

Computing ellipsoidal controlled invariant sets for stochastic programming

Benoît Legat and Raphaël Jungers (UCLouvain)

October 30, 2026

Relatively Complete Recourse

Controlled Invariant Sets

Hybrid Systems

Conclusion

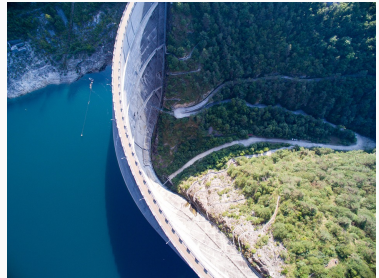
Relatively Complete Recourse

Motivation : Water Reservoir

- $h_{\min} \leq h \leq h_{\max}$: water height [m]
- $v_{\min} \leq v \leq v_{\max}$: height increase rate [m s^{-1}]
- $u_{\min} \leq u \leq u_{\max}$: v increase rate [m s^{-2}]

$$h_{k+1} = h_k + v_k \Delta t$$

$$v_{k+1} = v_k + u_k \Delta t$$



Relatively complete recourse

Risks of infeasibility

- $h > h_{\max}$: Water spills
- $h < h_{\min}$: Risk of electricity shortage

Bounds on u prevents last-minute actions.

No relatively complete recourse!

Shortage is catastrophic $\rightarrow$ need huge penalty $\rightarrow$ huge Lipschitz constant $L \rightarrow$ huge sample size

$$N \geq \mathcal{O}(L^2 \log(L) \cdots)$$

Shapiro and Nemirovski, *On complexity of stochastic programming problems*,
Continuous optimization, 2005

Controlled Invariant Sets

Controlled invariant set

Linear discrete control system

$$x_{k+1} = Ax_k + Bu_k, \quad x_k \in X, u_k \in U$$

Controlled invariant set

Set $S \subseteq X$ *controlled invariant* iff

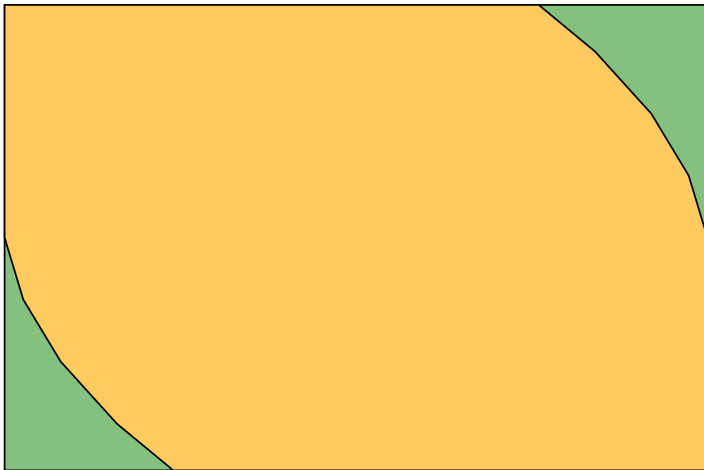
$$\forall x \in S, \exists u \in U \text{ such that } Ax + Bu \in S$$

Safe Stochastic Programming

- Compute controlled invariant set $S \subseteq$ safe set.
- Add constraint: state $\in S$.

Analytical controlled invariant set

Symmetrization: Assume $v_{\min} = -v_{\max}$ and $u_{\min} = -u_{\max}$ for simplicity. Assume $h_{\min} = -h_{\max}$ wlog.



Representation of polyhedral sets

$\mathcal{H}$ -representation

Representation of polyhedron S as intersection of halfspaces:

$$S = \bigcap_{i=1}^m \{x \mid a_i^\top x \leq \beta_i\}.$$

- Computing $A^{-1}S$ is **easy**: $a_i \rightarrow A^\top a_i$.
- Projection is **hard**. Algorithms: Fourier-Motzkin, ...

$\mathcal{V}$ -representation

Representation of polyhedron S as hull of points/rays:

$$S = \text{convexhull}(x_1, \dots, x_{m_x}) + \text{conichull}(r_1, \dots, r_{m_r}).$$

- Computing AS is **easy**: $x \rightarrow Ax$, $r \rightarrow Ar$.
- Projection is **easy**: project each point/ray.

Representation conversion is **hard**. Algorithms: double description, quick hull, ...

Unconstrained input without loss of generality

Constrained input

$$x_{k+1} = Ax_k + Bu_k, \quad x_k \in X, u_k \in U$$

Lift to unconstrained input: $x' = (x, u)$

$$x_{k+1} = \begin{pmatrix} A & B \end{pmatrix} x'_k$$

$$u_{k+1} = u'_k$$

where $x'_k \in X' = X \times U$.

