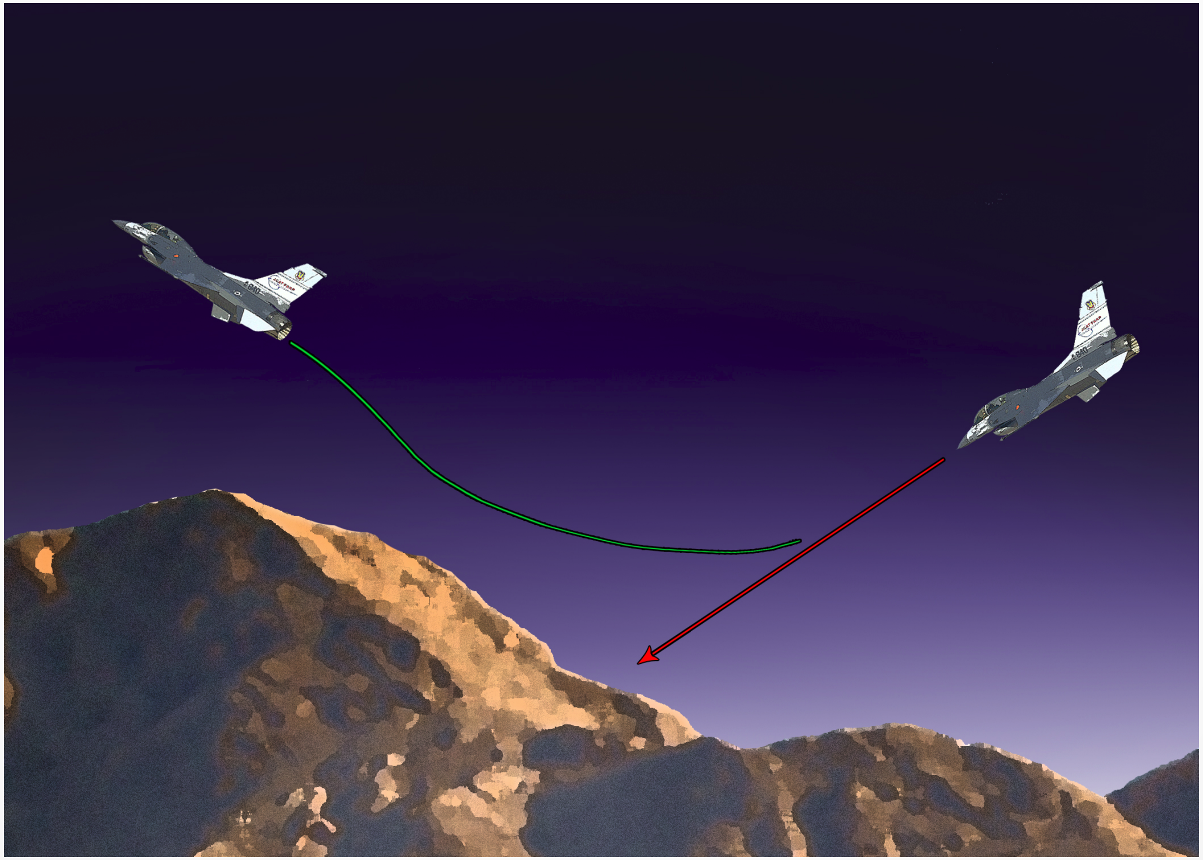
Simulation-Equivalent Reachability of Large Linear Systems with Inputs

