Invariant Set and Controller Synthesis

FEANICSES 2018 Workshop

Danylo Malyuta, Dylan Janak, Behçet Açıkmeşe

May 21, 2018

Autonomous Controls Laboratory, University of Washington

Overview

Robust Controlled Invariant Set

We consider Discrete Linear Time Invariant (DLTI) system:

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

where $x \in \mathbf{R}^n$, $p \in \mathcal{P} = \{p \in \mathbf{R}^d : Rp \le r\}$ and $u \in \mathcal{U} = \{u \in \mathbf{R}^m : Hu \le h\}$ 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:

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

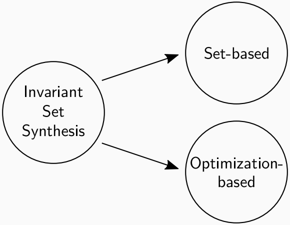
Robust Invariant Set

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

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

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

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

$\mathcal{X}(g^*)$ is the min-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

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

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

g^* can be computed iteratively:

Algorithm 1 Iterative computation of g^*.

1: Set $g \leftarrow 0$
2: **while** True **do**
3: $\quad g_i^* \leftarrow \sigma(G_i \mid A\mathcal{X}(g)) + \sigma(G_i \mid D\mathcal{P})\ i = 1, \ldots, n_g$
4: $\quad$ **if** $\|g - g^*\|_\infty < \epsilon_{\text{tol}}$ **then**
5: $\quad\quad$ **return** g^*
6: $\quad g \leftarrow g^*$

One-Step Minimal RPI Computation

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

- g^* can also be computed as a one-shot LP (main contribution of [1])
- Let $c_i(g) = \sigma(G_i \mid A\mathcal{X}(g))$, $d_i = \sigma(G_i \mid D\mathcal{P})$, $b_i(g) = \sigma(G_i \mid \mathcal{X}(g))$. Core realization (thanks to uniqueness of g^*):

$$g^* = \arg\min_{g}\{\|g\|_1 : c(g)+d = b(g)\} = \arg\max_{g}\{\|g\|_1 : c(g)+d = b(g)\}$$

- Recalling that $b(g) \le g$, the above is readily converted to an LP:

$$\begin{aligned} g^* = c^* + d^*, \text{ where } (c^*, d^*) = \; & \underset{\substack{\{c_i, d_i, \xi^i, \omega^i\} \\ \forall i \in \{1,\dots,n_g\}}}{\arg\text{ maximize}} && \sum_{i=1}^{n_g} c_i + d_i \\ & \text{subject to} && c_i \le G_i A \xi^i \\ & && G\xi^i \le c + d \\ & && d_i \le G_i D \omega^i \\ & && F\omega^i \le g. \end{aligned}$$

One-Step Minimal RPI Computation

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

$$\boxed{\text{Let } c_i(g) = \sigma(G_i \mid A\mathcal{X}(g)),\ d_i = \sigma(G_i \mid D\mathcal{P}),\ b_i(g) = \sigma(G_i \mid \mathcal{X}(g))}$$

$$
\begin{aligned}
g^* = c^* + d^*,\ \text{where } (c^*, d^*) = \arg \underset{\substack{\{c_i, d_i, \xi^i, \omega^i\} \\ \forall i \in \{1,\ldots,n_g\}}}{\text{maximize}} \quad & \sum_{i=1}^{n_g} c_i + d_i \\
\text{subject to} \quad & c_i \le G_i A \xi^i \\
& G\xi^i \le c + d \\
& d_i \le G_i D \omega^i \\
& F\omega^i \le g.
\end{aligned}
$$

The first two constraints evaluate $c_i(g)$ and the last two evaluate d_i. The first constraint holds with equality at optimality, since we want to maximize c_i. The RHS of the second constraint $= g^*$ at optimality, therefore the second constraint enforces $P\xi^i \le g^*$, i.e. the definition of $b(g^*)$.

One-Step Minimal RPI Computation

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

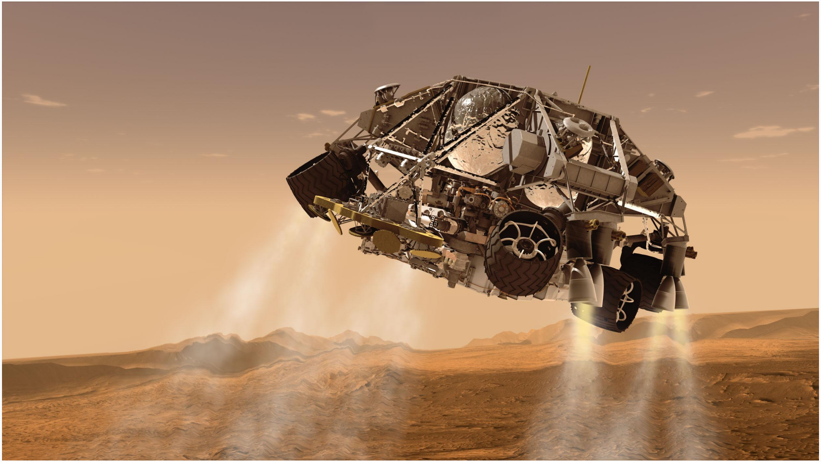
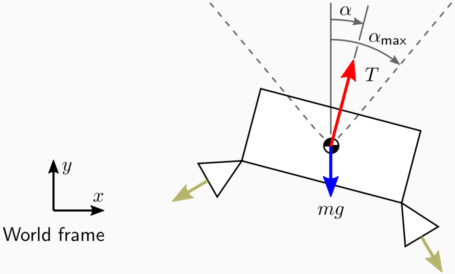


