Computing the viability kernel with pattern of cyclic trajectories

Eric Goubault Luc Jaulin Thomas Le Mézo Benjamin Martin
Sylvie Putot

28 June 2016

Problematic

Consider the following nonlinear system:

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned}$$

where $x \in \mathbb{R}^n$ states, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ control. $y \in \mathbb{R}^p$ is some output.

Problematic

Consider the following nonlinear system:

$$\dot{x} = f(x, u)$$
$$y = h(x)$$

where $x \in \mathbb{R}^n$ states, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ control. $y \in \mathbb{R}^p$ is some output.

We want to determine which states are viable with respect to some constraints on the output space.

Viability kernel

Viability Kernel

Given a system f,h and consider a sub-space of the output space $\mathcal{Y}$ and its inverse image in the state space $\mathcal{X} = h^{-1}(\mathcal{Y})$. The viability kernel $\mathrm{Viab}_{f,h}(\mathcal{Y}) \subset \mathcal{X}$ is a sub-space of the state space such that there exists a control u yielding $y = h(x)$ to stay within $\mathcal{Y}$ indefinitely.

Viability kernel

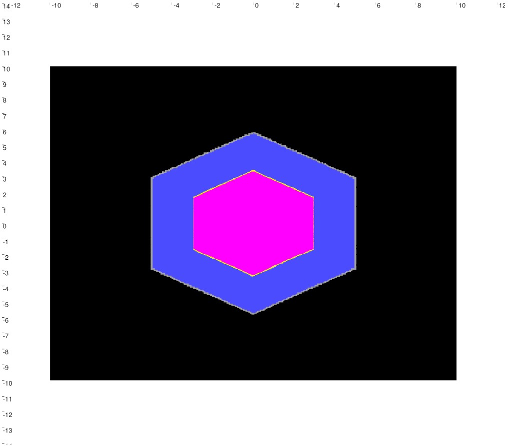
Viability Kernel

Given a system f, h and consider a sub-space of the output space $\mathcal{Y}$ and its inverse image in the state space $\mathcal{X} = h^{-1}(\mathcal{Y})$. The viability kernel $\mathrm{Viab}_{f,h}(\mathcal{Y}) \subset \mathcal{X}$ is a sub-space of the state space such that there exists a control u yielding $y = h(x)$ to stay within $\mathcal{Y}$ indefinitely.

In robotics: y is the pose of the robot and $\mathcal{Y}$ is delimited by obstacles. Which states are safe ?

Illustration

- Forbidden
- Region frontier
- Not viable
- Unkown
- Viable



How to solve ?

TO BE COMPLETED !

How to solve ?

TO BE COMPLETED !

Can be viewed as differential inclusion ($\mathcal{U}$ closed).
• Viability Kernel Algorithm [Saint-Pierre, Math. and Optim., 1994]: discretization of time and state spaces. Iterative approximation of the kernel (smh reachability analysis).

How to solve ?

TO BE COMPLETED !

Can be viewed as differential inclusion ($\mathcal{U}$ closed).
• Viability Kernel Algorithm [Saint-Pierre, Math. and Optim., 1994]: discretization of time and state spaces. Iterative approximation of the kernel (smh reachability analysis).

• Robotics: determining *Inevitable Collision States* from set of evasive trajectories (complement to viability kernel). Typically, breaking trajectories or cyclic (circular) trajectories (see ref. in [Bouguerra et al., ICRA 2015]).
Adaptation of Saint-Pierre algorithm [Bouguerra et al., ICRA 2015].

Our idea

Traditional techniques involves some form of numerical integration which can be costly.

Our idea

Traditional techniques involves some form of numerical integration which can be costly.

Hence, use predefined trajectories through patterns: functions that are zero on the corresponding trajectories.

Our idea

Traditional techniques involves some form of numerical integration which can be costly.

Hence, use predefined trajectories through patterns: functions that are zero on the corresponding trajectories.

- use of cyclic trajectories, and corresponding patterns, which implies viability;

Our idea

Traditional techniques involves some form of numerical integration which can be costly.

Hence, use predefined trajectories through patterns: functions that are zero on the corresponding trajectories.

- use of cyclic trajectories, and corresponding patterns, which implies viability;
- patterns allow to discard "time": the problem can be posed as a CSP.

Cyclic trajectories

Definition

Denote $\phi(x, t, u)$ a feasible trajectory starting from x and with control $u : \mathbb{R} \rightarrow \mathcal{U}$ ($\phi(x, 0, u) = x$). It is cyclic if $\exists \bar{t} > 0$ such that $\phi(x, \bar{t} + t', u) = \phi(x, t', u)$, and $u(\bar{t} + t') = u(t')$, for any $t' \geqslant 0$.