Stanley Bak and Parasara Sridhar Duggirala

Inspiration + Potential

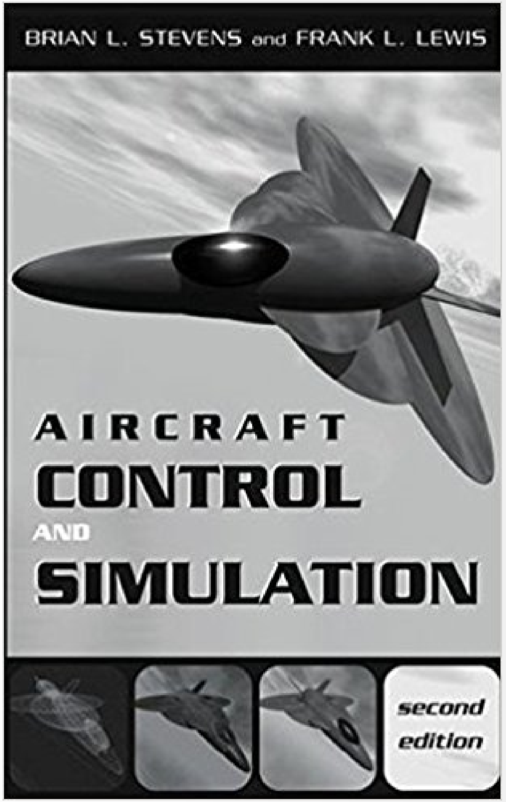
Automatic Ground Collision Avoidance System
(Auto-GCAS)



F-16 Aircraft

Good News:

Very well-studied system!



F-16 Aircraft

Good News:

Very well-studied system!

Bad News:

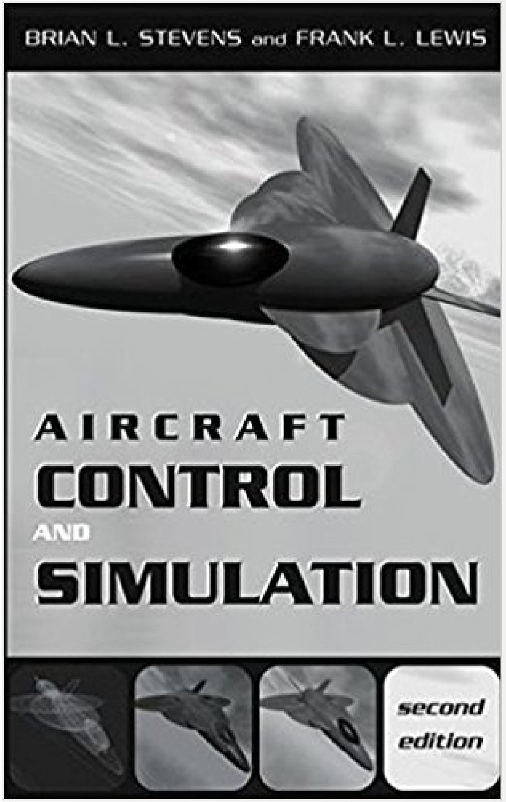
Differential Equations

~15 variables (dimensions)

Nonlinear

Discrete Switches

Look-up Tables



F-16 Aircraft

Good News:

Very well-studied system!

Bad News:

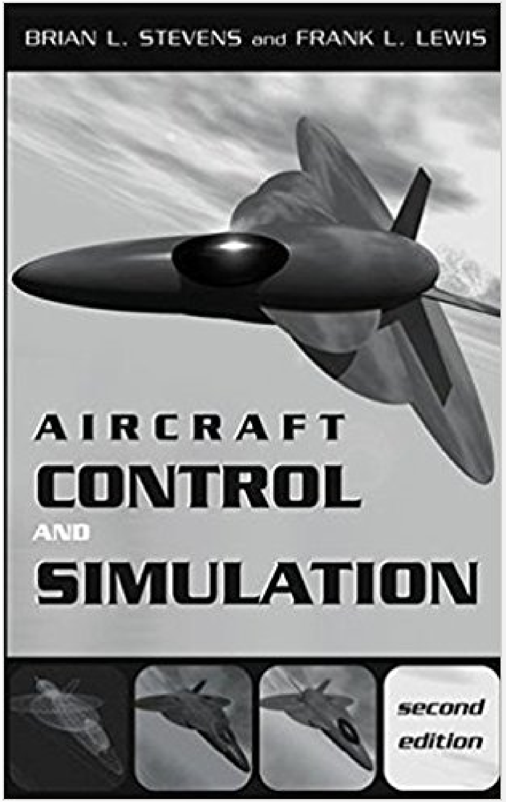
Differential Equations

~15 variables (dimensions)

Nonlinear

Discrete Switches

Look-up Tables



Good News:

We're researchers; lots of problems to work on

One-Slide Contribution Summary

Computing reachability (and safety checking) for linear systems with inputs

$$\boxed{\dot{x} = Ax + Bu(t)}$$

– State of the art: 10s to 100s of dimensions

One-Slide Contribution Summary

Computing reachability (and safety checking) for linear systems with inputs

$$\boxed{\dot{x} = Ax + Bu(t)}$$

- State of the art: 10s to 100s of dimensions
- **New method: 10000+ dimensions**

One-Slide Contribution Summary

Computing reachability (and safety checking) for linear systems with inputs

$$\boxed{\dot{x} = Ax + Bu(t)}$$

- State of the art: 10s to 100s of dimensions
- **New method: 10000+ dimensions**
- Bonus: Accurate counter-example generation

Linear Dynamical Systems

A linear system with inputs

$$\boxed{\dot{x} = Ax + Bu(t)}$$

has the solution

$$\boxed{x(t) = \underbrace{e^{At}x_0}_{\text{Input-free solution}} + \underbrace{\int_0^t e^{A(t-s)} Bu(t) \; ds}_{\text{Input Effects}}}$$

Set-based Constraints

If we have sets of initial states and inputs

$$\boxed{x(0) \in \mathcal{X}_0} \qquad \boxed{u(t) \in \mathcal{V}}$$

then the possible solutions at time t are

$$\boxed{\mathcal{X}_t = e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-s)} B\mathcal{V} \; ds}$$

Minkowski sum of two sets of states
(addition of every two points in the sets)

Minkowski Sum

Definition 2.2 (Minkowski sum). *The Minkowski sum of two sets $\mathcal{X}$ and $\mathcal{Y}$ is the set of sums of elements from $\mathcal{X}$ and $\mathcal{Y}$:*

$$\mathcal{X} \oplus \mathcal{Y} = \{x + y : x \in \mathcal{X} \text{ and } y \in \mathcal{Y}\}$$

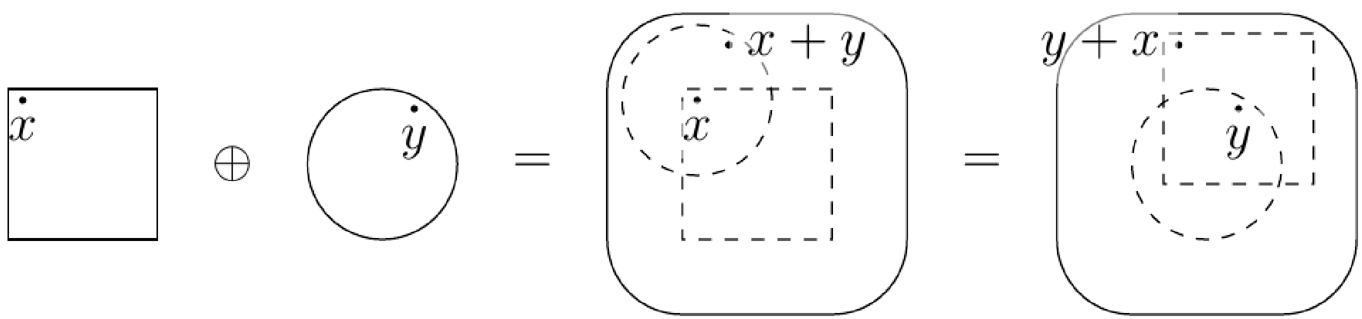


Figure 2.2: Minkowski sum of a square and a disk.

Image From: Colas Le Guernic. Reachability analysis of hybrid systems with linear continuous dynamics. Diss. Université Joseph-Fourier-Grenoble I, 2009.

Fixed-Step Fixed-Inputs

If we assume inputs are fixed, we can solve for the set of input-effects after one step

$$\boxed{\begin{aligned} \mathcal{X}_t &= e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-s)} B\mathcal{V}\ ds \\ &= e^{At}\mathcal{X}_0 \oplus {\color{red}\mathcal{U}} \end{aligned}}$$

Note: If we wanted dense-time reachability, we could add bloating terms to this equation.