Computation of invariant set

Linear discrete system : $x_{k+1} = Ax_k$, $x_k \in X$.

Pre-image

$$A^{-1}S \triangleq \{x \mid Ax \in S\}$$

Invariant set

Set $S \subseteq X$ *invariant* iff

$$\forall x \in S, Ax \in S \quad \Leftrightarrow \quad AS \subseteq S \quad \Leftrightarrow \quad S \subseteq A^{-1}S$$

Computation of invariant set

Linear discrete system : $x_{k+1} = Ax_k$, $x_k \in X$.

Pre-image

$$A^{-1}S \triangleq \{x \mid Ax \in S\}$$

Invariant set

Set $S \subseteq X$ *invariant* iff

$$\forall x \in S, Ax \in S \quad \Leftrightarrow \quad AS \subseteq S \quad \Leftrightarrow \quad S \subseteq A^{-1}S$$

Polyhedral computation

$$P_0 = X$$

$$P_{k+1} = P_k \cap A^{-1}P_k$$

A^{-1} easy with $\mathcal{H}$ -representation

$\cap$ easy with $\mathcal{H}$ -representation

Computation of invariant set

Linear discrete system : $x_{k+1} = Ax_k$, $x_k \in X$.

Pre-image

$$A^{-1}S \triangleq \{x \mid Ax \in S\}$$

Invariant set

Set $S \subseteq X$ *invariant* iff

$$\forall x \in S, Ax \in S \quad \Leftrightarrow \quad AS \subseteq S \quad \Leftrightarrow \quad S \subseteq A^{-1}S$$

Ellipsoidal computation

$$C = \{x \mid x^T Q x \leq 1\} \quad \Rightarrow \quad A^{-1}C = \{x \mid x^T A^T Q A x \leq 1\}$$

S-procedure:

$$Q - A^T Q A \text{ is PSD.}$$

$$P_0 = X$$

$$\begin{aligned} P_{k+1} &= P_k \cap \text{proj}_x \{ (x, u) \mid Ax + Bu \in P_k \} \\ &= P_k \cap \text{proj}_x [A, B]^{-1} P_k \end{aligned}$$

Not scalable...

$[A, B]^{-1}$	easy with $\mathcal{H}$ -representation
proj_x	easy with $\mathcal{V}$ -representation
$\cap$	easy with $\mathcal{H}$ -representation

Unconstrained control to algebraic system

Linear Discrete **Control** System (DT**CS**)

$$x_{k+1} = Ax_k + Bu_k, \quad x_k \in X$$

Linear Discrete **Algebraic** System (DT**AS**)

$$Ex_{k+1} = Ax_k, \quad x_k \in X$$

Invariant Set $S \subseteq X$ if

$$\forall x \in S, \exists y \in S : Ey = Ax$$

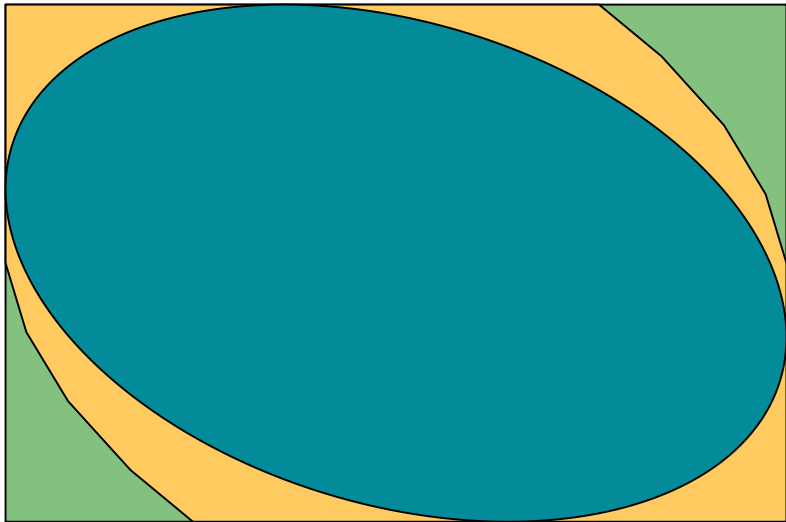
Reduction

The set S is controlled invariant for the DT**CS** iff it is invariant for the DT**AS**