Cyclic trajectories

Definition

Denote $\phi(x, t, u)$ a feasible trajectory starting from x and with control $u : \mathbb{R} \to \mathcal{U}$ ($\phi(x, 0, u) = x$). It is cyclic if $\exists \bar{t} > 0$ such that $\phi(x, \bar{t} + t', u) = \phi(x, t', u)$, and $u(\bar{t} + t') = u(t')$, for any $t' \geqslant 0$.

Main property

If a state x admit a cyclic trajectory ϕ such that $\phi(x, t, u) \in \mathcal{X}, \forall t \leqslant \bar{t}$, then $x \in \mathrm{Viab}_{f,h}(\mathcal{Y})$.

Pattern

Let $V : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$ be a convex function in the state $\times$ output space.

Definition

Given a control $u : \mathbb{R} \rightarrow \mathcal{U}$, V is a pattern for trajectory $\phi(x, t, u)$ if $V(x, h(\phi(x, t, u))) = 0, \forall t$.

Remarks,

Pattern

Let $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ be a convex function in the state $\times$ output space.

Definition

Given a control $u : \mathbb{R} \to \mathcal{U}$, V is a pattern for trajectory $\phi(x, t, u)$ if $V(x, h(\phi(x, t, u))) = 0, \forall t$.

Remarks,

- V can be viewed as a function parametrized by x

Pattern

Let $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ be a convex function in the state $\times$ output space.

Definition

Given a control $u : \mathbb{R} \to \mathcal{U}$, V is a pattern for trajectory $\phi(x, t, u)$ if $V(x, h(\phi(x, t, u))) = 0, \; \forall t$.

Remarks,

- V can be viewed as a function parametrized by x
- relation to control Lyapunov functions

Pattern

Let $V : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$ be a convex function in the state $\times$ output space.

Definition

Given a control $u : \mathbb{R} \rightarrow \mathcal{U}$, V is a pattern for trajectory $\phi(x, t, u)$ if $V(x, h(\phi(x, t, u))) = 0$, $\forall t$.

Remarks,

- V can be viewed as a function parametrized by x
- relation to control Lyapunov functions
- for $p = 2$, if V is a pattern for a cyclic trajectory, w.l.g, $V \leqslant 0$ is the interior of the cycle

Pattern

Let $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ be a convex function in the state $\times$ output space.

Definition

Given a control $u : \mathbb{R} \to \mathcal{U}$, V is a pattern for trajectory $\phi(x, t, u)$ if $V(x, h(\phi(x, t, u))) = 0$, $\forall t$.

Remarks,

- V can be viewed as a function parametrized by x
- relation to control Lyapunov functions
- for $p = 2$, if V is a pattern for a cyclic trajectory, w.l.g, $V \leqslant 0$ is the interior of the cycle
- for larger $p > 2$, several patterns can be used to describe a single trajectory

Pattern and viability

Property

Let V be a pattern function of some cyclic trajectory. Given states x, if $V(x, y) = 0 \implies y \in \mathcal{Y}$, then $x \in \mathrm{Viab}_{f,h}(\mathcal{X})$.

Pattern and viability

Property

Let V be a pattern function of some cyclic trajectory. Given states x, if $V(x,y) = 0 \implies y \in \mathcal{Y}$, then $x \in \mathrm{Viab}_{f,h}(\mathcal{X})$.

Consequence:

$$\mathcal{V} := \{x : \forall y, V(x,y) \neq 0 \vee y \in \mathcal{Y}\} \subseteq \mathrm{Viab}_{f,h}(\mathcal{Y})$$

Pattern and viability

Property

Let V be a pattern function of some cyclic trajectory. Given states x, if $V(x,y) = 0 \implies y \in \mathcal{Y}$, then $x \in \mathrm{Viab}_{f,h}(\mathcal{X})$.

Consequence:

$$\mathcal{V} := \{x : \forall y, V(x,y) \neq 0 \vee y \in \mathcal{Y}\} \subseteq \mathrm{Viab}_{f,h}(\mathcal{Y})$$

Suppose $\mathcal{Y}$ represented as constraints $P(y) \leqslant 0$.

Pattern and viability

Property

Let V be a pattern function of some cyclic trajectory. Given states x, if $V(x,y)=0 \implies y \in \mathcal{Y}$, then $x \in \mathrm{Viab}_{f,h}(\mathcal{X})$.

Consequence:

$$\mathcal{V} := \{x : \forall y, V(x,y) \neq 0 \vee P(y) \leqslant 0\} \subseteq \mathrm{Viab}_{f,h}(\mathcal{Y})$$

Suppose $\mathcal{Y}$ represented as constraints $P(y) \leqslant 0$.

