Formal Proofs for Control Theory and Robotics: A Case Study

Damien Rouhling

Université Côte d'Azur, Inria, France

June 6, 2018

Motivations

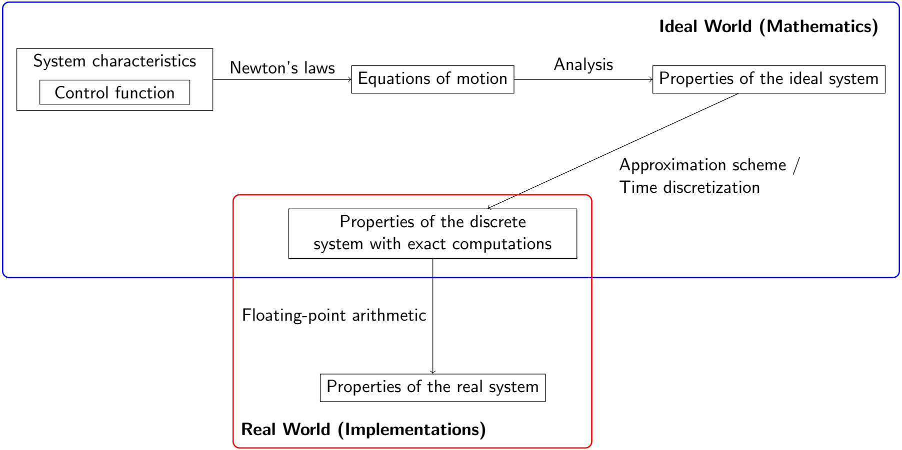
- Safety is critical in many applications of robotics.

  Example: control wheel steering (CWS) in an aircraft.
- We focus here on control theory: a program, or control function, operates a robot in order to achieve a goal.

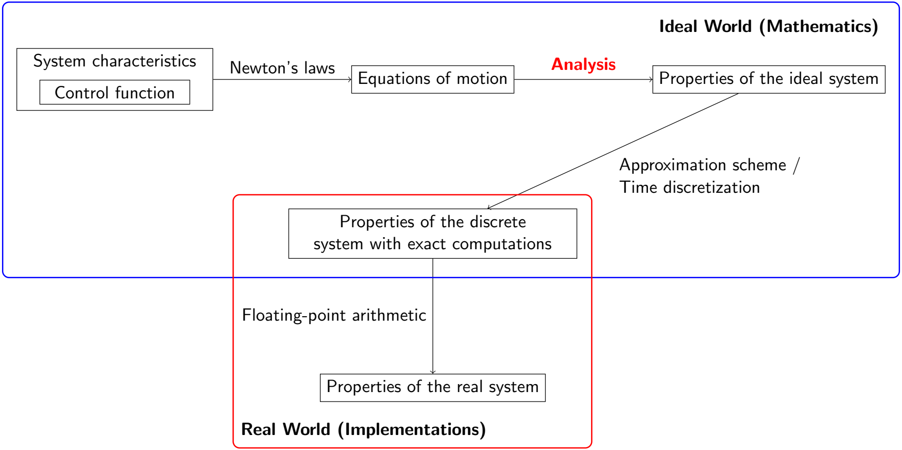
  Example: goal of the CWS: maintain the heading and attitude of the aircraft as set by the pilot.
- We want to bring formal guarantees on this control function: the goal is achieved, no safety condition is violated.

  Example: safety conditions for the CWS:
  - The altitude stays in a given range.
  - No abrupt variation of the aircraft position.
  - The aircraft does not deviate too much from its heading.
  - ...

Formal verification for such systems

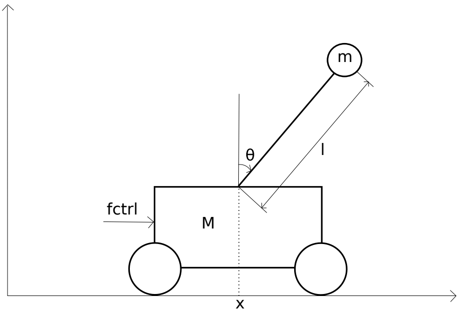


Formal verification for such systems



The inverted pendulum

The inverted pendulum is a standard example for testing control techniques.



- Goal: stabilize the pendulum on its unstable equilibrium.
- Control function: force fctrl applied to the cart.
- Safety condition: none/the cart stays near its starting point.

Context

- Control function and stability proof from [Lozano et al., 2000].
- Proof based on LaSalle's invariance principle [LaSalle, 1960].
- Principle: qualitative analysis of the solutions of a first-order autonomous differential equation:

$$\dot{y} = \mathrm{F} \circ y.$$

Contributions

- LaSalle's invariance principle generalized and formalized [Cohen and Rouhling, 2017].
- A stability proof for the inverted pendulum corrected and formalized [Rouhling, 2018].
- Yet another analysis library[1], compatible with MATHEMATICAL COMPONENTS.

