Explicit Set-based Estimation: Optimal Accuracy without Order Reduction Methods

D. Silvestre

ECC 2026 Workshop
Reykjavík, Iceland

July 7th 2026

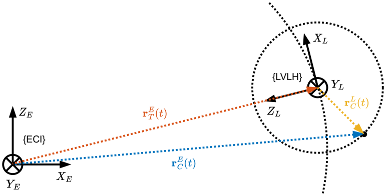
Outline

Outline

Motivating Applications

State Estimation

- Consider a satellite or other dynamical system;
- Measurements define sets of possible state values;
- Perform *prediction* using set operations;
- *Update* the estimates by intersecting with the measurements.



State Estimation Problem

- System is modeled as a Linear Parameter-Varying (LPV):

$$\begin{aligned} x(k+1) &= F(\rho(k))x(k) + B(\rho(k))u(k) + w(k) \\ y(k) &= C(\rho(k))x(k) + v(k) \end{aligned} \tag{1}$$

- We are borrowing the definition from the work of Michael Athans;
- LPVs can represent nonlinear systems for which we can measure the nonlinear terms at runtime.

Recursive Guaranteed State Estimation

- We can compute the set-valued estimates as:

$$X_{k+1} = (F_k X_k \oplus W + B_k u_k) \cap_{C_{k+1}} (y_{k+1} - V). \quad (2)$$

- *predict step*: The above expression is applying the dynamics to all points in the previous estimate.
- *update step*: Then, the propagated set is intersected with the measurement set.
- If all these operations are exact, then we get the optimal estimate (i.e., smallest set).

Required Set Operations (LPV case)

Linear Map $R\mathcal{X} + t$

Set obtained by applying the linear map to all points $\{Rx + t : x \in \mathcal{X}\}$

Minkowski sum $\mathcal{X} + \mathcal{Y}$

Set containing all sums of two vectors from the sets $\{x + y : x \in \mathcal{X}, y \in \mathcal{Y}\}$

Intersection through a Linear Map $\mathcal{X} \cap_R \mathcal{Y}$

All points in $\mathcal{X}$ that after the map R are also in $\mathcal{Y}$ $\{x : x \in \mathcal{X}, Rx \in \mathcal{Y}\}$

Outline

Intervals

An Interval $\mathcal{I}$ can be represented by lower bounds l_b and upper bounds u_b such that:

$$\mathcal{I} = \{x : l_b \leq x \leq u_b\} \tag{3}$$

with element-wise inequalities.

Intervals (alternative definition)

An Interval $\mathcal{I}$ is a specific linear map of the hyper-cube:

$$\mathcal{I} = \{G\xi + c : \|\xi\|_\infty \leq 1\} \tag{4}$$

with G a diagonal matrix.

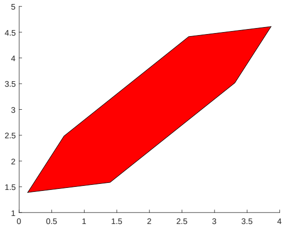
Set	lin. map	lin. eq.	base set	$G\mathcal{X}+t$	$\mathcal{X}+\mathcal{Y}$	$\mathcal{X}\cap_R\mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗

Zonotopes

A Zonotope $\mathcal{Z}$ is a linear map of the hyper-cube:

$$\mathcal{Z} = \{G\xi + c : \|\xi\|_\infty \leq 1\} \tag{3}$$

for any matrix G.



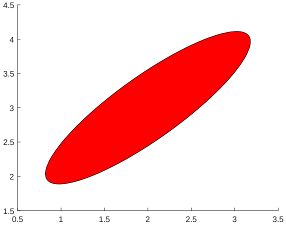
Set	lin. map	lin. eq.	base set	$G\mathcal{X} + t$	$\mathcal{X} + \mathcal{Y}$	$\mathcal{X} \cap_R \mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{Z}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗

Ellipsoids

An Ellipsoid $\mathcal{E}$ is a linear map of an ℓ_2 unit ball:

$$\mathcal{E} = \{G\xi + c : \|\xi\|_2 \leq 1\} \tag{3}$$

for any matrix G.



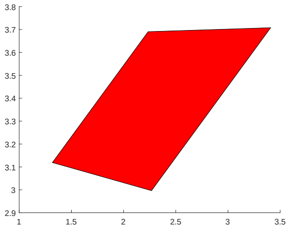
Set	lin. map	lin. eq.	base set	$G\mathcal{X}+t$	$\mathcal{X}+\mathcal{Y}$	$\mathcal{X}\cap_R\mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{Z}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{E}$	✓	✗	$\mathcal{B}_2$	✓	✗	✗

Constrained Zonotopes

A Constrained Zonotope $\mathcal{CZ}$ is a linear map of an ℓ_∞ unit ball with an added linear constraint:

$$\mathcal{CZ} = \{G\xi + c : \|\xi\|_\infty \leq 1, A\xi = b\} \tag{3}$$

for any matrix G.