Quantified constraint satisfaction

Example

Dubins car:

$$\left\{\begin{array}{rcl} \dot{x_1} & = & \cos(\theta) \\ \dot{x_2} & = & \sin(\theta) \\ \dot{\theta} & = & u \end{array}\right. ,$$

(position and heading), $u \in [-1, 1]$. We observe the position, i.e. $y_1 = x_1$ and $y_2 = x_2$.

Example

Dubins car:

$$\begin{cases} \dot{x_1} & = & \cos(\theta) \\ \dot{x_2} & = & \sin(\theta) \\ \dot{\theta} & = & u \end{cases},$$

(position and heading), $u \in [-1, 1]$. We observe the position, i.e. $y_1 = x_1$ and $y_2 = x_2$.
We know that for $u \neq 0$ constant, the car follows a cyclic (circular) trajectory:

$$\begin{cases} x_1(t) = \frac{\sin(ut+\theta(0))}{u} - \frac{\sin(\theta(0))}{u} + x_1(0) \\ x_2(t) = -\frac{\cos(ut+\theta(0))}{u} + \frac{\cos(\theta(0))}{u} + x_2(0) \\ \theta(t) = ut + \theta(0) \end{cases}$$

(note: θ modulo 2π)

Example: pattern

We have the following pattern:

$$V(x_1, x_2, \theta, y_1, y_2) := \left(x_1 - \frac{\sin(\theta)}{u} - y_1\right)^2 + \left(x_2 + \frac{\cos(\theta)}{u} - y_2\right)^2 - \frac{1}{u^2} = 0.$$

Example: pattern

We have the following pattern:

$$V(x_1, x_2, \theta, y_1, y_2) := \left(x_1 - \frac{\sin(\theta)}{u} - y_1\right)^2 + \left(x_2 + \frac{\cos(\theta)}{u} - y_2\right)^2 - \frac{1}{u^2} = 0.$$

Note that $(x, \theta) \notin \mathcal{X}(\equiv \mathcal{Y}) \implies (x, \theta) \notin \mathrm{Viab}_{f,h}(\mathcal{Y})$. $P(x_1, x_2) \leqslant 0$ is necessary for $(x, \theta) \in \mathrm{Viab}_{f,h}(\mathcal{Y})$.

Example: pattern

We have the following pattern:

$$V(x_1, x_2, \theta, y_1, y_2) := \left(x_1 - \frac{\sin(\theta)}{u} - y_1\right)^2 + \left(x_2 + \frac{\cos(\theta)}{u} - y_2\right)^2 - \frac{1}{u^2} = 0.$$

Note that $(x, \theta) \notin \mathcal{X}(\equiv \mathcal{Y}) \implies (x, \theta) \notin \mathrm{Viab}_{f,h}(\mathcal{Y})$. $P(x_1, x_2) \leqslant 0$ is necessary for $(x, \theta) \in \mathrm{Viab}_{f,h}(\mathcal{Y})$.

Checking $x \in \mathcal{V}$ requires to check $V(x_1, x_2, \theta, y_1, y_2) \neq 0$ for all $y, P(y) > 0$.

Example: pattern

We have the following pattern:

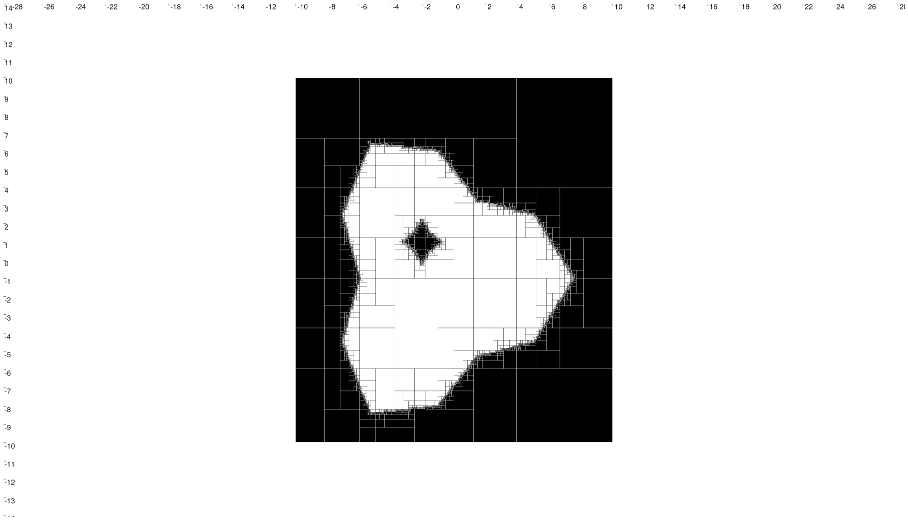
$$V(x_1, x_2, \theta, y_1, y_2) := \left(x_1 - \frac{\sin(\theta)}{u} - y_1\right)^2 + \left(x_2 + \frac{\cos(\theta)}{u} - y_2\right)^2 - \frac{1}{u^2} = 0.$$

Note that $(x, \theta) \notin \mathcal{X}(\equiv \mathcal{Y}) \implies (x, \theta) \notin \mathrm{Viab}_{f,h}(\mathcal{Y})$. $P(x_1, x_2) \leqslant 0$ is necessary for $(x, \theta) \in \mathrm{Viab}_{f,h}(\mathcal{Y})$.