[1] `https://github.com/math-comp/analysis`, joint work with Reynald Affeldt, Cyril Cohen, Assia Mahboubi and Pierre-Yves Strub.

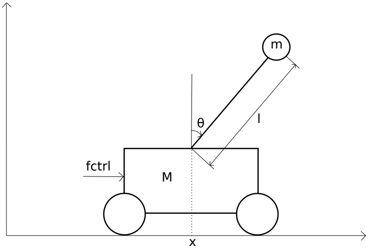
Contributions

- LaSalle's invariance principle generalized and formalized [Cohen and Rouhling, 2017].
- A stability proof for the inverted pendulum corrected and formalized [Rouhling, 2018].
- Yet another analysis library[1], compatible with MATHEMATICAL COMPONENTS.

Remark: we did the proofs twice, first using the COQUELICOT library [Boldo et al., 2015], then using the MATHEMATICAL COMPONENTS ANALYSIS library.

[1] `https://github.com/math-comp/analysis`, joint work with Reynald Affeldt, Cyril Cohen, Assia Mahboubi and Pierre-Yves Strub.

Homoclinic orbit



- Lozano et al. prove the convergence of solutions to a homoclinic orbit:

$$\frac{1}{2}ml^2\dot{\theta}^2 = mgl\left(1 - \cos\theta\right).$$

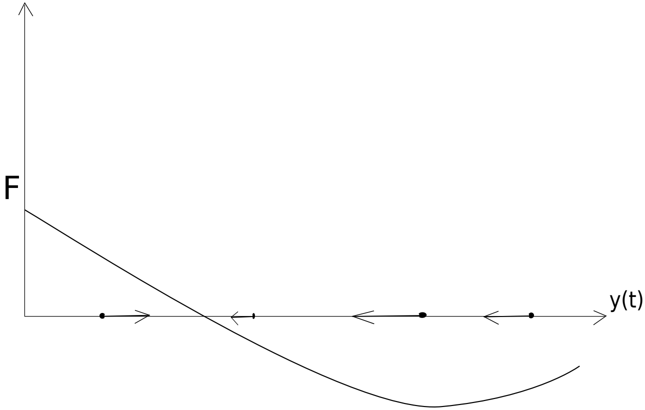
- This is done by an energy approach: the homoclinic orbit is characterised by $E = 0$ and $\dot{x} = 0$.
- They also want the cart to stop at its initial position:

$$x = 0 \text{ and } \dot{x} = 0.$$

LaSalle's invariance principle for real functions

The differential system as a vector field:

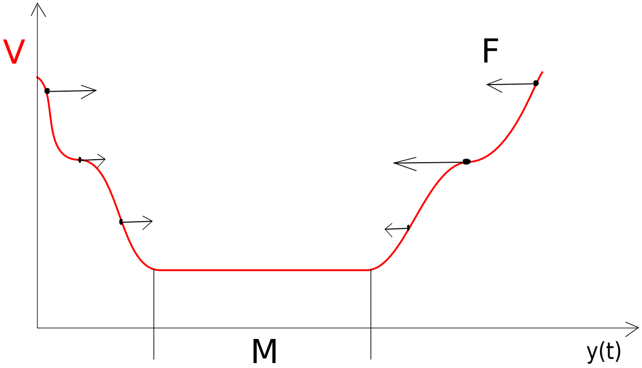
$$\dot{y} = \mathrm{F} \circ y$$



LaSalle's invariance principle for real functions

A sufficient condition for stability:

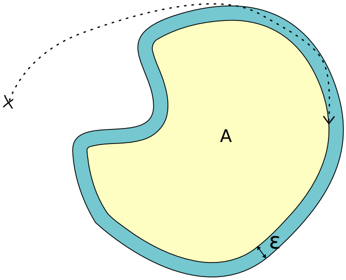
$$\dot{y} = \mathrm{F} \circ y$$



Preliminary definitions

- A set A is said to be invariant if every solution to $\dot{y} = \mathrm{F} \circ y$ starting in A (i.e. $y(0) \in A$) remains in A.
- A function of time $y(t)$ approaches a set A as t approaches infinity, denoted by $y(t) \to A$ as $t \to +\infty$, if

$$\forall \varepsilon > 0, \exists T > 0, \forall t > T, \exists p \in A, \|y(t) - p\| < \varepsilon.$$



The positive limiting set

Definition

Let y be a function of time. The positive limiting set of y, denoted by $\Gamma^+(y)$, is the set of all points p such that

$$\forall \varepsilon > 0, \forall T > 0, \exists t > T, \|y(t) - p\| < \varepsilon.$$

In other terms, $\Gamma^+(y)$ is the set of limit points of y at infinity.