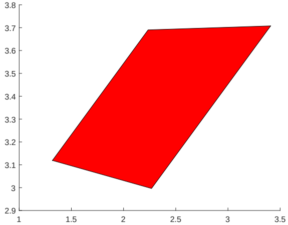


Set	lin. map	lin. eq.	base set	$G\mathcal{X}+t$	$\mathcal{X}+\mathcal{Y}$	$\mathcal{X}\cap_R\mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{Z}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{E}$	✓	✗	$\mathcal{B}_2$	✓	✗	✗
$\mathcal{CZ}$	✓	✓	$\mathcal{B}_\infty$	✓	✓	✓

Polytope

A Polytope $\mathcal{P}$ is the explicit representation equivalent to the implicit way of writing $\mathcal{CZ}$:

$$\mathcal{P} = \{x : Ax \leq b\} \tag{3}$$

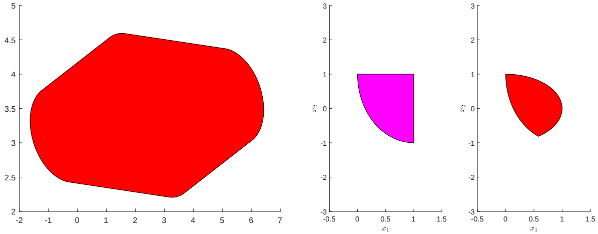


Set	lin. map	lin. eq.	base set	$G\mathcal{X}+t$	$\mathcal{X}+\mathcal{Y}$	$\mathcal{X}\cap_R\mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{Z}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{E}$	✓	✗	$\mathcal{B}_2$	✓	✗	✗
$\mathcal{C}\mathcal{Z}$	✓	✓	$\mathcal{B}_\infty$	✓	✓	✓
$\mathcal{P}$	✓	✓	$\mathcal{B}_\infty$	✓	✓	✓

Constrained Convex Generator

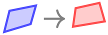
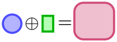
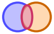
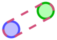
A Constrained Convex Generator $\mathcal{CCG}$ is an implicit representation where the generator variables are constrained to some convex sets:

$$\mathcal{CCG} = \{G\xi + c : A\xi = b, \xi \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_{n_p}\}. \tag{3}$$



Set	lin. map	lin. eq.	base set	$G\mathcal{X}+t$	$\mathcal{X}+\mathcal{Y}$	$\mathcal{X}\cap_R\mathcal{Y}$
$\mathcal{I}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{Z}$	✓	✗	$\mathcal{B}_\infty$	✓	✓	✗
$\mathcal{E}$	✓	✗	$\mathcal{B}_2$	✓	✗	✗
$\mathcal{CZ}$	✓	✓	$\mathcal{B}_\infty$	✓	✓	✓
$\mathcal{P}$	✓	✓	$\mathcal{B}_\infty$	✓	✓	✓
$\mathcal{CCG}$	✓	✓	any	✓	✓	✓

Closure Properties of Constrained Convex Generators

Visual	Operation	Expression	Reference
	Affine mapping	$A\mathcal{C} + b$	[1]
	Minkowski sum	$\mathcal{C}_1 \oplus \mathcal{C}_2$	[1]
	Generalized intersection	$\mathcal{C}_1 \cap_R \mathcal{C}_2$	[1]
	Convex hull	$\mathrm{CH}(\mathcal{C}_1, \mathcal{C}_2)$	[2]
	Pontryagin difference	$\mathcal{C}_1 \ominus \mathcal{C}_2$	[3]

All operations preserve the CCG representation with polynomial complexity growth.

Convex Set Representations

- an interval corresponds to $(G, c, [\,], [\,], \|\xi\|_\infty \leq 1)$, for a diagonal matrix G;
- a zonotope is given by $(G, c, [\,], [\,], \|\xi\|_\infty \leq 1)$;
- an ellipsoid is defined by $(G, c, [\,], [\,], \|\xi\|_2 \leq 1)$, for a square matrix G;
- a CZ or polytope is $(G, c, A, b, \|\xi\|_\infty \leq 1)$;
- a convex cone in $\mathbb{R}^n$ is $(G, c, [\,], [\,], \xi \geq 0)$;
- ellipsotopes are given by $(G, c, A, b, \|\xi\|_{p_1} \leq 1, \cdots , \|\xi\|_{p_m} \leq 1)$, for some $p_i > 0$, $1 \leq i \leq m$;
- AH-polytopes are given by $(G, c, [\,], [\,], A\xi \leq b)$.

Outline

State Estimation for Linear Time-Invariant (LTI)

- System dynamics is assumed to be linear:

$$\begin{aligned} x_{k+1} &= Fx_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{3}$$

- The LTI model allows for less crowded notation.
- The recursive estimation iteration is given by

$$X_{k+1} = (FX_k \oplus W + Bu_k) \cap_C (y_{k+1} - V). \tag{4}$$

- Disturbance W and noise V sets can take any convex form, from ellipsoidal uncertainties to interval constraints or general convex polytopes.