Checking $x \in \mathcal{V}$ requires to check $V(x_1, x_2, \theta, y_1, y_2) \neq 0$ for all $y, P(y) > 0$.

$\Rightarrow$ Build a paving of $\mathcal{Y}$. Inner boxes: initial conditions; boundary and outer boxes: domains of y.

Example: solving



Example: paving of $\mathcal{Y}$.

Example: solving

About the heading $\theta \in [-\pi, \pi]$?

Example: solving

About the heading $\theta \in [-\pi, \pi]$?
For the sake of 2D representation: projection by quantifying θ.

- $\approx \exists$ by sampling the domain of θ (here 8 subintervals),
- $\forall\ \theta$ in its domain.

Example: solving

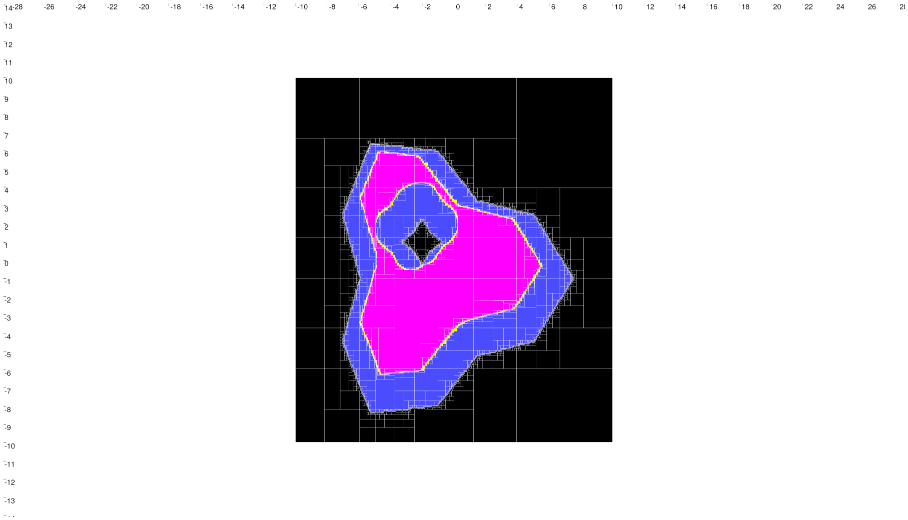
About the heading $\theta \in [-\pi, \pi]$?
For the sake of 2D representation: projection by quantifying θ.

- $\approx \exists$ by sampling the domain of θ (here 8 subintervals),
- $\forall\ \theta$ in its domain.

Other remarks:

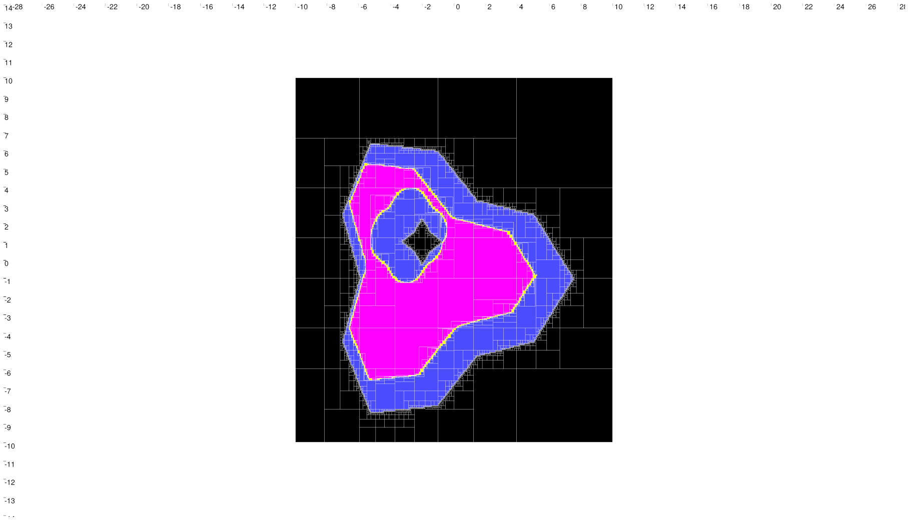
- implementation in Ibex, using a separator on the quantified constraint,
- for convenience: consider $V \leqslant 0$ instead of $V = 0$,
- Cpu times on this instance: 155 s for the sampled version, 35 s on the original domain.

