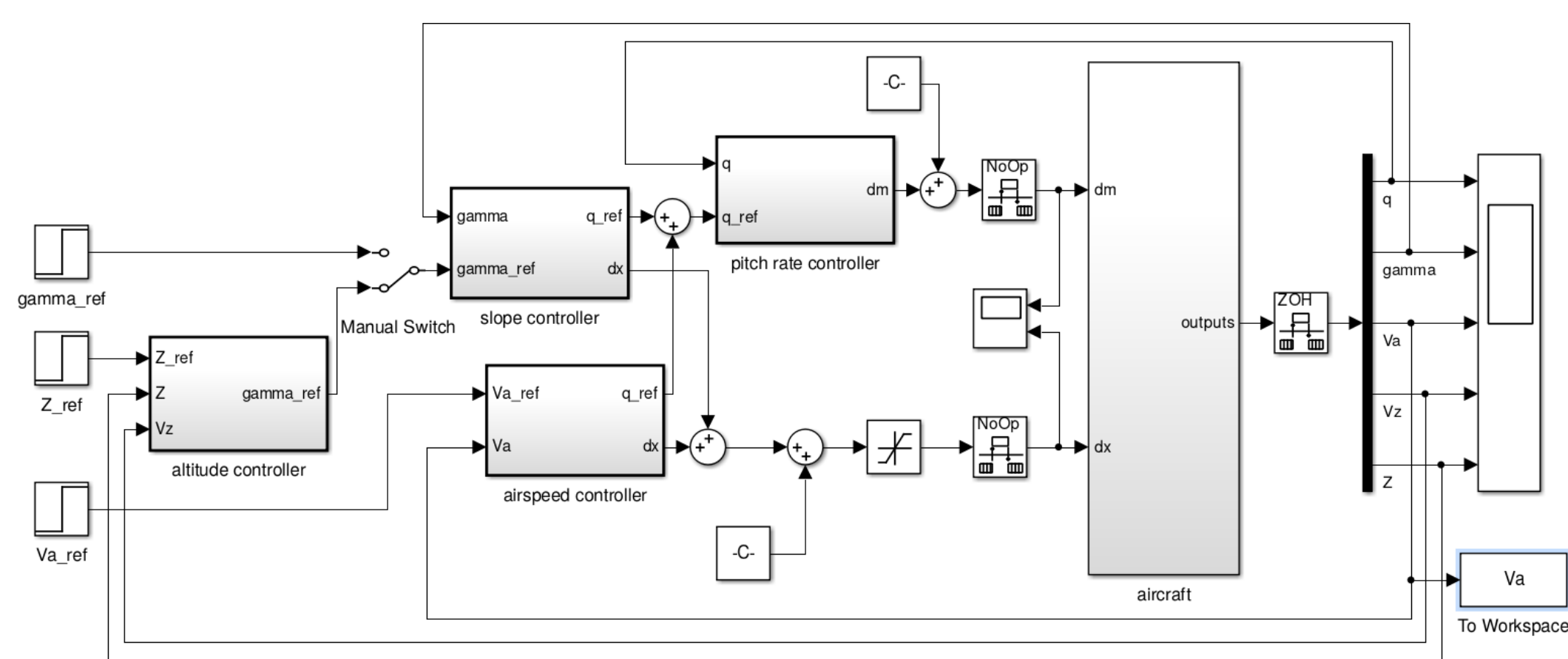


SYNCHRONOUS LANGUAGES AT ONERA TOULOUSE

Frédéric Boniol, Rémi Delmas, Pierre-Loïc Garoche,
Thomas Loquen, Claire Pagetti, Pierre Roux

Overview



Matlab Simulink

static analysis

```
1 node pitch_rate_syslonr_pitchratecontroller
2   (q_1_1 : real; q_ref_1_1 : real)
3   returns (dm_1_1 : real);
4 var
5   Gain_1_1 : real;
6   Gain_1_1_1 : real;
7   Gain_2_1_1 : real;
8   Saturation_1_1 : real;
...