Complexity Issue using an Example

Consider a random system characterized by:

- Dimension: 15 states ($x_k \in \mathbb{R}^{15}$).
- Inputs: 5 ($u_k \in \mathbb{R}^5$).
- Outputs: 3 ($y_k \in \mathbb{R}^3$).

The control input is constant and given by:

$$u_k = 20\mathbf{1}_5, \quad \forall k \geq 0.$$

Complexity Issue using an Example

The disturbance and noise sets are represented as CCGs:

- **Disturbance Set**:

$$\begin{aligned} \mathfrak{C}_W &= \{\xi : \|\xi\|_\infty \leq 1\} \times \{\xi : \|\xi\|_2 \leq 1\}, \\ G_W &= \begin{bmatrix} 2I_{15} & I_{15} \end{bmatrix}, \\ c_W &= \mathbf{0}_{15}. \end{aligned}$$

- **Noise Set**:

$$\begin{aligned} \mathfrak{C}_V &= \{\xi : \|\xi\|_\infty \leq 1\} \times \{\xi : \|\xi\|_2 \leq 1\}, \\ G_V &= \begin{bmatrix} I_3 & 2I_3 \end{bmatrix}, \\ c_V &= \mathbf{0}_3. \end{aligned}$$

Generator increase for the Recursive Approach

- Increasing computational cost.
- Require order reduction, introducing conservatism.

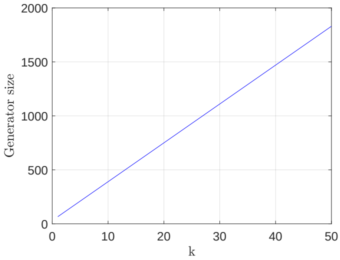


Figure: Number of generators.

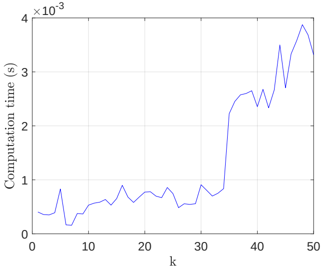


Figure: Computing time.

Proposed Method

Approach

- Coarse ellipsoidal estimate as a baseline.
- Refine with fixed past inputs and measurements.

Features

- Pre-computation offline.
- Minimizes runtime calculations.
- Avoids order reduction.

This is possible because we can add polytopes and ellipsoids!

First approach: Ellipsoidal observer

Consider the Luenberger observer:

$$\hat{x}_{k+1} = F\hat{x}_k + Bu_k + L\left(y_k - C\hat{x}_k\right), \tag{7}$$

The estimation error dynamics is given by:

$$e_{k+1} = (F - LC)e_k + w_k - Gv_k. \tag{8}$$

First approach: Ellipsoidal observer

An ellipsoidal observer can be given in CCG form as:

$$\hat{X}_k = \left(\left(a^k e_{init} + \frac{e_{noise}}{1-a} \right) P^{-\frac{1}{2}}, \hat{x}_k, [\,], [\,], \|\xi\|_2 \leq 1 \right). \tag{9}$$

with

$$e_{init} := \max_{\xi \in X_0} \|P^{\frac{1}{2}} \xi\|_2 \tag{10}$$

$$e_{noise} := \max_{\xi \in W \oplus -LV} \|P^{\frac{1}{2}} \xi\|_2 \tag{11}$$

First approach: Ellipsoidal observer

- Low computational cost.
- No need to compute the set description online. It is only necessary to compute the Luenberger state estimate.

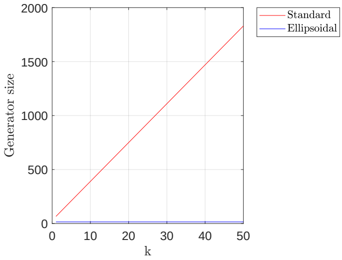


Figure: Number of generators.

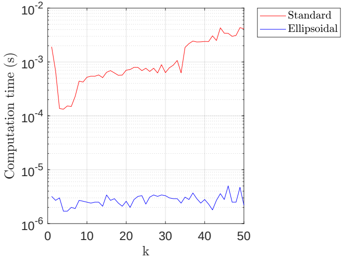
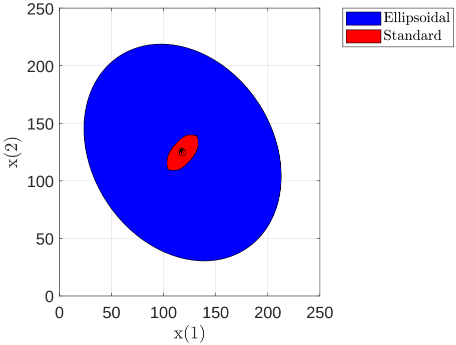


Figure: Computing time.

First approach: Ellipsoidal observer

- High conservatism.



