Tool Library Workshop

Marcus Lucas and Paulo Tabuada

Cyber-Physical Systems Laboratory
Department of Electrical and Computer Engineering
University of California at Los Angeles

Background

- Algorithm verification is vital to the development of cyber-physical systems
  - Safety Critical
  - Expensive to test
  - Difficult to reason about
- Verification tools are generally good for the research community
  - Aids reproducability of results
  - Facilitates benchmarking
  - Improves quality of research/knowledge dissemination

Issues

- But verification can be a lot of extra work
  - Finding, understanding, and then using tools is non-trivial
- System designers need to better understand verification and what types of problems can be verified
- Low level of knowledge and confidence regarding existing tools
- Design process is not understood in terms of verification

Goals

- Embed verification into the engineering design process as a first-order objective
- Communicate the circumstances under which verification should be applied
- Normalize the publication and support of tools created by community members
- Grow confidence in verification's utility

How to Get There?

- Reduce barriers to entry through the CPS-VO
  - Tool Repository
  - Tutorials
  - Integrated tools
- Encourage student engagement
- Create benchmarks and standard models for evaluating tools

Verification Tool Library

Verification Tool Library

CPS-VO » CPS TOOLS AND DESIGN STUDIOS » VERIFICATION TOOL LIBRARY » SEARCH FOR TOOLS

Search for Tools

EDIT GROUP | TRACK | TAXONOMY | BROADCAST | PANELS | GROUP STATS

- Home
- README
- Search for Tools
- Browse Tool Library
- Members
- Tutorials
- Tutorial - CORA (Hidden)
- Tutorial - SpaceEx
- Tutorial - S-TALIRO (Hidden)
- Bulk Edit (Hidden)
- Taxonomic Information
- Forums
- Files

COLLABORATE

Search results for Verification Tool Library

Search for:

Types of systems handled

<Any>

Techniques used

Abstractions
Agent and Mode Architecture
Ant Colony Optimization
Approximate (bi)simulation
Approximation of solutions to Hamilton-Jacobi partial differential equati
Bisimulation minimization
Boolean Equation Systems
Bounded Model Checking
Branching-Time Temporal Logics

Type of verification

Bisimulation generation
Cost optimal reachability analysi
Description/modeling language
Falsification
Invariant analysis
Invariant set computation
Model checking
Modeling
Parameter synthesis

Interface

C++
Integrated to SpaceEx
Java
Matlab
Model editor integrated
OMNet++
Own interface
Own language
Python

Termination guarantees

Guaranteed to terminate
Not guaranteed to terminate

OS

Linux
Mac OS X
Solaris
Windows

Free

Contact the author
Free

Apply **Reset**

GTTS

WIKIPAGE

Submitted by nanli on Tue, 09/25/2018 - 2:53pm

Automotive Testing Simulation Validation and Verification Autonomous Driving Simulator test generation verification and validation Tool Free Java Linux Mac OS X Solaris Windows Simulation

CyPhyLab UCLA

Verification Tool Library

- VTL documents mature tools available to the CPS community
- Searchable by taxonomic terms and includes a wiki
- Links to tutorials, integrated tools, more thorough documentation

Verification Tool Library

Repository Demo

Verification Tool Library

Links

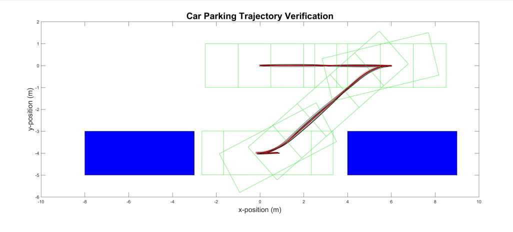
- Tool Library: https://cps-vo.org/group/verification_tools/
- Submission Form: https://cps-vo.org/group/tools/submit