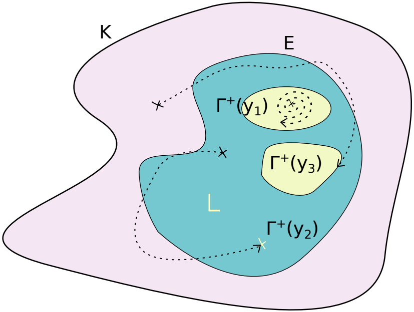
Remark: a function with values (ultimately) in a compact set converges to its positive limiting set as time goes to infinity.

Generalization of LaSalle's invariance principle

Assume

- F is such that we have the existence and uniqueness of solutions to $\dot{y} = \mathrm{F} \circ y$ and the continuity of solutions relative to initial conditions in K.
- K compact and invariant.
- $V : \mathbb{R}^n \to \mathbb{R}$ is continuous in K and differentiable along trajectories of solutions starting in K.
- $\tilde{V}(p) \leq 0$ in K where $\tilde{V}(p)$ is the directional derivative of V at point p along the trajectory of the solution starting at p.

Then, for $L := \bigcup_{\substack{y \text{ solution} \\ \text{starting in } K}} \Gamma^+(y)$ and $E := \left\{ p \in K \mid \tilde{V}(p) = 0 \right\}$, L is an invariant subset of E and for all solution y starting in K, $y(t) \to L$ as $t \to +\infty$.



LaSalle's invariance principle for the inverted pendulum

- The Lyapunov function V is **minimised** along trajectories. Our goal is $E = 0$, $x = 0$ and $\dot{x} = 0$. A possible choice is

$$V = \frac{k_E}{2} E^2 + \frac{k_v}{2} \dot{x}^2 + \frac{k_x}{2} x^2.$$

- The laws of Physics give a second-order differential equation. We transform the equation on (x, θ) into a first-order equation on

$$p = (p_0, p_1, p_2, p_3, p_4) = \left(x, \dot{x}, \cos\theta, \sin\theta, \dot{\theta} \right).$$

- We lose pieces of information. The invariant compact set K will help keeping them as invariants.

$$K = \left\{ p \in \mathbb{R}^5 \mid p_2^2 + p_3^2 = 1 \text{ and } V(p) \leq k_0 \right\}.$$

A few aspects of the formalization

We make a pervasive use of filters.

- A set of sets F is a filter if
  - $F \neq \emptyset$.
  - $\forall P, Q \in F, P \cap Q \in F$.
  - $\forall P \in F, \forall Q \supseteq P, Q \in F$.
- Examples:
  - Neighbourhood filter of a point p, written `locally` p.
  - Neighbourhood filter of $+\infty$, written `Rbar_locally p_infty`.
  - Image of a filter `F` by a function `f`:
    `filtermap f F` $:= \{A \mid \texttt{f}^{-1}(A) \in \texttt{F}\}$.
- Convergence: $\lim_{x} f = y$ written
  `filterlim f (locally x) (locally y)` in Coquelicot.
- Compactness can also be expressed in terms of filters.

A few aspects of the formalization (cont.)

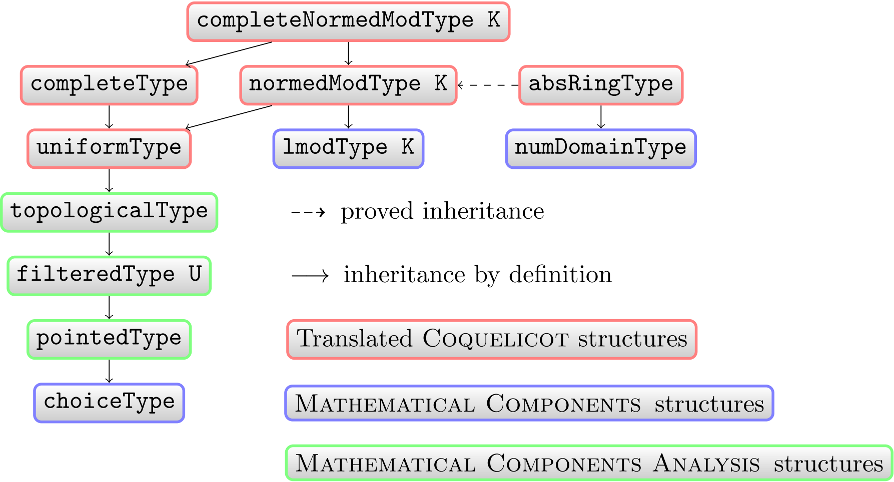
For this formalization we worked on:

- A theory of sets, together with notations.
- An inference mechanism for filters and notations for limits.
- Several aspects of topology:
  - Compact sets.
  - Closed sets.
  - Topological spaces.
  - Tychonoff's Theorem and Heine-Borel's Theorem.
- A mechanism for automatic differentiation/derivation.
- The compatibility between the vectors of MATHEMATICAL COMPONENTS and COQUELICOT's structures.