```

Lustre (, Prelude)

```
1 #include <assert.h>
2 #include "pitch_rate_syslonr.h"
3 /* C code generated by lustrec */
4 /* Struct definitions */
5 struct pitch_rate_syslonr_mem {
6   struct pitch_rate_syslonr_reg {
7     double __pitch_rate_syslonr_2;
8     double __pitch_rate_syslonr_3;
...

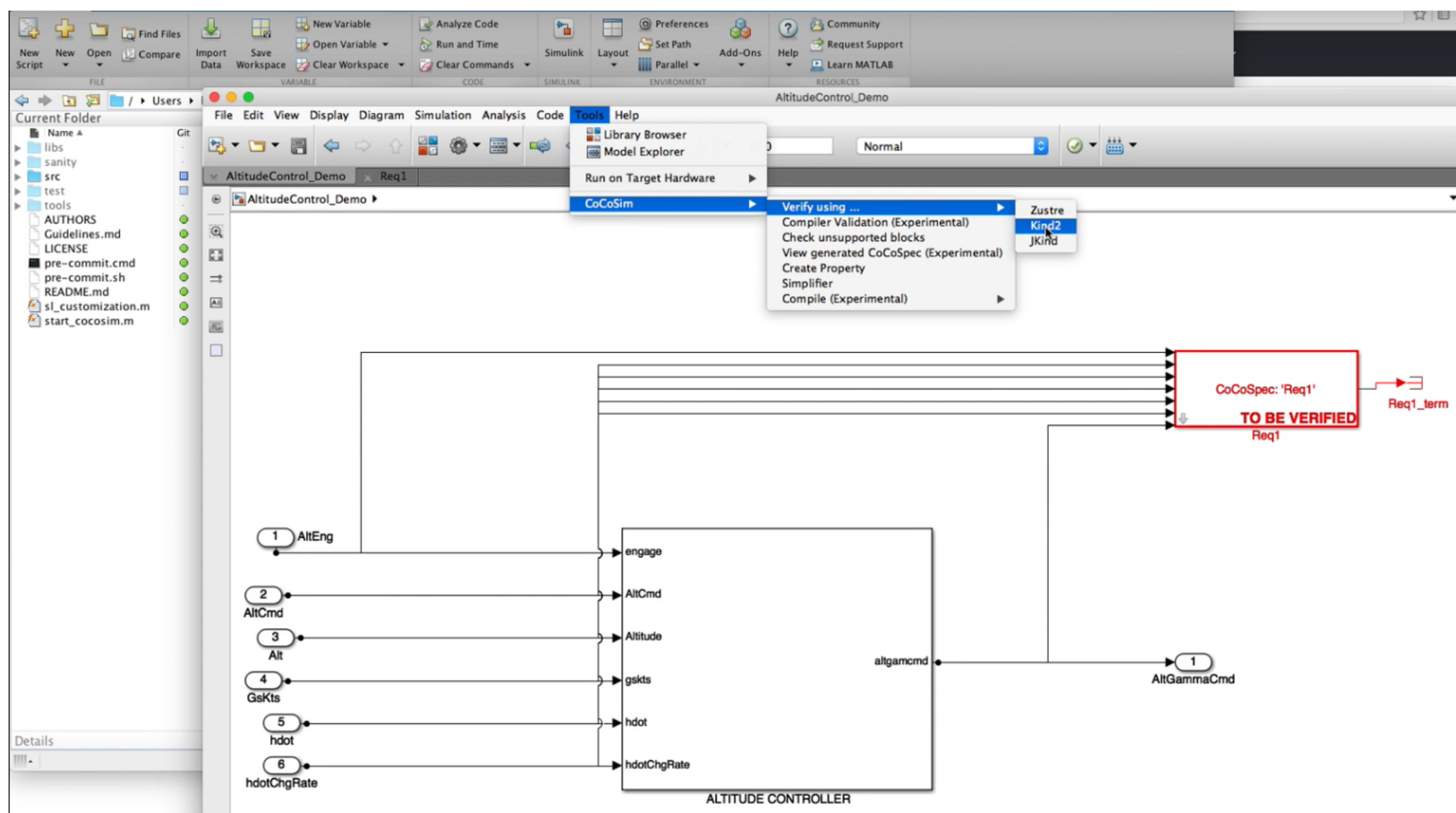
```

C

Cocosim: from Simulink to Lustre

Hamza Bourbough (NASA), Christophe Garion (ISAE), Pierre-Loïc Garoche,
Temesghen Kahsai (NASA), Thomas Loquen, Claire Pagetti

- Compiles a (discrete subset of) **Matlab Simulink** to **Lustre**.
- Interface to call **model checkers** (JKind, Kind2, Zestre) and display result.
- <https://github.com/coco-team/cocoSim>



lustrec: from Lustre to C

Pierre-Loïc Garoche, Xavier Thirioux (IRIT)

- Compiles **Lustre** to **C**.
- "industrial" compilation scheme (one C function for each Lustre node)
- Other backend: **Horn Clauses** (for model checking).
- LGPL license.
- <https://cavale.enseeiht.fr/redmine/projects/lustrec>

Project:

- compile Lustre+annotations to C+ACSL (ANSI C Specification Language).

Hybrid Models

Rémi Delmas, Thomas Loquen

- Model of physical systems often contain **continuous time** parts.
- Goal: compile hybrid Simulink models (with both discrete and continuous time) to a format usable by hybrid model checkers to find bugs.
- Challenge: industrial benchmarks are fairly large.
- Ongoing project.

Static Analysis

Pierre-Loïc Garoche, Pierre Roux

- Synthesizing **quadratic invariants** for control-command systems.
- Using **numerical optimization solvers** (SDP).
- Ensuring soundness despite massive use of **floating-point arithmetic**.