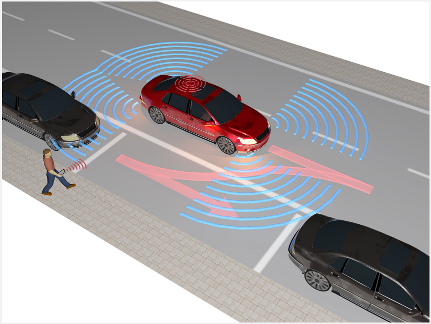
Tutorials

- SpaceEx
  - Only handles hybrid systems with affine continuous dynamics
  - Particularly accessible to high-school & undergraduate students
- CORA
  - Formal verification of non-linear systems
- S-TALIRO
  - Falsification of logical constraints

Tutorials

Autonomous Parking Example





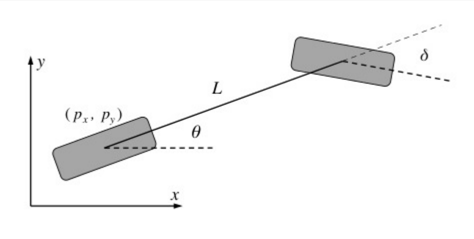
- Finding and following parking trajectories is a basic requirement of autonomous cars

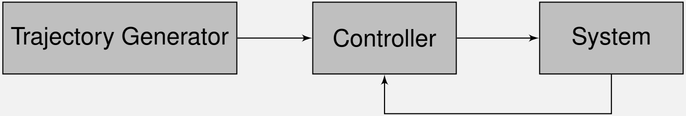
Tutorials

Autonomous Parking Example

Kinematic Bicycle Model of Car

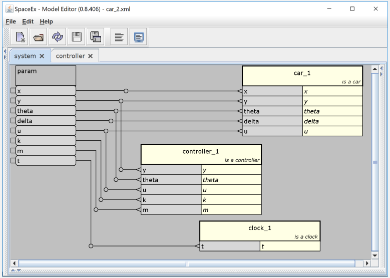
$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \frac{v}{L}\sin(\delta)$$
$$\dot{\delta} = u$$

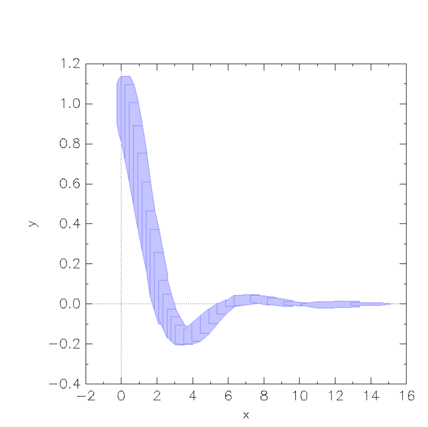




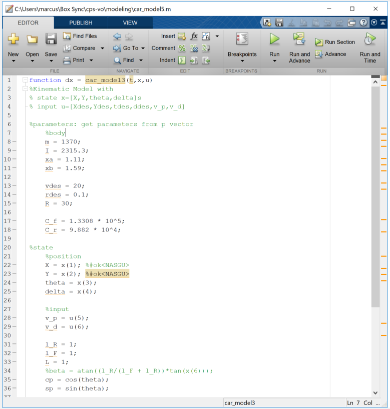
- What if you have a valid trajectory, and want to test if your controller tracks it properly?

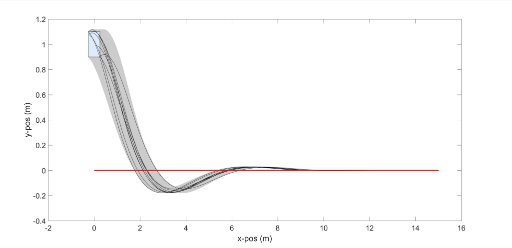
- SpaceEx can be used to evaluate simple control problems such as stabilizing to a horizontal trajectory.