One step ahead observer

We can use the ellipsoidal observer as first approach and then perform a standard set based observer :

$$\hat{X}_{k+1} = F\left(\hat{X}_k \cap_C (y_k - V)\right) \oplus W + Bu_k. \tag{12}$$

One step ahead observer

The estimated state set is then given by:

$$\hat{X}_{k+1} = \left(G_{k+1}, c_{k+1}, A_{k+1}, b_{k+1}, \mathfrak{C}_{k+1}\right), \tag{13}$$

with

$$\mathfrak{C}_{k+1} = \mathfrak{C}_w \times \mathfrak{C}_v \times \{\xi : \|\xi\|_2 \leq 1\}, \tag{14}$$

$$G_{k+1} = \left[G_w \quad 0 \quad F\left(a^{k-l}e_{init} + \frac{e_{noise}}{1-a}\right)P^{-\frac{1}{2}}\right], \tag{15}$$

$$c_{k+1} = c_w + F\hat{x}_k, \tag{16}$$

$$A_{k+1} = \left[0 \quad G_v \quad C\left(a^{k-l}e_{init} + \frac{e_{noise}}{1-a}\right)P^{-\frac{1}{2}}\right], \tag{17}$$

$$b^l_{k+1} = y_k - c_v - C\hat{x}_k. \tag{18}$$

One step ahead observer

- Low computational cost.
- No need to compute the set description online. It is only necessary to compute the Luenberger state estimate.

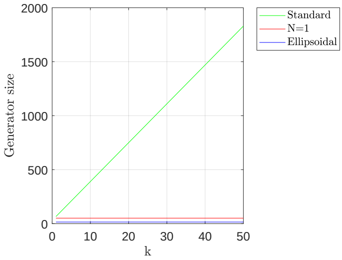


Figure: Number of generators.

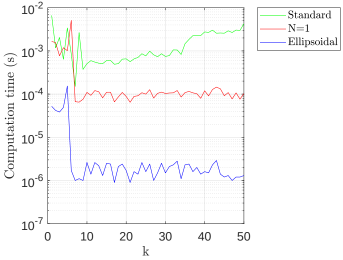
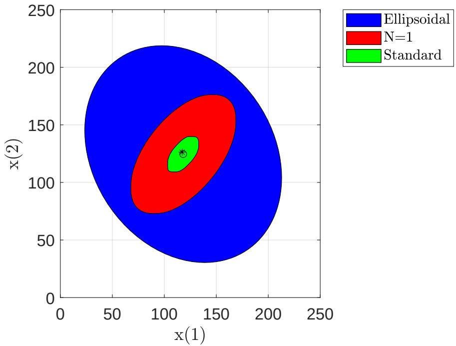


Figure: Computing time.

One step ahead observer

- Lower conservatism.



Proposed Method

Store the previous l sensor measurements and control inputs:

$$Y_k^l := \begin{bmatrix} y_{k-1} \\ \vdots \\ y_{k-l} \end{bmatrix}, \quad U_k^l := \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-l} \end{bmatrix}, \tag{19}$$

Proposed Method

The state estimates performing l state updates after the ellipsoidal observer takes the form:

$$X_k^l = \left(G_{X,k}^l, c_{X,k}^l, A_{X,k}^l, b_{X,k}^l, \mathfrak{C}_X^l\right) \subset \mathbb{R}^n, \tag{20}$$

where

$$G_{X,k}^l = \begin{bmatrix} G^l & \left(a^{k-l} e_{init} + \frac{e_{noise}}{1-a}\right) G_0^l \end{bmatrix}, \tag{21a}$$

$$c_{X,k}^l = c^l + c_U^l U_k^l + c_0^l \hat{x}_{k-l}, \tag{21b}$$

$$A_{X,k}^l = \begin{bmatrix} A^l & \left(a^{k-l} e_{init} + \frac{e_{noise}}{1-a}\right) A_0^l \end{bmatrix}, \tag{21c}$$

$$b_{X,k}^l = Y_k^l + b^l + b_U^l U_k^l + b_0^l \hat{x}_{k-l}. \tag{21d}$$

Proposed Method

One wishes to pre-compute matrices G^l, G^l_0, c^l, c^l_U, c^l_0, A^l, A^l_0, b^l, b^l_U and b^l_0.
For $l = 0$ we have the ellipsoidal observer with $G^0, c^0_U, A^0, A^0_0, b^0, b^0_U, b^0_0 = [\,]$ and

$$\mathfrak{C}^0_X := \{\xi : \|\xi\|_2 \leq 1\}, \tag{22a}$$

$$G^0_0 := P^{-\frac{1}{2}}, \tag{22b}$$

$$c^0 := \mathbf{0}_n, \tag{22c}$$

$$c^0_0 := I_n. \tag{22d}$$

Proposed Method

For $l > 0$ the update equations yield the following recursions:

$$\mathfrak{C}_X^{l+1} = \mathfrak{C}_w \times \mathfrak{C}_v \times \mathfrak{C}_X^l, \qquad (23a)$$

$$G^{l+1} = \begin{bmatrix} G_w & \mathbf{0} & FG^l \end{bmatrix}, \qquad (23b)$$