Multiple Step Solution

For two steps, we can express the set of states recursively

$$\boxed{\begin{aligned} \mathcal{X}_{2t} &= e^{At}\mathcal{X}_t \oplus \mathcal{U} \\ &= e^{A2t}\mathcal{X}_0 \oplus e^{At}\mathcal{U} \oplus \mathcal{U} \end{aligned}}$$

which generalizes after i steps to

$$\boxed{\mathcal{X}_{it} = e^{Ait}\mathcal{X}_0 \oplus e^{A(i-1)t}\mathcal{U} \oplus e^{A(i-2)t}\mathcal{U} \oplus \ldots \oplus e^{At}\mathcal{U} \oplus \mathcal{U}}$$

Step-by-step computation

$$\mathcal{X}_{it} = e^{Ait}\mathcal{X}_0 \oplus e^{A(i-1)t}\mathcal{U} \oplus e^{A(i-2)t}\mathcal{U} \oplus \ldots \oplus e^{At}\mathcal{U} \oplus \mathcal{U}$$

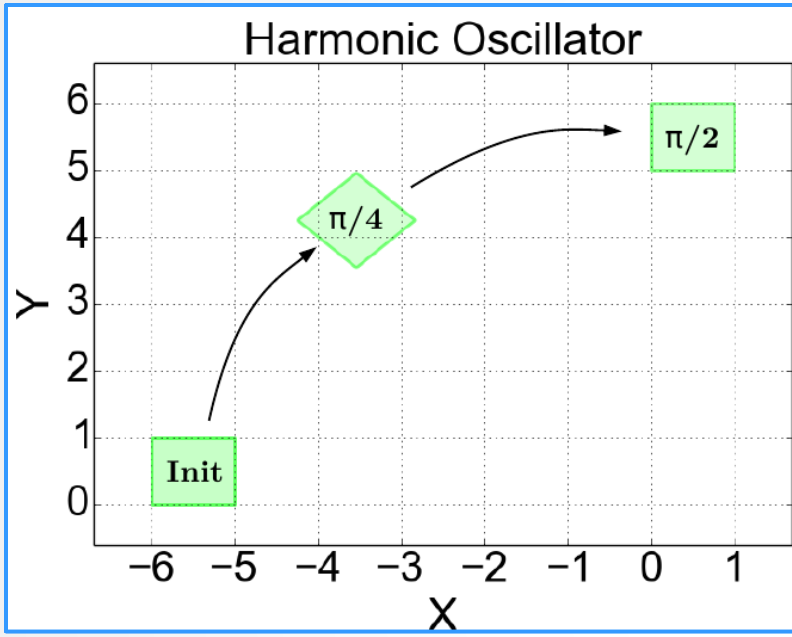
At each new time step, we need to:

- $e^{Ait} X_0$
- $e^{A(i-1)t} U$
- Perform Minkowski sum
- (Optional) Intersection check with unsafe states

Linear Star Representation

- Linear Stars are a specialized version of generalized star sets
  - Presented in CAV '16
- Efficient for computing:
  - Time elapse (matrix multiplication)
  - Intersection checking (linear programming)
  - **Minkowski sum (new)**

