Invariant Set and Controller Synthesis

FEANICSES 2018 Workshop

Danylo Malyuta, Dylan Janak, Behçet Açıkmeşe

May 19, 2018

Autonomous Controls Laboratory, University of Washington

Overview

Invariant Set Synthesis

Optimization-Based Methods

Set-Theoretic Methods

Invariant Controller Synthesis

Robust Controlled Invariant Set

We consider Discrete Linear Time Invariant (DLTI) system:

$$x^+ = Ax + Bu + Dp,$$

where $x \in \mathbf{R}^n$, $p \in \mathcal{P} = \{p \in \mathbf{R}^d : Rp \leq r\}$ and
$u \in \mathcal{U} = \{u \in \mathbf{R}^m : Hu \leq h\}$ are "specification" polytopes.

Controlled Robust Positively Invariant Set

A set $\mathcal{X}$ is called *controlled robust positively invariant* (CRPI) if:

$$\mathcal{X} = \{x \in \mathbf{R}^n : \exists u \in \mathcal{U} \text{ s.t. } Ax + Bu + Dp \in \mathcal{X},\ \forall p \in \mathcal{P}\}.$$

Robust Invariant Set

Robust Controlled Invariant Set

A set $\mathcal{X}$ is called *controlled robust positively invariant* (CRPI) if:

$$\mathcal{X} = \{x \in \mathbf{R}^n : \exists u \in \mathcal{U} \text{ s.t. } Ax + Bu + Dp \in \mathcal{X},\ \forall p \in \mathcal{P}\}.$$

Now consider that some control law exists and the system reduces to an autonomous one:

$$x^+ = Ax + Dp.$$

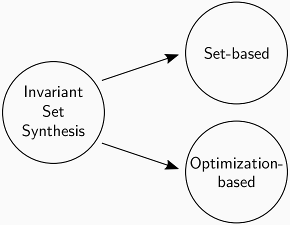
Robust Invariant Set

A set $\mathcal{X}$ is called *robust positively invariant* (RPI) if:

$$Ax + Dp \in \mathcal{X}, \quad \forall x \in \mathcal{X},\ p \in \mathcal{P}.$$

Goal: find an RPI $\mathcal{X}$.

Two Ways to Synthesize an Invariant Set



- Optimization-based methods rely on an explicit optimization problem (LP, LMI, etc.) to find $\mathcal{X}$
- Set-based methods rely on polytopic operations[1], i.e. computational geometry.

[1] These operations may implicitly involve an optimization, but what differentiates set-based methods is that people don't "talk" about it – they just assume that one can compute e.g. the Pontryagin difference.

One-Step Minimal RPI Computation

Trodden, “A One-Step Approach to Computing a Polytopic Robust Invariant Set”, 2016. [1]

Equivalent RPI Condition

$$\mathcal{X}(g) = \{x : Gx \leq g\} \text{ RPI} \Leftrightarrow \sigma(G_i \mid A\mathcal{X}(g)) + \sigma(G_i \mid \mathcal{P}) \leq \sigma(G_i \mid \mathcal{X}(g)),$$

where $g \in \mathbf{R}^{n_g}$ and $\sigma(z \mid \mathcal{S}) \triangleq \sup\{y^T z : y \in \mathcal{S}\}$ is the support function of (some) set $\mathcal{S}$.

Note: $\sigma(G_i \mid \mathcal{X}(g)) \leq g_i$ with $< \Leftrightarrow$ facet i is redundant.

One-Step Minimal RPI Computation

Trodden, "A One-Step Approach to Computing a Polytopic Robust Invariant Set", 2016. [1]

Existence of an RPI Set

Fix G in $\mathcal{X}(g) = \{x : Gx \leq g\}$ (i.e. pick a "template"). Assumptions:

A1. $\mathcal{P}$ contains the origin

A2. $\lambda < 0 \ \forall \lambda \in \mathrm{spec}(A)$

A3. The interior of $\mathcal{X}$ contains the origin

A4. For the chosen G, a g exists such that $\mathcal{X}(g)$ is RPI

Then there exists a g^* such that

$$\sigma(G_i \mid A\mathcal{X}(g^*)) + \sigma(G_i \mid \mathcal{P}) = \sigma(G_i \mid \mathcal{X}(g^*)) = g^* \quad \forall i = 1, ..., n_g.$$

$\mathcal{X}^*$ is the minimum-volume RPI set, i.e. g^* achieves minimum $\|g^*\|_1$.

Fixed-Point Solution Uniqueness

Given assumptions A1-A4, the g^* in the above statement is unique.