Example: computation



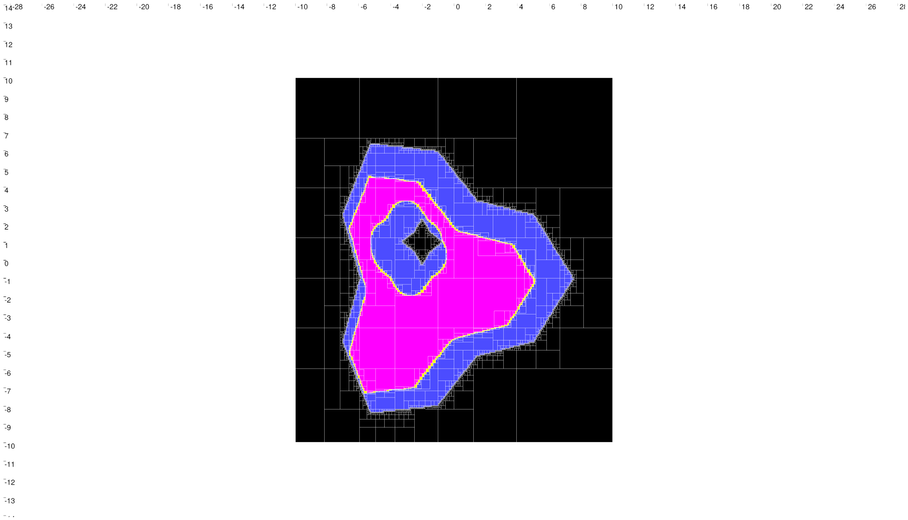
$$\theta \in [-\pi, -\frac{3}{4}\pi]$$

Example: computation



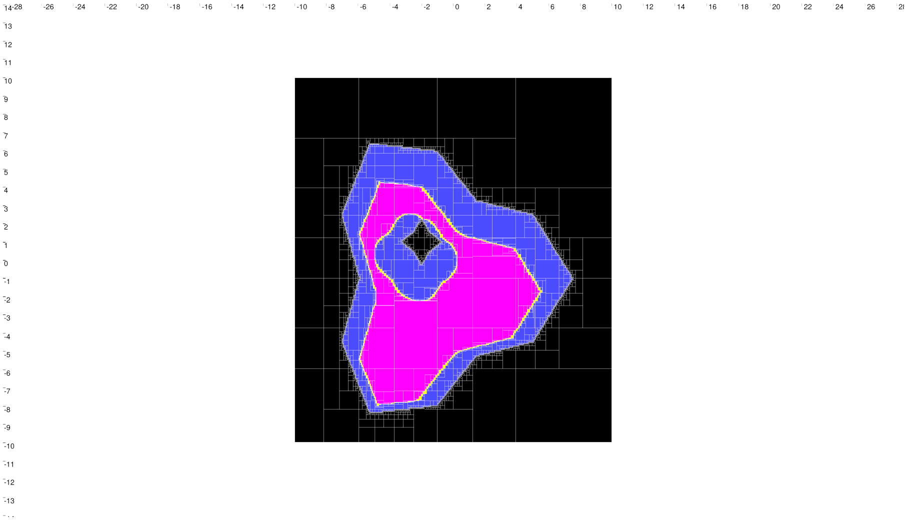
$$\theta \in [-\frac{3}{4}\pi, -\frac{1}{2}\pi]$$

Example: computation



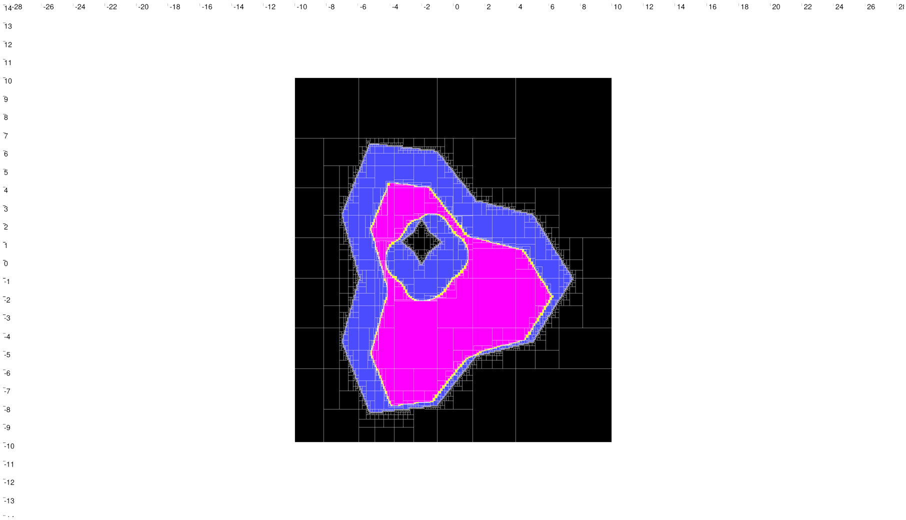
$$\theta \in [-\frac{1}{2}\pi, -\frac{1}{4}\pi]$$