Linear Star Constraints



Basis Matrix - Relationship between initial states and states at current step

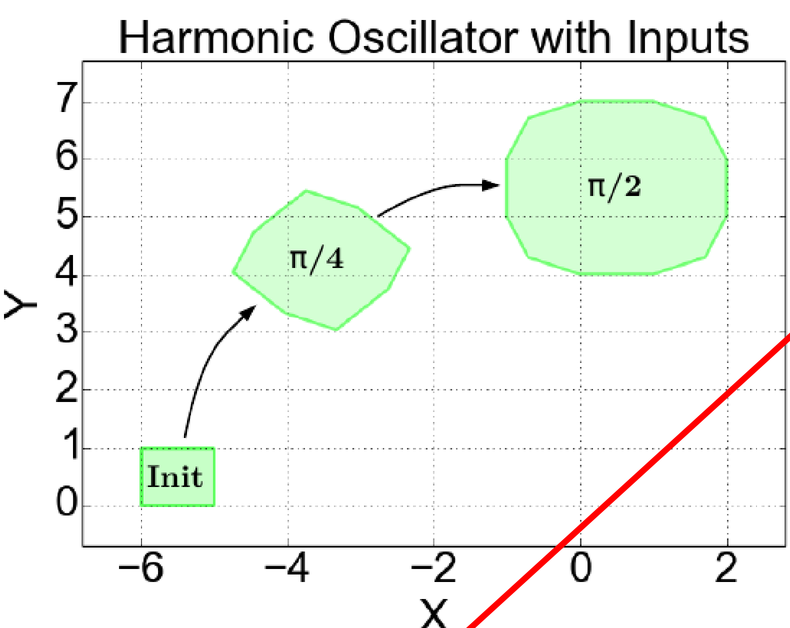
$$\begin{pmatrix} -1 & 0 & 0.707 & 0.707 \\ 0 & -1 & -0.707 & 0.707 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \alpha_1 \\ \alpha_2 \end{pmatrix} \begin{matrix} = \\ = \\ \leq \\ \leq \\ \leq \\ \leq \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 6 \\ -5 \\ 0 \\ 1 \end{pmatrix}$$

Constraints on initial states

The basis matrix is equal to the matrix exponential at the current step (e^{At}).

Minkowski Sum

Minkowski sum with linear stars can be done by combining the basis matrices & constraints



$$\begin{pmatrix} -1 & 0 & 0.0 & 1.0 & 0.707 & 0.293 & 0.293 & 0.707 \\ 0 & -1 & -1.0 & 0.0 & -0.293 & 0.707 & -0.707 & 0.293 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \alpha_1 \\ \alpha_2 \\ u_1 \\ u_2 \\ u'_1 \\ u'_2 \end{pmatrix} \begin{matrix} = \\ = \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 6 \\ -5 \\ 0 \\ 1 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

$$\mathcal{X}_{it} = e^{Ait}\mathcal{X}_0 \oplus e^{A(i-1)t}\mathcal{U} \oplus e^{A(i-2)t}\mathcal{U} \oplus \ldots \oplus e^{At}\mathcal{U} \oplus \mathcal{U}$$

Optimizations

"Engineering matters: you can't properly evaluate a technique without an efficient implementation"

- Ken McMillan

- Optimization 0 (old) – Use Simulations for e^{At}
- Optimization 1 – Decomposed LP
- Optimization 2 – Warm-Start LP

Optimization #1

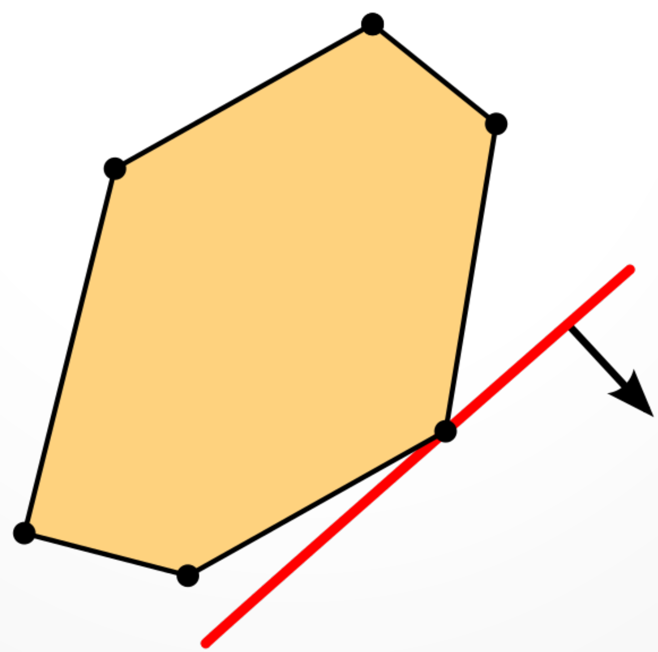
$$\mathcal{X}_{it} = e^{Ait}\mathcal{X}_0 \oplus e^{A(i-1)t}\mathcal{U} \oplus e^{A(i-2)t}\mathcal{U} \oplus \ldots \oplus e^{At}\mathcal{U} \oplus \mathcal{U}$$