$$G_0^{l+1} = FG_0^l, \qquad (23c)$$

$$c^{l+1} = Fc^l + c_w, \qquad (23d)$$

$$c_U^{l+1} = \begin{bmatrix} B & Fc_U^l \end{bmatrix}, \qquad (23e)$$

$$c_0^{l+1} = Fc_0^l, \qquad (23f)$$

$$A^{l+1} = \begin{bmatrix} \mathbf{0} & G_v & CG^l \\ \mathbf{0} & \mathbf{0} & A^l \end{bmatrix}, \qquad (23g)$$

$$A_0^{l+1} = \begin{bmatrix} CG_0^l \\ A_0^l \end{bmatrix}, \qquad (23h)$$

$$b^{l+1} = \begin{bmatrix} -Cc^l - c_v \\ b^l \end{bmatrix}, \qquad (23i)$$

$$b_U^{l+1} = \begin{bmatrix} \mathbf{0} & Cc_U^l \\ \mathbf{0} & b_U^l \end{bmatrix}, \qquad (23j)$$

$$b_0^{l+1} = \begin{bmatrix} Cc_0^l \\ b_0^l \end{bmatrix}. \qquad (23k)$$

Proposed Method

for $k \geq N$ **do**

$\hat{x}_{k-N+1} = F\hat{x}_{k-N} + Bu_{k-N} + L\left(y_{k-N} - C\hat{x}_{k-N}\right)$;

Compute Y^N_{k+1} by storing y_k and discarding y_{k-N};

Compute U^N_{k+1} by storing u_k and discarding u_{k-N};

$G^N_{X,k+1} = \left[G^N \quad \left(a^{k+1-N} e_{init} + \frac{e_{noise}}{1-a}\right) G^N_0\right]$;

$c^N_{X,k+1} = c^N + c^N_U U^N_{k+1} + c^N_0 \hat{x}_{k+1-N}$;

$A^N_{X,k+1} = \left[A^N \quad \left(a^{k+1-N} e_{init} + \frac{e_{noise}}{1-a}\right) A^N_0\right]$;

$b^N_{X,k+1} = Y^N_{k+1} + b^N + b^N_U U^N_{k+1} + b^N_0 \hat{x}_{k+1-N}$;

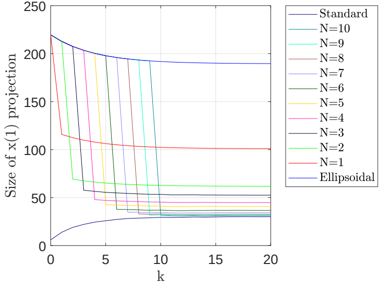
$X_{k+1} = \left(G^N_{X,k+1}, c^N_{X,k+1}, A^N_{X,k+1}, b^N_{X,k+1}, \mathfrak{C}^N_X\right)$;

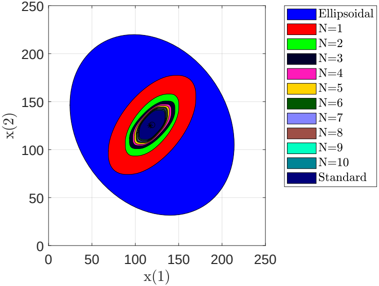
end

Numerical Results

Simulation Setup

- Random system with 15 states, 5 inputs, and 3 outputs.
- Initial state and disturbances modelled with CCGs.





Numerical Results

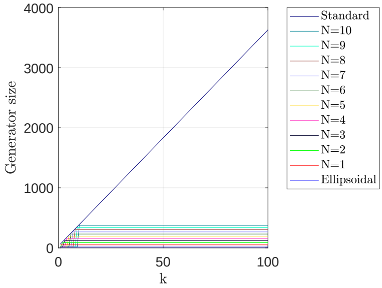


Figure: Number of generators

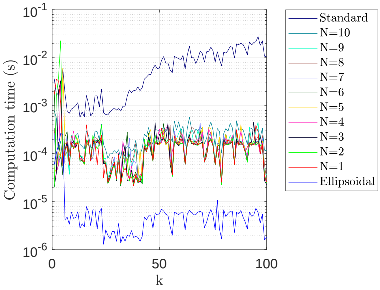
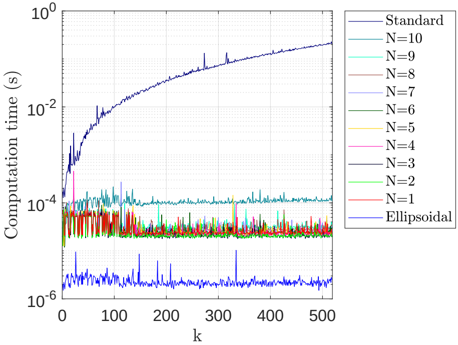


Figure: Computing time.