Example: computation



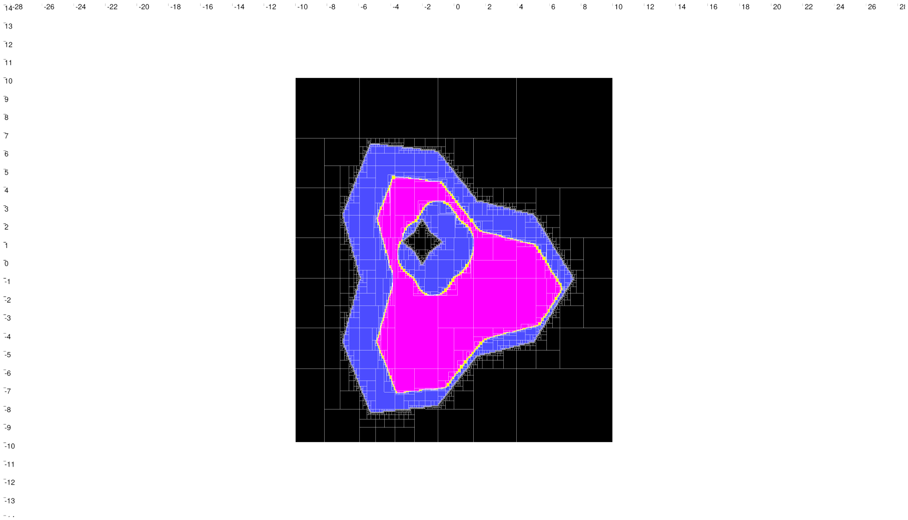
$$\theta \in [-\tfrac{1}{4}\pi, 0]$$

Example: computation



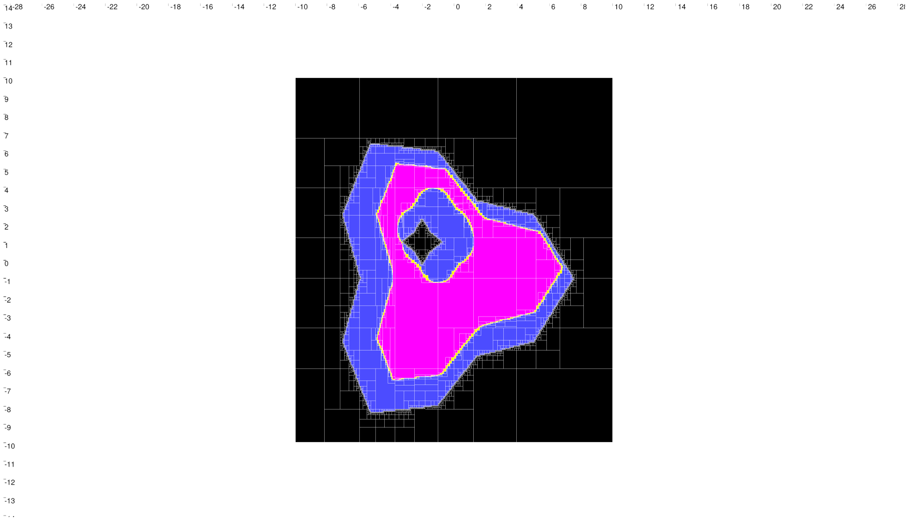
$$\theta \in [0, \frac{1}{4}\pi]$$

Example: computation



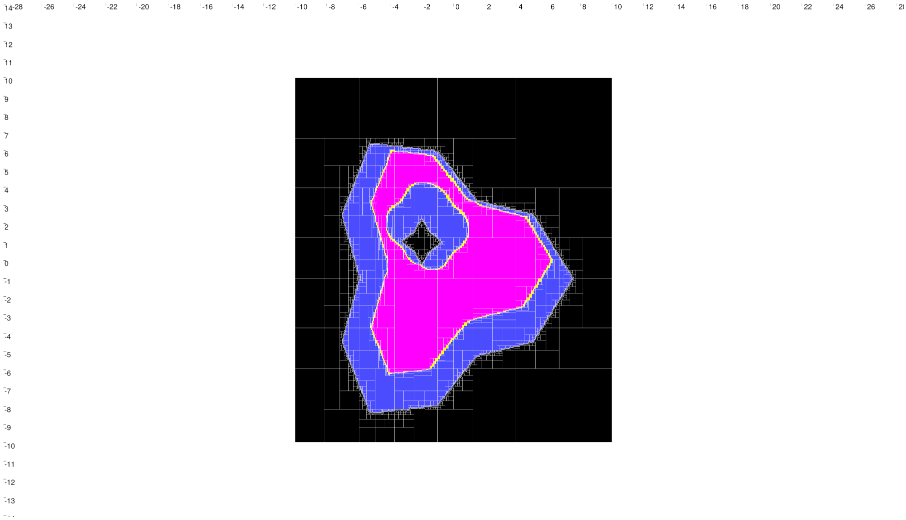
$$\theta \in [\frac{1}{4}\pi, \frac{1}{2}\pi]$$

Example: computation



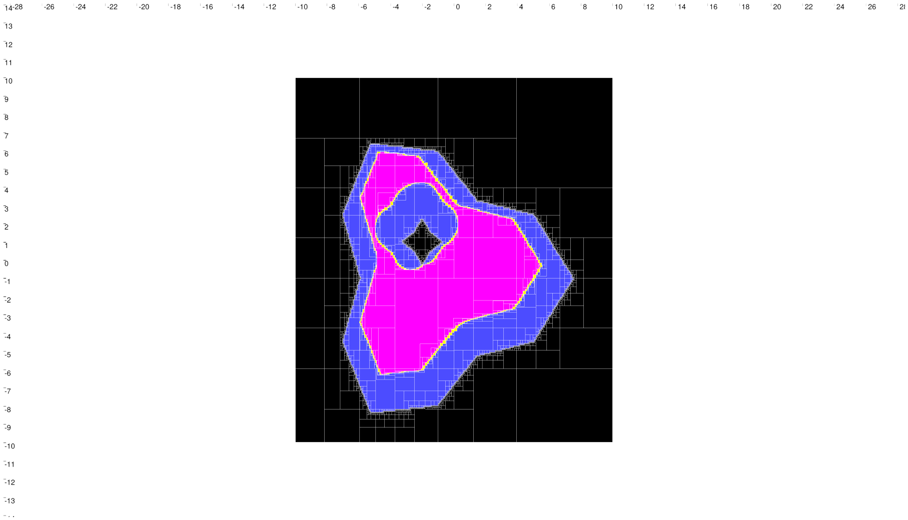
$$\theta \in [\tfrac{1}{2}\pi, \tfrac{3}{4}\pi]$$

Example: computation



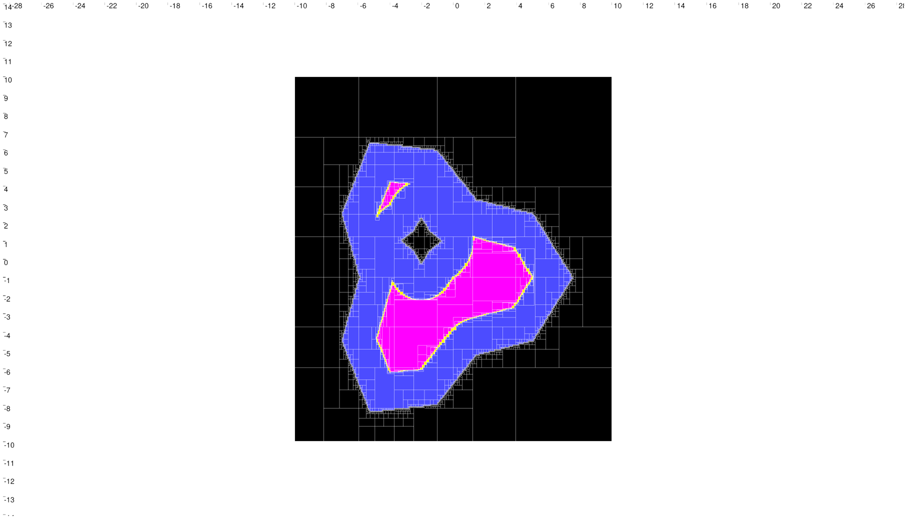
$$\theta \in [\frac{3}{4}\pi, \pi]$$

Example: computation



$$\theta \in [-\pi, -\frac{3}{4}\pi]$$

Example: computation



$$\theta \in [-\pi, \pi]$$

Discussion

- Computation times are reasonable...

Discussion

- Computation times are reasonable...
- ... but only inner approximation of $\mathrm{Viab}_{f,h}(\mathcal{Y})$.

Discussion

- Computation times are reasonable...
- ... but only inner approximation of $\mathrm{Viab}_{f,h}(\mathcal{Y})$.
- Extension / other version with verified integration ?