Proposition 1. *If* $S = S_1 \oplus S_2$, *then* $\max_v(S) = \max_v(S_1) + \max_v(S_2)$

- Maximizing over a Minkowski sum can be decomposed into maximizing over the component sets
  - Necessary and sufficient for single constraint
  - Necessary for a conjunction of constraints
- Possible future optimization with conjunctions of constraints: Danzig-Wolfe decomposition for LP

Optimization #2

- Safety checking requires solving a linear program (LP) **at each step**
- We can reuse previous solutions to seed subsequent computations
  - Usually reduces required LP iterations to zero!



Benchmarks

We implemented the method in a tool, **Hylaa**. We then evaluated the method on a large linear systems with inputs verification benchmark suite*:

- Motor (11 dims)
- Building (50 dims)
- Partial Differential Equation (86 dims)
- Heat (202 dims)
- International Space Station (274 dims)
- Clamped Beam (350 dims)
- MNA1 (588 dims)
- FOM (1008 dims)
- MNA5 (**10923 dims**)

* "Large-scale linear systems from order-reduction", H. D. Tran, L. V. Nguyen, and T. T. Johnson, 3rd Applied Verification for Continuous and Hybrid Systems Workshop (ARCH 2016)

Benchmarks

We implemented the method in a tool, **Hylaa**. We then evaluated the method on a large linear systems with inputs verification benchmark suite*:

Hylaa succeeded on <u>all</u> benchmarks!

(first tool to do this)

- FOM (1008 dims)
- MNA5 (**10923 dims**)

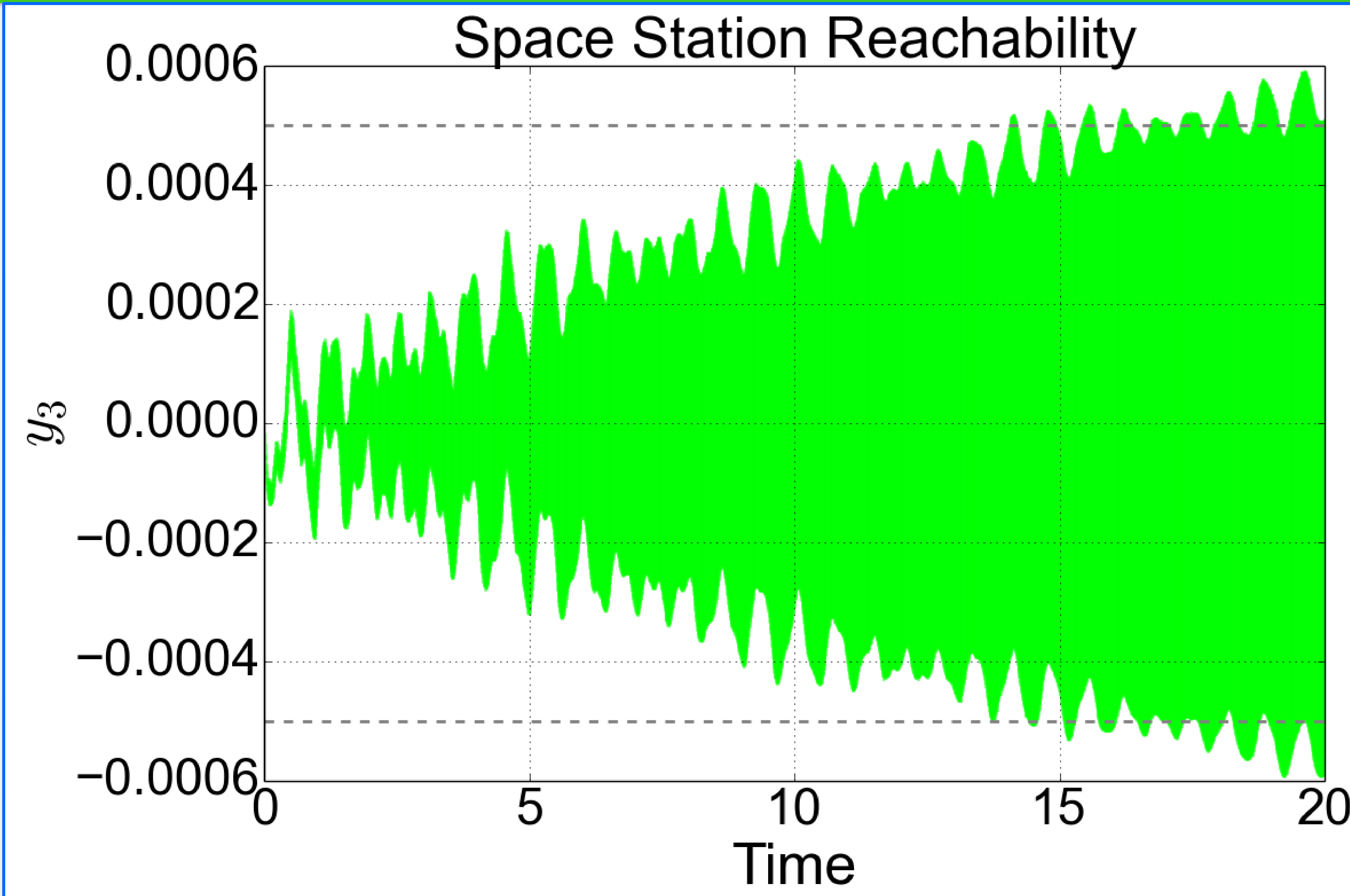
* "Large-scale linear systems from order-reduction", H. D. Tran, L. V. Nguyen, and T. T. Johnson, 3rd Applied Verification for Continuous and Hybrid Systems Workshop (ARCH 2016)