The Mathematical Components Analysis library

- Classical analysis.
- Inspired from Coquelicot.
- Compatible with Mathematical Components.
- Includes various facilities:
  - Notations for limits and convergence, based on filter inference:
    `f @ x --> y`, `lim (f @ x)`, `cvg (f @ +oo)`, `u --> -oo`.
  - A differential function, together with a notation: `'d_x f`.
  - Equational Bachmann-Landau notations:
    `f = g +o_F e`, `f =O_F e`.
  - Automatic proof of positivity.
  - Automatic differentiation/derivation.
  - A set of tactics for delayed instantiation of existential witnesses (`near`).

Hierarchy of topological structures



Comparison

Lines of code: [2]

	Using COQUELICOT	Using our library
LaSalle's invariance principle	$\sim$ 370	$\sim$ 370
Inverted pendulum	$\sim$ 980	$\sim$ 900

Tactics:

	Using COQUELICOT	Using our library
`ring`	✓	✓ [3]
`field`	✓	almost [3]
`lra`	✓	✗
`near`	✗	✓

[2] Not counting the parts that were integrated to our library.

[3] Thanks to Pierre-Yves Strub:

https://github.com/jasmin-lang/jasmin/blob/master/proofs/3rdparty/ssrring.v

Conclusion

Contributions:

- A case study involving standard tools:
  - An important theorem in stability analysis.
  - A common benchmark for control techniques.
- A new library based on what we learnt on the way.

Potential continuations:

- A certified implementation?
- Verification of the equations of motion.

Bibliography

Boldo, S., Lelay, C., and Melquiond, G. (2015).
Coquelicot: A User-Friendly Library of Real Analysis for Coq.
Mathematics in Computer Science, 9(1):41–62.

Cohen, C. and Rouhling, D. (2017).
A Formal Proof in Coq of LaSalle's Invariance Principle.
In Ayala-Rincón, M. and Muñoz, C. A., editors, Interactive Theorem Proving - 8th International Conference, ITP 2017, Brasília, Brazil, September 26-29, 2017, Proceedings, volume 10499 of Lecture Notes in Computer Science, pages 148–163. Springer.

LaSalle, J. (1960).
Some Extensions of Liapunov's Second Method.
IRE Transactions on Circuit Theory, 7(4):520–527.

Lozano, R., Fantoni, I., and Block, D. (2000).
Stabilization of the inverted pendulum around its homoclinic orbit.
Systems & Control Letters, 40(3):197–204.

Rouhling, D. (2018).
A Formal Proof in Coq of a Control Function for the Inverted Pendulum.
In Andronick, J. and Felty, A. P., editors, Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2018, Los Angeles, CA, USA, January 8-9, 2018, pages 28–41. ACM.

Axioms used in the formalization

- Extensionality axioms:

```
propext : forall (P Q : Prop), (P <-> Q) -> (P = Q).
funext : forall (T U : Type) (f g : T -> U),
         (forall x, f x = g x) -> f = g.
```

- Classical axioms:

```
pselect : forall (P : Prop), {P} + {~P}.
gen_choiceMixin : forall (T : Type), Choice.mixin_of T.
```

Canonical structures for filter inference

Three structures:

- `filter_on_term X Y`: structure that records terms $x : X$ with a filter in Y.
  Allows to infer the canonical filter associated to a term by looking at its type.
- `filteredType U`: interface type for types whose elements represent filters on `U`.
- `Filtered.source Y Z`: structure that records types X such that there is a function mapping functions of type `X -> Y` to filters on `Z`.
  Allows to infer the canonical filter associated to a function by looking at its source type.

Filter-based compactness

Clustering generalizes to filters the notion of limit point.

$$\texttt{cluster}\ F := \{p \in U \mid \forall A \in F, \forall B \text{ neighbourhood of } p, A \cap B \neq \emptyset\}$$

A is compact iff every proper filter on A clusters in A.

```
Definition compact (A : set U) := forall (F : set (set U)),
  F A -> ProperFilter F -> A `&` cluster F !=set0.
```