Formal Verification of Stability Properties of Cyber-physical Systems

Matthew Chan, Daniel Ricketts, Sorin Lerner, Gregory Malecha
University of California, San Diego
mattchan@ucsd.edu, daricket@cs.ucsd.edu, lerner@cs.ucsd.edu, gmalecha@eng.ucsd.edu

1. Introduction

We increasingly rely on computers to interact with the physical world for us. At the large end, software underlies the control systems of commercial aircraft and power plants, and at the small end it controls medical devices and hobbyist UAVs. The failure of any of these systems can have severe consequences which are often measured in the loss of human lives. Formal verification has proven a promising approach to achieving very strong guarantees in more classic areas of computer science. In this work we present an overview of our experiences formalizing stability properties of cyber-physical systems (CPSs) using the Coq proof assistant.

In particular, we describe and contrast two approaches for proving the stability of the linear, one-dimensional proportional controller (P-controller) depicted in Figure 1. This system runs in a loop where the controller sets the velocity (v) of the system and then the position (x) evolves continuously according to the differential equation $\dot{x} = v$ for at most Δ time while v remains constant. The goal of the controller is to move the system to $x = 0$.

2. System Specification

We carry out our verification on top of the VeriDrone project [1], a formalization of cyber-physical systems in the Coq proof assistant. VeriDrone expresses CPSs and their properties uniformly in a linear temporal logic deeply embedded inside of Coq. For example, the specification of our system take the following form:

```
Init ∧ □ (Discr ∨ World)
```

The first conjunct (`Init`) is a predicate over the initial state of the system. The second conjunct (□(. . .)) expresses the transitions of the system. It specifies that all temporally adjacent states are related by either `Discr`, a discrete transition that is morally of type `State → State → Prop`, or `World`, a continuous transition of the physical world expressed with predicates over the state and the time-derivative of variables in the state. The discrete and continuous transitions for our system are the following:

```
Definition Discr := v! = −x / Δ ∧ T! ≤ Δ ∧ t! = t ∧ x! = x.

Definition World :=
  Cont (fun ∂ ⇒ ∂x = v ∧ ∂v = 0 ∧ ∂t = 1 ∧ ∂T = −1 ∧ T! > 0).
```

Note that `x!` represents the value of `x` in the next state.

In addition to the variables `x` and `v`, we need two additional variables to fully specify our system: `t` tracks the current time of the system, and `T` acts as a stopwatch which is reset every time the discrete system runs. In the definition of]oqeDiscr, `v! = −x/Δ` specifies the P controller, `T! ≤ Δ` specifies the resetting of the stopwatch, and `t!=t ∧ x!=x` specify that the continuous variables are unchanged. In the definition of `World`, `∂ x = v` specifies the differential equation $\dot{x} = v$, `∂ v = 0` specifies that the velocity remains constant during continuous transitions, `∂ t = 1` specifies that `t` tracks the current time, `∂ T = −1` count down at the same rate that time advances. The final assertion (`T! ≥ 0`) ensures that the continuous dynamics can not evolve when the stopwatch has no time left.

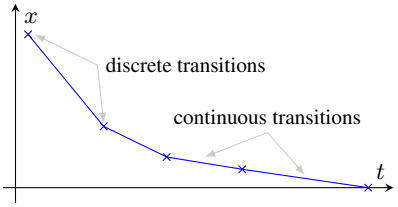

Figure 1. Visualizing the proportional controller.

3. Stability

In this work we use the VeriDrone system to reason about "stability" in the control theory sense. Stability is a classic property that captures a temporal notion of boundedness in terms of distance from a goal point. We will focus on two notions of stability which are shown graphically in Figure 2:

1. **Lyapunov stability** states that if the system starts within $\beta > 0$ from the goal, then it will stay within α of the goal point for all time. In our temporal logic, `x` is Lyapunov stable if

   ```
   Definition LyapunovStable (x : Term) : Formula :=
     Forall α : R, α > 0 → (* boundary *)
       Exists β : R, β > 0 ∧ (* start *)
         (| x| < β) → □(|x| < α).
   ```

   Note that the universal quantification of the boundary implies that if the system starts arbitrarily close, then it will remain arbitrarily close. However, the definition does not imply that the system converges. For example, the system that stays a constant distance from the goal is Lyapunov stable.

2. **Exponential stability** is a refinement of stability that guarantees that the system converges to the goal exponentially fast. The formal definition is the following:

   ```
   Definition ExpStable (x : Term) : Formula :=
     Exists α : R, α > 0 ∧ Exists β : R, β > 0 ∧
     Exists x0: R, x = x0 ∧ Exists t0 : R, t0 = t ∧
       □(|x| ≤ (α ∗ |x0| ∗ e^(−β ∗ (t − t0)))).
   ```

   The first line existentially quantifies both α and β, and the second line captures the initial configuration (time in t_0 and the value of x in x_0). In this definition t is the global time which increases monotonically, i.e. we must prove that our system is confined by *a single* exponential regardless of the number of discrete and continuous transitions it takes.

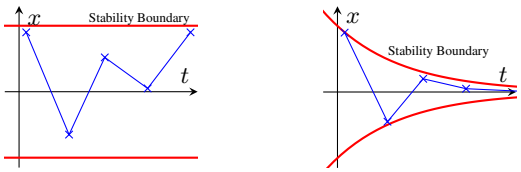

Figure 2. Visualization for Lyapunov and exponential stability.

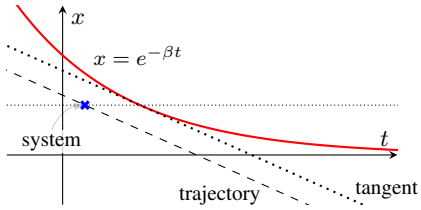

Figure 3. Graphical "proof" of exponential stability.