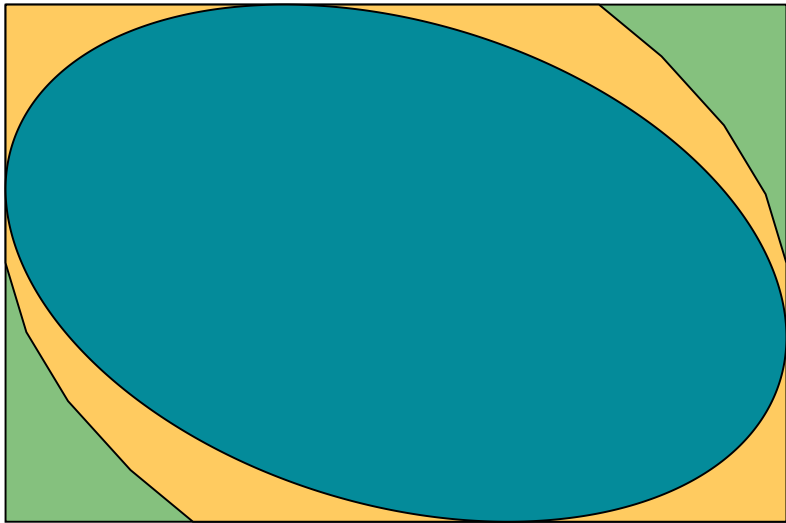
$$\Pi x_{k+1} = \Pi Ax_k, \quad x_k \in X$$

where Π is a projection on $\text{Im}(B)^\perp$.

Result for reservoir



Result for reservoir



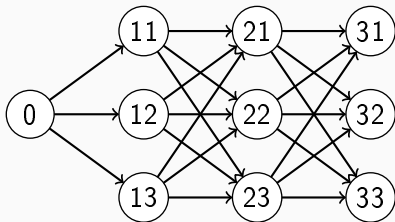
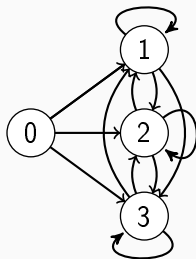
Hybrid Systems

Multistage Stochastic Program to Hybrid System

Precomputing controlled invariant set

Feasibility only, no objective function.

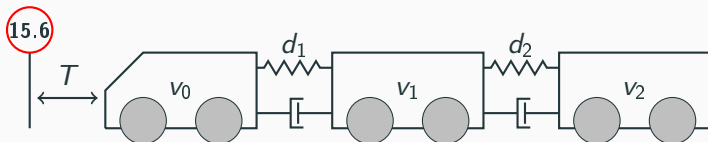
Stochastic Program can be transformed to Hybrid System.



Controlled invariant set for Hybrid Systems

Four speed limits with a delay T to satisfy limit

→ Hybrid System



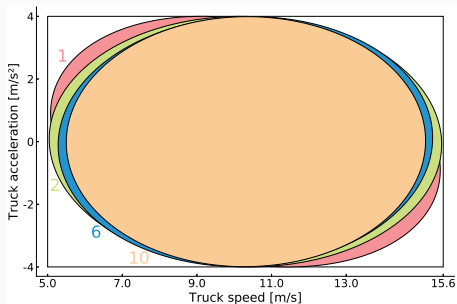
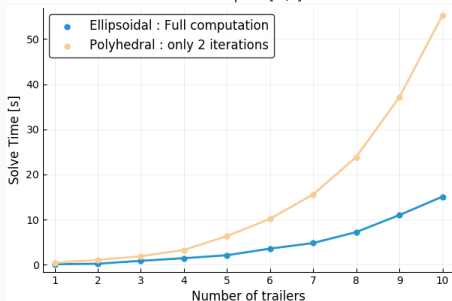
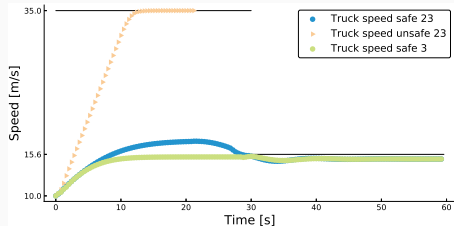
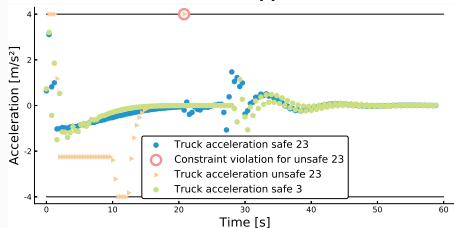
Generalization to Hybrid Systems:

L., B.; Tabuada, P. & Jungers, R. M.

Computing controlled invariant sets for hybrid systems with applications to model-predictive control.

Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems, 2018

Results for Hybrid Systems



Conclusion

Conclusion

- Solving stochastic problems without relatively complete recourse is **difficult**.
- Need to compute maximal **controlled invariant** sets of an **hybrid** system.
- **Ellipsoidal** controlled invariant sets can be computed using **SDP**.

Future work

- **Semialgebraic** invariant sets using **Sum-of-Squares**.
- Improve **BMI** optimization for **non-symmetric** case.
- **Piecewise** quadratic invariant set.