Example: on the following code

node example(in : real) **returns** (x0, x1, x2 : real)

let

assert(in >= -1. && in <= 1.);

x0 = 0 → **pre** (0.9379 * x0 - 0.0381 * x1 - 0.0414 * x2) + 0.0237 * in;

x1 = 0 → **pre** (-0.0404 * x0 + 0.968 * x1 - 0.0179 * x2) + 0.0143 * in;

x2 = 0 → **pre** (0.0142 * x0 - 0.0197 * x1 + 0.9823 * x2) + 0.0077 * in;

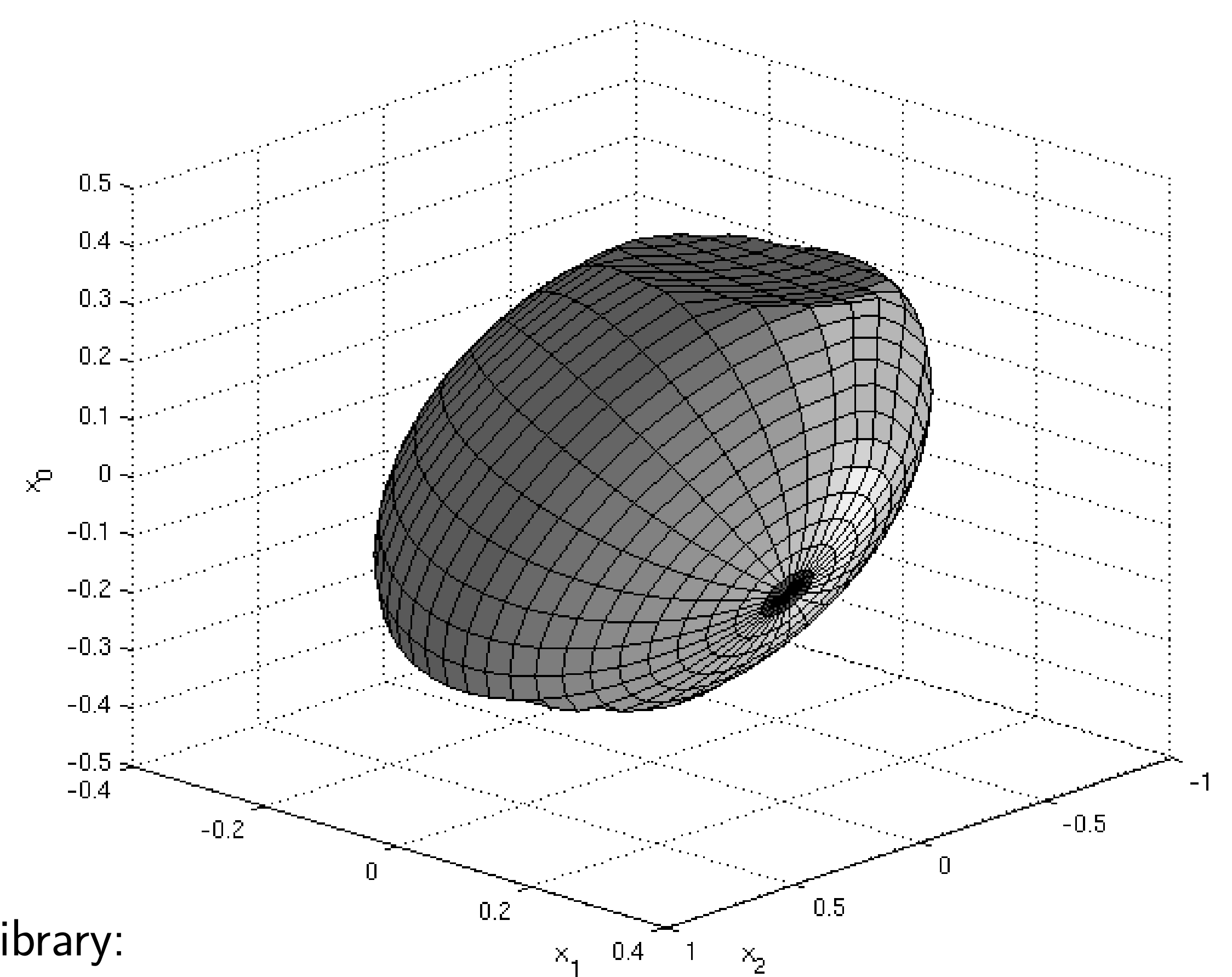
tel

one can synthesize the quadratic invariant (displayed below)

$$6.2547x_0^2 + 12.1868x_1^2 + 3.8775x_2^2 - 10.61x_0x_1 - 2.4306x_0x_2 + 2.4182x_1x_2 \leq 1.0029$$

$$\wedge x_0^2 \leq 0.1795 \wedge x_1^2 \leq 0.1136 \wedge x_2^2 \leq 0.2757.$$

which implies $|x_0| \leq 0.4236 \wedge |x_1| \leq 0.3371 \wedge |x_2| \leq 0.5251$.



OCaml library:

- LGPL license.
- <https://cavale.enseeiht.fr/osdp/>

Prototype analyzer:

- GPL license.
- <https://cavale.enseeiht.fr/riny/>

Prelude

Julien Forget (LIFL), Claire Pagetti

- **Multi periodic** synchronous language.
- See detailed poster by Julien Forget.