
A Voronoi decomposition (look for “Broad Street cholera outbreak”):

- considers a collection of n centers c_i
- to these correspond n Voronoi cells

$$\mathcal{V}(c_i) = \left\{ x \in \mathbb{R}^d :\ d(x, c_i) \leq d(x, c_j),\ \forall j \neq i \right\}$$

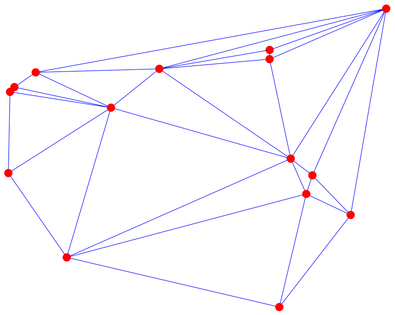
- Equivalently, each cell is a polyhedral set of form

$$\mathcal{V}(c_i) = \left\{ x \in \mathbb{R}^d :\ \left(x - \frac{c_i + c_j}{2} \right)^\top (c_i - c_j) \leq 0,\ \forall j \neq i \right\}$$



Delaunay cell decomposition

- It is a triangularization of the convex hull defined by centers c_i
- such that no circle circumscribed to a triangle contains any other centers
- It is dual to the Voronoi decomposition (there is a bijection between them)
- Computed through the Bowyer–Watson algorithm



Outline

Software tools

Interesting packages:

- polyhedral computations and model predictive control: MPT3 (Matlab)
- many types of convex and non-convex sets (initially for zonotopes): CORA (Matlab)
- interval computations: IBEX (C++) and CODAC (Python) and INTLAB (Matlab)
- many tools gathered in Polymake (Perl overlay over C++ tools, also accessible through incomplete wrappers in Julia and Python)
- polyhedral and ellipsoid manipulations: pycvxset (Python)

Some routines have their own independent implementations (and are used as building blocks in larger packages):

- reverse search enumeration: cdd
- convex hull computations (plus Voronoi and Delaunay decompositions): qhull
- Fourier-Motzkin elimination: pyfme

Relevant resources

- K. Fukuda. “Frequently Asked Questions in Polyhedral Computation”. In: (2022)
- K. Fukuda. “Polyhedral computation”. In: (2020). Publisher: Department of Mathematics, Institute of Theoretical Computer Science ETH Zurich
- F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-17932-2 978-3-319-17933-9. DOI: 10.1007/978-3-319-17933-9
- G. M. Ziegler. *Lectures on polytopes*. Vol. 152. Springer Science & Business Media, 2012
- D. Henrion. “Moments for polynomial optimization - An illustrated tutorial”. In: (2025)