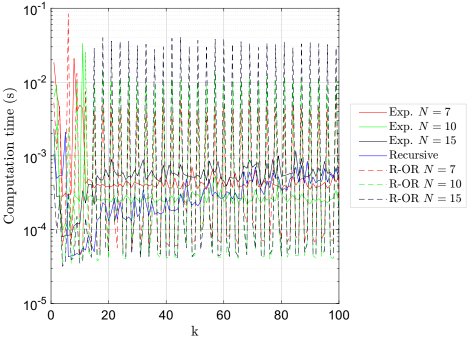
Numerical Results

- The recursive approach would require order reductions.
- Regardless of the horizon, the computing time stays bounded.



Numerical Results

- The proposed approach also works for time-varying matrices.
- It gets slower for computing matrices products online.
- This resolves the *sawtooth* pattern in the computing time.



Outline

Definition of the Minimal RPI

- Consider systems with linear dynamics:

$$x(k+1) = Ax(k) + w(k) \tag{24}$$

RPI Definition

A set $\Omega \subset \mathbb{R}^n$ is a Robust Positively Invariant (RPI) set for the system in (24) if and only if $A\Omega \oplus W \subseteq \Omega$.

Iterative Solution

$$F_\infty = \bigoplus_{i=0}^{\infty} A^i W. \tag{25}$$

RPI computation using CCGs

- Main idea is to split the computation and consider a remainder:

$$F_\infty = \bigoplus_{i=0}^{H} A^i W \oplus \underbrace{\bigoplus_{i=H+1}^{\infty} A^i W}_{\text{remainder } \Theta}, \qquad (26)$$

- Outer approximation corresponds to overbounding Θ by the ellipsoid written in CCG format as:

$$(\alpha\beta I_n, 0_n, [\,], [\,], \|\xi\|_2 \leq 1)$$

- $\alpha = \sum_{i=1}^{\infty} \|A^{H+i}\|_2$ and β is the radius of W

RPI computation using CCGs

- Inner approximation is *easier* and Θ can be replaced by:

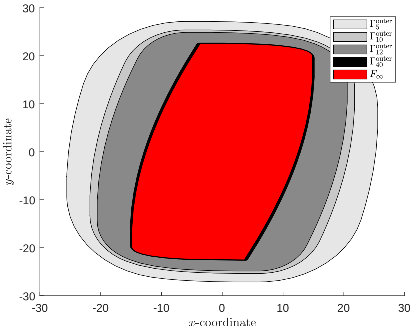
$$\left[(I_n - A)^{-1} - \sum_{i=0}^{H} A^i\right] W$$

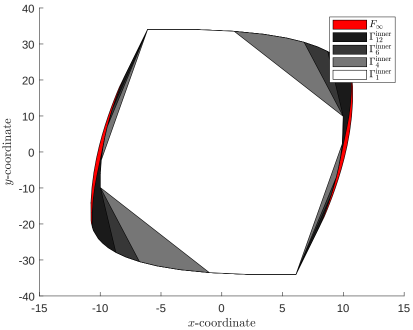
- Overall, since CCGs represent both ellipsoids and polytopes, different methods to bound Θ can be used.
- The possibility to represent various sets is what makes this approximation accurate!

Numerical Results for the RPI Accuracy

Simulation Setup

- Random system with 2 states and 2 disturbance signals.
- Disturbance is the set $[-2, 2] \times [-2, 2]$.



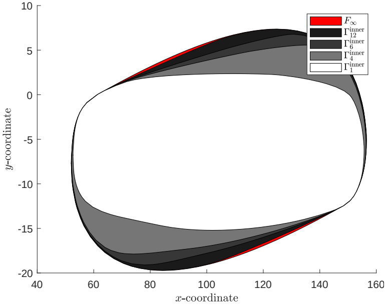


Numerical Results for the RPI Accuracy

Simulation Setup

- Disturbance $W = (G, c, A, b, \mathfrak{C})$, with $G \in \mathbb{R}^{2\times 20}$ and $A \in \mathbb{R}^{10\times 20}$;
- $\mathfrak{C} = \{\xi : \|\xi\|_\infty \leq 1\} \times \{\xi : \|\xi\|_2 \leq 1\}$.

- F^∞ has 3120 generators and 1560 constrainsts;
- $\Gamma_{12}^{\text{inner}}$ has 280 and 140.



Outline

Safe Navigation Problem

- We will consider an ego vehicle described by

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{f}(\mathbf{p}) + \mathbf{G}(\mathbf{p})\mathbf{z}, \\ \dot{\mathbf{z}} &= \mathbf{f}_1(\mathbf{p}, \mathbf{z}) + \mathbf{G}_1(\mathbf{p}, \mathbf{z})\mathbf{u}, \end{aligned} \tag{27}$$

- Ego vehicle has a body shape provided by

$$\mathcal{P}(\mathbf{p}) = \mathbf{p} + \bar{\mathcal{P}}, \tag{28}$$

- There are M dynamics obstacles with convex bodies

$$\mathcal{O}_i(\mathbf{o}_i) = \mathbf{o}_i + \bar{\mathcal{O}}_i, \tag{29}$$

Safe Navigation Problem

- Obstacles follow linear dynamics (our recent LCSS submission deals with uncertain nonlinear dynamics):