One-Step Minimal RPI Computation

Trodden, "A One-Step Approach to Computing a Polytopic Robust Invariant Set", 2016. [1]

Existence of an RPI Set

$$\sigma(G_i \mid A\mathcal{X}(g^*)) + \sigma(G_i \mid \mathcal{P}) = \sigma(G_i \mid \mathcal{X}(g^*)) = g^* \quad \forall i = 1, \ldots, n_g.$$

$\mathcal{X}^*$ is the minimum-volume RPI set, i.e. g^* achieves minimum $\|g^*\|_1$.

g^* can be computed iteratively:

Algorithm 1 Iterative computation of g^*.

```
Set g ← 0
while True do
    g*_i ← σ(G_i | AX(g)) + σ(G_i | P)  i = 1, ..., n_g
    if ||g − g*||_∞ < ε_tol then
        return g*
    g ← g*
```

First Way: Optimization

The control problem can be formulated as an optimization problem:

Control Policy Synthesis via Optimization

Let $\mathcal{I} \triangleq \{x \in \mathbf{R}^n \mid Gx \leq g\}$ be the maximal positively invariant set induced by the control policy $u[k] = \mu_k(z[k])$. Consider the following sequence of optimization problems (for $i = 1, ..., n_\mathrm{p}$):

$$
\begin{aligned}
g_i^+ = \underset{x,u,w,v,e,G,g,k}{\text{maximize}} \quad & G_i(A[k]x + B[k](u+e) + E[k]w) \\
\text{subject to} \quad & x \in \mathcal{I},\ u \in \mathcal{U},\ w \in \mathcal{W}(x,u),\ v \in \mathcal{V}(x),\ e \in \mathcal{L}(u) \\
& u = c_k(x+v) \\
& \mathcal{I} \subseteq \mathcal{X}, \quad k \in \mathbf{Z}_+.
\end{aligned}
$$

The control policy solves the control problem if and only if $g^+ \leq g$.

Authors employing optimization solve this problem via clever tricks for the particular structure that they consider (yields an LP, an SDP, etc.).

Second Way: Set-Based Iterative

Predecessor Set

Given a set $\mathcal{R} \subseteq \mathcal{X}$, the *predecessor set* $\text{Pre}(\mathcal{R})$ is:

$$\text{Pre}(\mathcal{R}) \triangleq \{x \in \mathbf{R}^n \mid \exists u \in \mathcal{U} \text{ s.t. } A[k]x + B[k](u+v) + E[k]w \in \mathcal{R} \\ \forall v \in \mathcal{V}(x), w \in \mathcal{W}(x,u)\},$$

i.e. $\mathcal{R}$ is 1-step robustly *reachable* from $\text{Pre}(\mathcal{R})$.

Consider the algorithm:

$$\mathcal{I}_0 = \mathcal{X}, \quad \mathcal{I}_{k+1} = \text{Pre}(\mathcal{I}_k).$$

Then $\mathcal{I}_{k+1} \subseteq \mathcal{I}_k \ \forall i \in \mathbf{Z}_+$ and the *maximal robust controlled invariant* set in $\mathcal{X}$ is $\mathcal{I}_\infty \subseteq \bigcap_{i \in \mathbf{Z}_+} \mathcal{I}_i$ and $\mathcal{I}_\infty = \mathcal{I}_j$ for some $j \in \mathbf{Z}_+ \Leftrightarrow \mathcal{I}_{j+1} = \mathcal{I}_j$.

The resulting control policy is set valued and is obtained a posteriori:

$$c_k(x) = \{u \in \mathcal{U} \mid A[k]x + B[k](u+v) + E[k]w \in \mathcal{I}_\infty \ \forall v \in \mathcal{V}(x), w \in \mathcal{W}(x,u)\}.$$

Can then use e.g. dynamic programming to obtain some optimal point-valued policy.

Comparison of Set-Based versus Optimization-Based

- Optimization-based methods are faster and compute point-valued controllers directly
- Set-based methods are slower but can potentially accommodate more features and can be *anytime* (i.e. aborted at any point and yield a valid albeit imprecise answer anyway)
- Set-based methods compute set-valued controllers $\Rightarrow$ post-processing (e.g. dynamic programming) required to obtain point-valued controllers.

Whether one or the other will solve all our problems remains to be seen as we *actually try to solve all our problems*.

Overview

Overview

- Present “the control problem”
- LQR, Linear From Spec, Bertsekas, perhaps other new ones...

Bibliography

[1] P. Trodden, "A one-step approach to computing a polytopic robust positively invariant set," *IEEE Transactions on Automatic Control*, vol. 61, pp. 4100–4105, dec 2016.