Image credit: NASA/JPL-Caltech

One-Step Minimal RPI Computation

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



Parameters [2]:

$$m_{\text{wet}} = 1905 \text{ kg}$$
$$g = -3.7114 \text{ m/s}^2$$
$$g_e = 9.81 \text{ m/s}^2$$
$$I_{sp} = 225 \text{ s} \quad T_{\max} = 3.1 \text{ kN}$$
$$\phi = 27 \text{ deg} \quad n = 6$$

Dynamics:

$$(\dot{x}, \dot{y}) = (v_x, v_y)$$
$$(\dot{v}_x, \dot{v}_y) = (T_x, T_y)/m + g$$
$$\dot{m} = -\frac{\|(T_x, T_y)\|_2}{I_{sp} g_e \cos\phi}$$

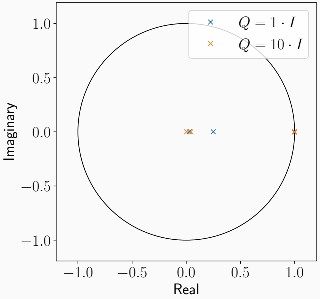
Letting $T \leftarrow T + mg$ be the gravity compensated control, the system is linearized about $(\bar{x}, \bar{y}, \bar{v}_x, \bar{v}_y, \bar{m}) = (0, 0, 0, 0, m_{\text{wet}})$ and $(\bar{T}_x, \bar{T}_y) = (0, 0)$.

One-Step Minimal RPI Computation

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

Synthesize an LQR stabilizing controller:

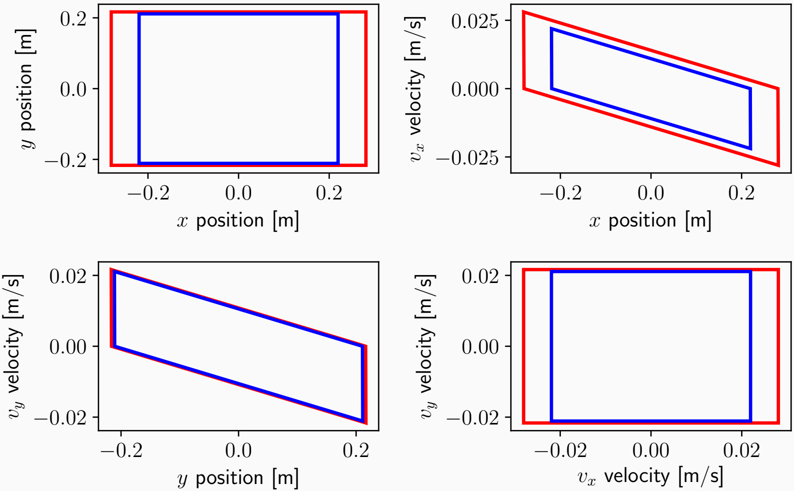
- State scaling: $D_x = \begin{bmatrix} 1 & 1 & 0.05 & 0.05 & 0.1 \end{bmatrix}$
- Input scaling: $D_u = \begin{bmatrix} nT_{\max} \cos\phi \sin\alpha_{\max} & nT_{\max} \cos\phi \end{bmatrix}$
- State penalty $Q = D_x^{-1} \hat{Q} D_x$ with $\hat{Q} \in \{I_5, 10I_5\}$
- Input penalty $R = D_x^{-1} \hat{R} D_x$ with $\hat{R} = I_2$



One-Step Minimal RPI Computation

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

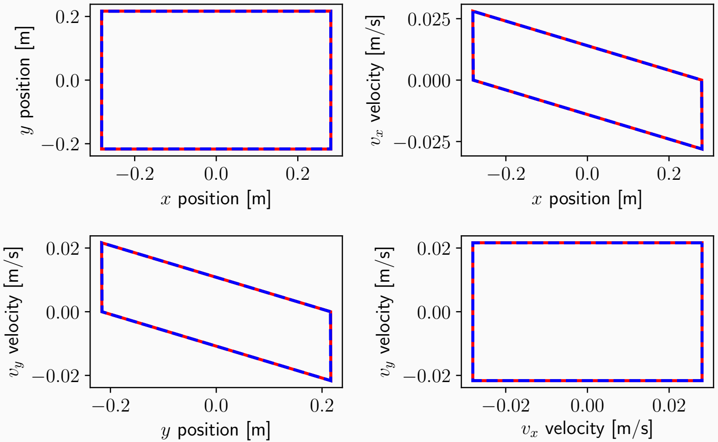
Direct application of LP on slide 10 ($\hat{Q} = I_5$, $\hat{Q} = 10I_5$):



One-Step Minimal RPI Computation

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

The one-shot LP of slide 10 and the iterative algorithm of slide 8 are identical...



... but iterative takes ≈ 315 s while one-shot takes ≈ 0.2 s!

Maximal RCI Computation

Kvasnica et al., "Reachability Analysis and Control Synthesis for Uncertain Linear Systems...", 2015. [3]

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 $i = 1, \ldots, n_\mathrm{p}$):

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

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.

[2] B. Acikmese and S. R. Ploen, “Convex programming approach to powered descent guidance for mars landing,” *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 1353–1366, sep 2007.

[3] M. Kvasnica, B. Takács, J. Holaza, and D. Ingole, “Reachability analysis and control synthesis for uncertain linear systems in MPT,” *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 302–307, 2015.