International Space Station Model (271 dimensions)

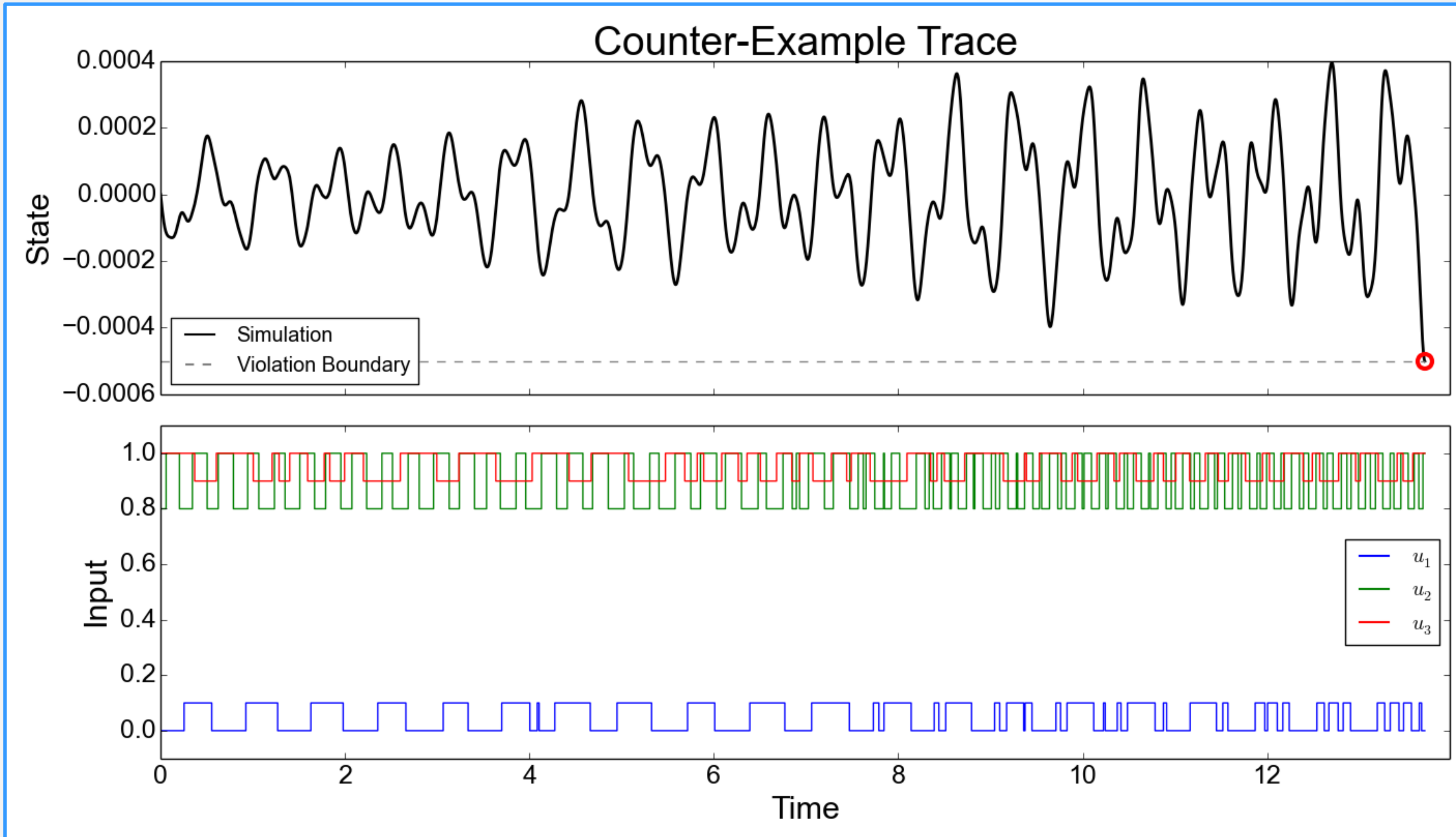
ISS	271	$y_3 \notin [-0.0007, 0.0007]$	Hylaa	1m28s	✓	-	-
ISS*	271	$y_3 \notin [-0.0005, 0.0005]$	Hylaa	1m23s		$8.5{\cdot}10^{-6}/1.3{\cdot}10^{-5}$	13.71