Discussion

- Computation times are reasonable...
- ... but only inner approximation of $\mathrm{Viab}_{f,h}(\mathcal{Y})$.
- Extension / other version with verified integration ?
- Generating V ?

Generating V ? An example

On the previous example:

$$V := (x_1 - c_1)^2 + (x_2 - c_2)^2 - d^2,$$

where c_1, c_2 and d are parameters depending on initial conditions of the states. In particular the flow satisfy $V = 0$.

Generating V ? An example

On the previous example:

$$V := (x_1 - c_1)^2 + (x_2 - c_2)^2 - d^2,$$

where c_1, c_2 and d are parameters depending on initial conditions of the states. In particular the flow satisfy $V = 0$.
Hence,

$$\dot{V} = 2g(x_1 - c_1) + 2h(x_2 - c_2) = 0,$$

where $g = \cos(\theta), h = \sin(\theta)$.

Generating V ? An example

On the previous example:

$$V := (x_1 - c_1)^2 + (x_2 - c_2)^2 - d^2,$$

where c_1, c_2 and d are parameters depending on initial conditions of the states. In particular the flow satisfy $V = 0$.
Hence,

$$\dot{V} = 2g(x_1 - c_1) + 2h(x_2 - c_2) = 0,$$

where $g = \cos(\theta), h = \sin(\theta)$.
Going further:

$$\ddot{V} = u\left(2g(x_2 - c_2) - 2h(x_1 - c_1)\right) + 2 = 0,$$

entailing $u = -2/\left(2g(x_2 - c_2) - 2h(x_1 - c_1)\right)$

Generating V ? An example

This control ensure the non variation of $\dot{V}$. We can note $\dot{u} = 0$ (due to $\dot{V} = 0$). It can be fixed to a $\tilde{u} \neq 0$ in $\mathcal{U}$.

Generating V ? An example

This control ensure the non variation of $\dot{V}$. We can note $\dot{u} = 0$ (due to $\dot{V} = 0$). It can be fixed to a $\tilde{u} \neq 0$ in $\mathcal{U}$.

Hence, the following system (where x_1, x_2, θ must be viewed as initial conditions).

$$\begin{cases} (x_1 - c_1)^2 + (x_2 - c_2)^2 - d^2 & = 0 \\ 2g(x_1 - c_1) + 2h(x_2 - c_2) & = 0 \\ \tilde{u}\,(2g(x_2 - c_2) - 2h(x_1 - c_1)) + 2 & = 0 \end{cases}$$

Generating V ? An example

This control ensure the non variation of $\dot{V}$. We can note $\dot{u} = 0$ (due to $\dot{V} = 0$). It can be fixed to a $\tilde{u} \neq 0$ in $\mathcal{U}$.

Hence, the following system (where x_1, x_2, θ must be viewed as initial conditions).

$$\begin{cases} (x_1 - c_1)^2 + (x_2 - c_2)^2 - d^2 & = 0 \\ 2g(x_1 - c_1) + 2h(x_2 - c_2) & = 0 \\ \tilde{u}\,(2g(x_2 - c_2) - 2h(x_1 - c_1)) + 2 & = 0 \end{cases}$$

Solution: $c_1 = x_1 - \frac{\sin(\theta)}{\tilde{u}}$, $c_2 = x_2 + \frac{\cos(\theta)}{\tilde{u}}$, $d = \frac{1}{\tilde{u}}$.

Generalization ?

Some hints:

- consider specific class of parametric functions (with a "cyclic" level curve).

Generalization ?

Some hints:

- consider specific class of parametric functions (with a "cyclic" level curve).
- find appropriate control law

Generalization ?

Some hints:

- consider specific class of parametric functions (with a "cyclic" level curve).
- find appropriate control law
- in particular, it must satisfy: $u \in \mathcal{U}$

Generalization ?

Some hints:

- consider specific class of parametric functions (with a "cyclic" level curve).
- find appropriate control law
- in particular, it must satisfy: $u \in \mathcal{U}$

Smarter ways, E.g. starting from given "time" cyclic trajectories ?

Summary

An interval constraint-based approach for computing inner approximation of viability kernels, based on pattern of cyclic trajectories.

Summary

An interval constraint-based approach for computing inner approximation of viability kernels, based on pattern of cyclic trajectories.

- allows contraction of domains, i.e. captures accurately e.g. the boundary of the approximation,
- promising performances...

Summary

An interval constraint-based approach for computing inner approximation of viability kernels, based on pattern of cyclic trajectories.

- allows contraction of domains, i.e. captures accurately e.g. the boundary of the approximation,
- promising performances...
- ... but needs a comparison with other approaches (e.g. based on verified integration)