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{F}_i \mathbf{x}_i + \mathbf{w}_i, \\ \mathbf{o}_i &= \mathbf{E}_i \mathbf{x}_i, \end{aligned} \tag{30}$$

- We get measurements related to the state of the obstacles

$$\mathbf{y}_{i,k} = \mathbf{C}_i \mathbf{x}_{i,k} + \mathbf{v}_{i,k}, \tag{31}$$

Problem Formulation

For the trajectory $\varphi : \mathbb{R}_{\geq 0} \to \mathbb{R}^p$ of the ego vehicle ensure

$$\mathcal{P}(\varphi(t)) \cap \left(\bigcup_{i \in \mathcal{I}} \mathcal{O}_i(\varphi_i(t)) \right) = \emptyset, \forall t \in \mathbb{R}_{\geq 0} \tag{32}$$

Safe Navigation Solution

- For this problem, the set-valued flows for the obstacles need to be retrieved from the measurements $\mathbf{y}_{i,k}$.
- Since collision avoidance must be guaranteed, we adopt a set-valued observer.
- Another main challenge is the requirement to have a continuous-time observer such that h is well-defined.
- Our first step is to reduce the ego vehicle to a single point of mass by doing

$$\mathcal{O}_i^+(\mathbf{o}_i) = \mathbf{o}_i + \bar{\mathcal{O}}_i^+ = \mathbf{o}_i + \bar{\mathcal{O}}_i \oplus \left(-\bar{\mathcal{P}}\right), \qquad (33)$$

Safe Navigation Solution

- The safety condition becomes

$$\varphi(t) \notin \bigcup_{i\in\mathcal{I}} \mathcal{O}_i^{+}(\varphi_i(t)) \tag{34}$$

- The obstacle enlarged body evolution in time corresponds

$$\hat{\mathcal{O}}_{i,k}^{+}(t) = \mathbf{E}_i\left(\mathbf{\Phi}_i(t-t_k)\hat{\mathcal{X}}_{i,k} \oplus \mathbf{\Gamma}_i(t-t_k)\tilde{\mathcal{W}}_i\right) \oplus \bar{\mathcal{O}}_i^{+}, \tag{35}$$

- Where

$$\mathbf{\Phi}_i(s) = \exp(\mathbf{F}_i s), \quad \mathbf{\Gamma}_i(s) = \frac{\exp(\|\mathbf{F}_i\| s) - 1}{\|\mathbf{F}_i\|} \max_{\mathbf{w}_i \in \mathcal{W}_i} \|\mathbf{w}_i\| \tag{36}$$

Safe Navigation Solution

- Ingredients for the solution:
  - We use CCGs because of the closure for all set operations;
  - A nonlinear QP-CLF-CBF controller for the ego vehicle:
    - Control Lyapunov Function (CLF) - accounts for the stability objective;
    - Control Barrier Function (CBF) - maintains safety.
  - A method to convert the set description into a valid CBF $h_{i,k}(\mathbf{p}, t))$ for each obstacle;
  - Use the LogSumExp trick to convert to a single CBF;
  - Apply backstepping to handle the higher relative order dynamics.

Set-valued Observer

- The *continuous* evolution in-between sensor samples was given by

$$\hat{\mathcal{O}}^{+}_{i,k}(t) = \mathbf{E}_i \left(\mathbf{\Phi}_i(t - t_k)\hat{\mathcal{X}}_{i,k} \oplus \mathbf{\Gamma}_i(t - t_k)\tilde{\mathcal{W}}_i \right) \oplus \bar{\mathcal{O}}^{+}_{i}, \quad (37)$$

- The set-valued estimate for the measurement time k is

$$\hat{\mathcal{X}}_k = \left(\mathbf{\Phi}(T_s)\hat{\mathcal{X}}_{k-1} \oplus \mathbf{\Gamma}(T_s)\tilde{\mathcal{W}} \right) \cap_{\mathbf{C}} (\mathbf{y}_k - \mathcal{V}), \quad (38)$$

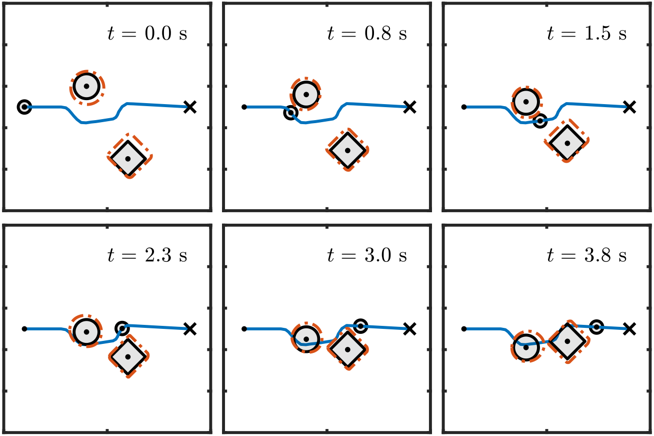
- Both expressions are of fixed complexity!