- The original safety specification was created using simulations. For most models it was safe.
- For the International Space Station model, however, **it was not!** This shows that simulation can miss errors. The error was not known before analysis with **Hylaa**.

Reachability Plot



Space Station Specification Violation



$2^{270} \times 8^{(13.71/0.005)} = 3 \times 10^{2557}$ cases!

Falsification tool did not succeed after 4 hours.

MNA5 Model (10923 dimensions)

MNA5*	10914	$x_1 \geq 0.2 \vee x_2 \geq 0.15$	Hylaa	6h23m	✓	-	-
MNA5	10914	$x_1 \geq 0.1 \vee x_2 \geq 0.15$	Hylaa	37m27s		$1.4 \cdot 10^{-6} / 1.8 \cdot 10^{-6}$	1.92

- Completed analysis of 11000 dimensional system, for 4000 steps, in about 6.5 hours
- Unsafe variant counter-example is highly accurate (relative error of about $1.8 * 10^{-6}$)

The Journey to 10000 Dimensions

- The original 10000-dimensional MNA5 benchmark model file is empty!
- SpaceEx Model Editor freezes! Use text editor. Gedit → Geany
- Hyst conversion (ANTRL Grammar Exception), 11k * 2 initial conditions
- Hyst stack overflow → internal expression tree unbalanced
- 800MB Python script → OS freezes (cannot run first line)
- OS effectively freezes when swap is active and large sim time is used
- Change Hyst to initialize matrix of zeros and assign entries (sparse matrix)
- Out of memory while computing... 800 MB * 4000 steps = 3.2 TB RAM needed!
  - Don't use explicit Jacobian in ODEINT
  - Python uses processes for parallelism... keep dynamics sparse
  - Run simulations a few steps at a time
- Random crashes “pickling” matrices, LP solving GLPK errors...
  bad memory stick!

Conclusion

Time-bounded safety verification including counter-example generation for large (10000+ dimensional) linear systems with inputs

– **Two orders of magnitude improvement!**

Future: even larger linear systems, verification methods for partial-differential equations, hybrid and nonlinear models

Hylaa Source: github.com/stanleybak/hylaa