4. Proving Stability Directly

We began by proving that our system is Lyapunov stable directly. While cumbersome due to the heavy use of arithmetic, the proof is quite similar to those performed when verifying safety properties of monitors in VeriDrone [1].

The proof proceeds by the following rule which describes temporal logic induction. We divide the safe region into two parts: one where $\mathtt{x} \geq 0$ and the other where $\mathtt{x} \leq 0$. Because the controller is symmetric around 0, it suffices to prove just the case where $\mathtt{x} \geq 0$ and use a substitution of $x \mapsto -x$ to prove the $\mathtt{x} \leq 0$ case. Since the controller always directs the system towards 0, showing that the upper bound is not violated is simple. The difficulty lies in showing that the system never crosses 0, which guarantees that it will never cross the lower bound. We argue this using differential induction [2], which allows us to prove that an inequality is preserved by a continuous transition if the inequality holds between the derivative of the two sides. For example, to show that $x \geq 0$, we need to strengthen the statement to state that $x \geq -v \cdot T$, i.e. the distance from 0 is at least the distance the system will travel before the stopwatch expires. Differential induction allows us to prove this by showing the following

$$\dot{x} = v \geq \frac{d}{dt}\left[-v \cdot T\right] = -\dot{v} \cdot T + -v \cdot \dot{T} = v$$

Exponential Stability We are currently in the process of proving the same controller exponentially stable. The proof generally follows the same structure except that we must now show the system approaches the goal at least as quickly as the exponential that is bounding it. Figure 3 provides a graphical sketch of the reasoning.

As with the Lyapunov stability proof, we will simply consider positive values of x and use symmetry to argue the lower bound. We pick $\alpha = 1$ and $\beta = \frac{1}{\delta}$. The choice of β here is essential for this particular controller. In particular, note that the velocity chosen by the controller $-\frac{x}{\delta}$ is exactly the slope of the exponential *where it crosses the horizontal line through* x. Therefore, the trajectory of the system is parallel to the tangent line. Since the exponential is concave up it is always greater than its tangent line (a fact we proved in Coq). By transitivity the system is always below the exponential during continuous transitions.

To extend this proof to the entire system we must incorporate the behavior of the discrete transition, in particular ensuring that it sets the velocity appropriately. We have found this to be difficult due to the structure of our proofs which mention explicitly the state at the beginning of the continuous transition. When included in the global proof, we must manifest these values explicitly and connect them to the system, which has been more difficult than we anticipated.

5. Stability Proofs using Lyapunov Functions

In addition to the visually appealing approach described in Section 4, we have also begun exploring the use of Lyapunov functions to prove stability. Lyapunov functions allow one to prove the stability of a system using a notion of the energy of the system which decreases over time until it reaches an equilibrium. More precisely, a Lyapunov function for a one-dimensional system with an equilibrium at 0 is a function $V : \mathcal{R} \to \mathcal{R}$ such that $V(0) = 0$, $V(x) > 0$ for $x \neq 0$, and satisfying some condition on the time-derivative, $\dot{V}(x)$. The condition on $\dot{V}(x)$ depends on the particular notion of stability one would like to prove. For Lyapunov stability and exponential stability, the conditions are $\dot{V}(x) \leq 0$ and $\dot{V}(x) \leq \alpha V(x)$ for $\alpha < 0$, respectively.

For a purely continuous proportional controller, specified by the differential equation $\dot{x} = -x$, the function $V(x) = \frac{1}{2}x^2$ serves as a Lyapunov function satisfying all three conditions on the time-derivative and thus establishes both notions of stability. To see this, note that $\dot{V}(x) = x\dot{x} = -x^2$. Unfortunately, for our hybrid P-controller, the same computation gives us $\dot{V}(x) = x\dot{x} = x * v$. However, we can prove that our P-controller is a refinement of the system in which $\dot{x} = \frac{-x}{\Delta-(t-T)}$. In other words, the velocity is proportional to the distance from the equilibrium and inversely proportional to the time remaining before the controller must run again. Using this abstraction, it is possible to establish all three conditions on the time derivative of V and hence both notions of stability. Thus far, we have only completed the Coq proof of Lyapunov stability of our P-controller using a Lyapunov function.

6. Discussion

It is interesting to contrast the process of formalizing the graphically-inspired proofs with the proofs based on Lyapunov functions. Though the graphically-inspired proofs appear to be more intuitive on paper, our current experience suggests that they are more challenging to formalize due to a lack of abstraction of time. Lyapunov functions provide this abstraction and hence seem to provide an cleaner path to a fully formal proof. However, the time-abstraction seems a bit tenuous when using Lyapunov functions for a hybrid system rather than a purely continuous one. For example, our approach above implicitly relies on being able to completely solve the differential equations, something that is often best to avoid. This is especially the case since differential equations are often not equations at all, but rather inequalities. More investigation is necessary to determine how both of these approaches scale to more complex hybrid systems.

Our initial work on stability has focused on a one-dimensional P-controller. P-controllers are the first step on the path to PID (Proportional-Integral-Derivative) controllers, which will allow expressing more exciting behaviors such as oscillatory convergence to a goal. While still relatively simple, PID controllers form the vast majority of controllers in practice. Ideally, the construction of a verified PID controller would follow from a layered composition, allowing for the separate construction of the I and D controllers.

References

[1] Daniel Ricketts and Gregory Malecha and Mario M. Alvarez and Vignesh Gowda and Sorin Lerner. Towards Verification of Hybrid Systems in a Foundational Proof Assistant. In *MEMOCODE '15*, 2015.

[2] André Platzer. A temporal dynamic logic for verifying hybrid system invariants. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 4514 of *LNCS*, pages 457–471. Springer, 2007.