Explicit Finite-Horizon Estimator

- The recursive observer has a number of generators and constraints that is linearly increasing.
- With a fixed horizon, the estimate for in-between samples is given by the CCG

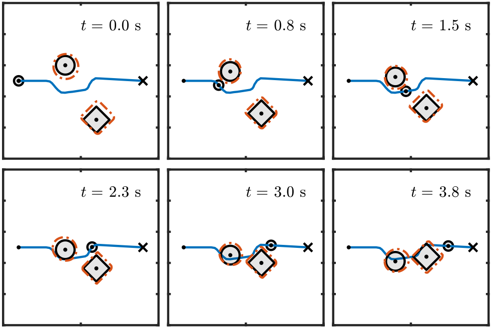
$$
\begin{aligned}
\hat{\mathcal{O}}_k^+(t) &= \left(\mathbf{G}_{\hat{o},k}^+(t), \mathbf{c}_{\hat{o},k}^+(t), \mathbf{A}_{\hat{o},k}^+, \mathbf{b}_{\hat{o},k}^+, \mathfrak{G}_{\hat{o},k}^+\right), \\
\mathbf{G}_{\hat{o},k}^+(t) &= \begin{bmatrix}\mathbf{E}\mathbf{\Phi}(t-t_k)\mathbf{G}_{\hat{x},k} & \mathbf{E}\mathbf{\Gamma}(t-t_k)\mathbf{G}_{\tilde{w}} & \mathbf{G}_{\bar{o}}^+\end{bmatrix}, \\
\mathbf{c}_{\hat{o},k}^+(t) &= \mathbf{E}\mathbf{\Phi}(t-t_k)\mathbf{c}_{\hat{x},k} + \mathbf{E}\mathbf{\Gamma}(t-t_k)\mathbf{c}_{\tilde{w}} + \mathbf{c}_{\bar{o}}^+, \\
\mathbf{A}_{\hat{o},k}^+ &= \begin{bmatrix}\mathbf{A}_{\hat{x},k} & \mathbf{0} & \mathbf{0}\end{bmatrix}, \\
\mathbf{b}_{\hat{o},k}^+ &= \mathbf{b}_{\hat{x},k}, \\
\mathfrak{G}_{\hat{o},k}^+ &= \mathfrak{G}_{\hat{x},k} \times \mathfrak{G}_{\tilde{w}} \times \mathfrak{G}_{\bar{o}}^+,
\end{aligned}
\tag{39}
$$

Simulations on Single-Integrator Ego Vehicle



Simulations on Single-Integrator Ego Vehicle

- Consider $\dot{\mathbf{p}} = \mathbf{z}$ with nominal controller $\mathbf{k}_{\mathsf{d}}(\mathbf{p}) = K(\mathbf{p} - \bar{\mathbf{p}})$.
- Ego vehicle has ellipsoidal shape with two obstacles with polytopic and ellipsoidal shapes.



Outline

Concluding Remarks

- We are currently working on a CCG implementation for Python along with Mitsubishi Electric Research Laboratories (MERL).
- Expect in a couple of months a C++ implementation of the explicit state estimation.
- The translation of CCGs into CBFs is available in arXiv.
- An interesting challenge is deriving ways to perform explicit observers for nonlinear systems.

Student Contributions

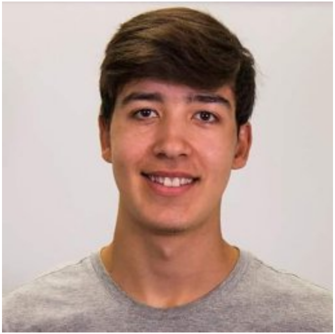


Figure: Hugo Matias (PhD student) - for the topics related to Hybrid CLF-CBF controllers.

Figure: Francisco Rego (former PostDoc) - for the topics related to explicit CCG observers.

References I

[1] D. Silvestre, "Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements," *IEEE Control Systems Letters*, vol. 6, pp. 1610–1615, 2022. DOI: 10.1109/LCSYS.2021.3129729

[2] D. Silvestre, "Exact set-valued estimation using constrained convex generators for uncertain linear systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9461–9466, 2023, 22nd IFAC World Congress, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2023.10.241

[3] D. Silvestre and A. P. Vinod, "Robust controllable set computation using constrained convex generators,", 2026.

[4] H. Matias and D. Silvestre, "Safe navigation under uncertain obstacle dynamics using control barrier functions and constrained convex generators," *arXiv preprint arXiv:2601.07715*, 2026.

[5] F. Rego and D. Silvestre, "Explicit computation of guaranteed state estimates using constrained convex generators," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 1400–1405. DOI: 10.1109/CDC56724.2024.10885976

The end

- Thank you for your time.

Explicit Set-based Estimation: Optimal Accuracy without Order Reduction Methods

D. Silvestre

ECC 2026 Workshop
Reykjavík, Iceland

July 7th 2026