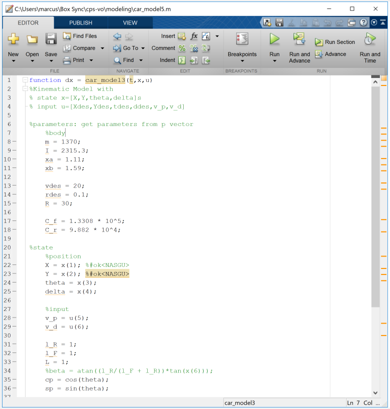


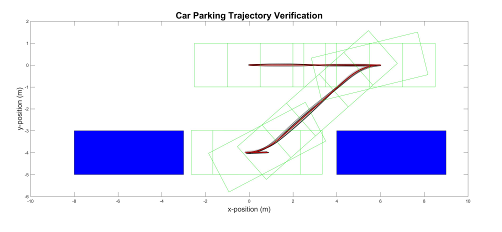
- CORA can do the same thing



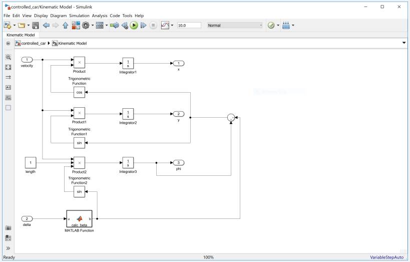


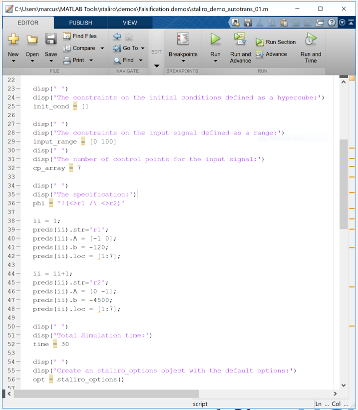
- But it is also capable of analyzing non-linear systems, making it much more useful for verifying systems like the full parking controller.





- S-TALIRO does not provide formal guarantees, but can be used to falsify the same classes of problems

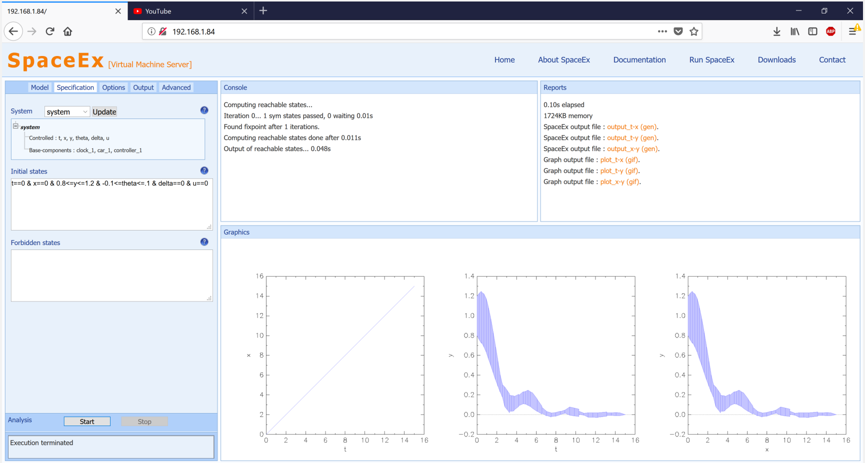




CyPhyLab UCLA

Embedded Tools

- Currently have SpaceEx running



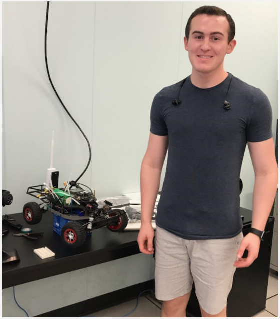
Running Tools on the CPS-VO

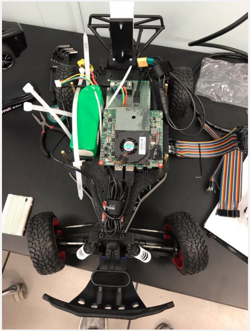
SpaceEx Example

SpaceEx Demo

Student Engagement

- Want to normalize the use of verification tools in design process
- Targeting undergraduate and even high-school students
- Ideal tools integrate easily with familiar software (i.e. Matlab/Simulink)





Future Work

Benchmarks

- We're announcing a new benchmarking competition!
- Expansion of the "Friendly Competition" held during the ARCH Workshop
  - https://cps-vo.org/group/ARCH
- Tentative structure:
  - Participants integrate verification/synthesis tools into VO
  - Test period of about a month
  - Competition organizers evaluate tools against benchmarks using VO resources
- Competition results will be advertised through the CPS-VO