# Sound Invariant Generation for Continuous and Hybrid Systems

**André Platzer (PI), Nathan Fulton, Andrew Sogokon** NSF CNS-1739629

Computer Science Department, Carnegie Mellon University

## Objectives

- Develop a sound **automatic invariant generator** for continuous systems (ODEs) incorporating many existing results and new approaches under a unified framework.
- Integrate the continuous invariant generator into the **KeYmaera X** proof assistant.
- Explore synergies between continuous and discrete invariant generation to **improve proof automation** in KeYmaera X.
- Apply the improved system to automatically **verify quadrotor software**.

## Introduction

**KeYmaera X** is an interactive proof assistant for **differential dynamic logic** ($d\mathcal{L}$) which allows one to *specify* and *deductively verify* properties such as *safety* and *liveness* in hybrid systems.

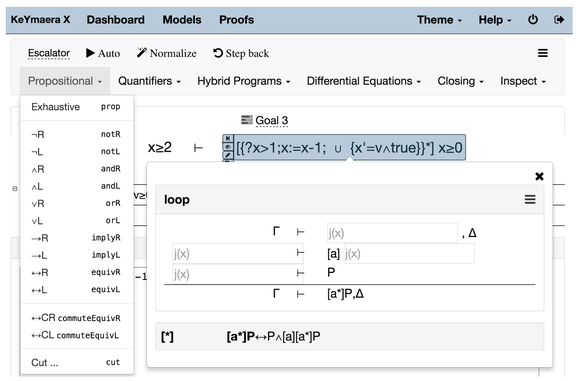



Figure: KeYmaera X web interface

## Problem

**Using verification tools is hard** even for experts. The amount of **manual effort and ingenuity** required to be effective presents a **practical bottleneck**.

## Continuous invariants

A **continuous invariant** is a sound over-approximation of the reachable set of a continuous system (given by ODEs) from some initial set of states. Continuous invariants play a **major role in deductive proofs** of safety properties in hybrid systems, but finding them can be very challenging.

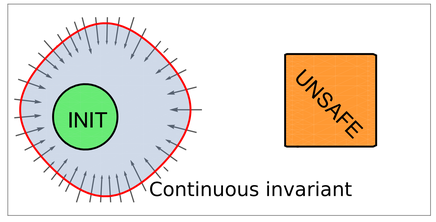



Figure: Safety with continuous invariants

Ability to **automatically generate continuous invariants** would radically enhance the user experience and practicality of KeYmaera X.

**There is no "silver bullet"** for the problem of continuous invariant generation. However, many methods can **target special classes of problems** by exploiting their structure.

## Approach

Our approach aims to **leverage specialized invariant generation methods** for continuous systems. To do this, we:

1. **Classify problems** based on certain pertinent criteria, such as the **dimension** of the system and the **"kind" of functions appearing in the ODEs**, e.g. *constant, linear, affine, non-linear*, etc.
2. Organize invariant generation methods into effective **strategies that exploit structure** in the verification problem.

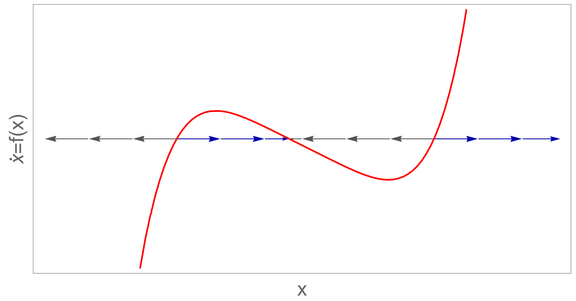
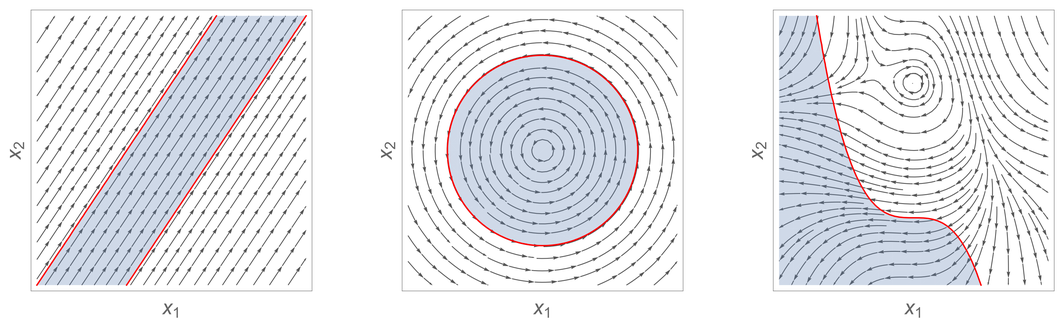
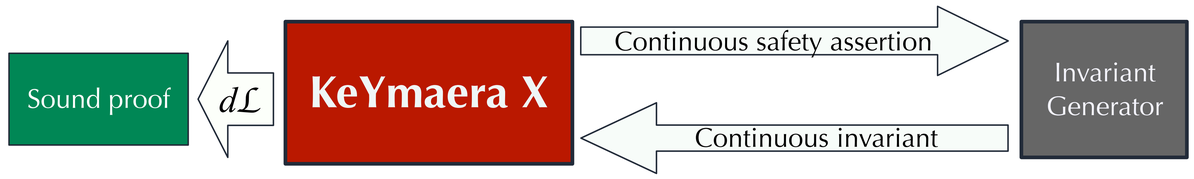



Figure: Continuous invariants in a one-dimensional system




(a) Constant system (b) Linear system (c) Nonlinear system

Figure: Continuous invariants in planar systems




## Soundness

The generated invariants are to be **checked** using the $d\mathcal{L}$ proof calculus inside KeYmaera X. This gives us a much greater degree of confidence in the correctness of the proof. KeYmaera X compares favorably to other verification tools due to its small (less than 2000 lines of code) trusted core.

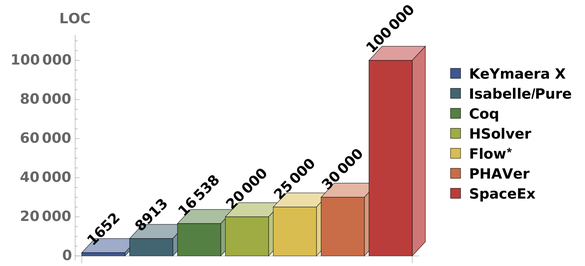



Figure: Lines of critical code in popular verification tools

## Applications

Our goal is to develop a suite of formally verified control software for **quadrotors**, e.g. verified collision avoidance, ground avoidance, geo-fencing, and auto-pilot algorithms.

The ultimate goal of our work is to design proof search algorithms that *automatically* verify industrially relevant models.

## KeYmaera X

`http://www.keymaeraX.org`

Carnegie Mellon University