Formal Methods, Machine Learning, and Cyber-Physical Systems

Sanjit A. Seshia
Professor
UC Berkeley

ATVA 2018 Tutorial
October 7, 2018
Connections in this Lecture

1. Formal Methods

2. Cyber-Physical Systems

1. Machine Learning

Formal Methods

Cyber-Physical Systems
Synthesis of Controllers/Plans for Cyber-Physical Systems

Joint work with:
Ankush Desai, Shromona Ghosh, Pierluigi Nuzzo, Indranil Saha, Yasser Shoukry, Vijay Kumar, George Pappas, Shaz Qadeer, Alberto Sangiovanni-Vincentelli, Paulo Tabuada

[Shoukry et al., HSCC 2017, CDC 2017, Proc. IEEE 2018; Desai et al. ICCPS 2017; Saha et al., IROS 2014]
Goal: Correct-by-Construction Motion Planning for Robotics

Declarative Task Specification (Temporal Logic) [+ Examples]

Compiler

Component Library

Safe, Correct Executable Software

Drone executing plan in ROS/Gazebo Simulator
Challenges

• Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control

• Implementation in software on top of networked robotics platforms

• Dealing with untrusted components, uncertainty in environment, and ML-based perception
Motion Planning Problem

- Problem: Compute an obstacle-free trajectory that is feasible with the given robot dynamics within the target workspace.
- Core problem for autonomous vehicles
- Challenges:
  - Handling high-level tasks (captured by formalisms such as Linear Temporal Logic)
  - Handling distributed team of robots
  - Handling uncertainty

[Shoukry et al., HSCC 2017, CDC 2017, Proc. IEEE 2018; Saha et al., IROS 2014]
Related Work: Sampling-based Methods


• Do not handle high-level specifications given by formal models, such as LTL.
Related Work: Abstraction Based Techniques

Discrete → Continuous


- ...
Related Work: Optimization Based Techniques

Discrete $\rightarrow$ Continuous

- ...
Two Extremes

Discrete $\leftrightarrow$ Continuous (Abstraction based)

\[ x^{(t+1)} = Ax^{(t)} + Bu^{(t)} \]
\[ y^{(t)} = Cx^{(t)} \]

Scales poorly as the number of continuous states increases

Discrete $\rightarrow$ Continuous (Optimization based)

\[ x^{(t+1)} = Ax^{(t)} + Bu^{(t)} \]
\[ y^{(t)} = Cx^{(t)} \]

\[
\begin{align*}
\min & \quad J(x) \\
\text{subject to:} & \quad x^{(t+1)} = Ax^{(t)} + Bu^{(t)} \\
& \quad -b_1^{(t)} + b_2^{(t+1)} + b_9^{(t+1)} = 0 \\
& \quad -b_2^{(t)} + b_1^{(t+1)} + b_3^{(t+1)} = 0 \\
& \quad x^{(0)} - \bar{x} = Mb_1^{(0)}
\end{align*}
\]
Satisfiability Modulo Convex Programming [Shoukry et al., HSCC 2017]

- **SAT Solvers**: central tools in formal methods to reason about discrete dynamics.
- **Convex Optimization**: central tools in control theory to reason about continuous dynamics.
- Approach for hybrid dynamics: **SAT Solving + Convex Programming**
Motivating Example: Obstacle Avoidance

- Given:
  - Robot dynamics (linear)
  - Input constraints
  - Initial and final states
- Generate the input sequence (controller)
Motivating Example: Obstacle Avoidance

- Given:
  - Robot dynamics (linear)
  - Input constraints
  - Initial and final states
  - Generate the input sequence (controller)
Motivating Example: Obstacle Avoidance

\[ \varphi ::= b^0_{\text{start}} \wedge b^L_{\text{goal}} \]

(Initial partition)

(Goal partition)
Motivating Example: Obstacle Avoidance

\( \varphi ::= b^0_{\text{start}} \)
\( \land b^L_{\text{goal}} \)
\( \land b_i^j \rightarrow \bigvee_{i' \in \Pi(i)} b_{i'}^{j+1} \)

(Initial partition)
(Goal partition)
(Adjacency constraints)

\( \land \sum_{i=1}^{m} b_i^j = 1 \)
\( \land x^{j+1} = Ax^j + Bu^j \)
\( \land \|u^j\| \leq \bar{u} \)
\( \land x^0 = x \)
\( \land b_i^j \rightarrow x^j \in \mathcal{P}_i \)

\( \forall j \in \{0, \ldots, L - 1\} \)
\( \forall j \in \{0, \ldots, L - 1\} \)
\( \forall j \in \{0, \ldots, L - 1\} \)
\( \forall j \in \{0, \ldots, L - 1\} \)
\( \forall j \in \{0, \ldots, L - 1\} \)

(Initial state)
(Robot dynamics)
(Input constraints)
(Region constraints)

\( \phi(b) \)
\( g(x) \leq 0 \)
\( \phi'(b) \lor g'(x) \leq 0 \)
Motivating Example: Obstacle Avoidance

Definition: **Monotone SMC Formula**

\[
\text{formula ::= } \{ \text{clause} \land \}^* \text{clause} \\
\text{clause ::= } (\{ \text{literal} \lor \}^* \text{literal}) | \text{pB\_predicate} \\
\text{literal ::= } \text{bool\_var} | \neg \text{bool\_var} | \top | \bot | \\
\text{conv\_constraint ::= } \text{equation} | \text{inequality} \\
\text{equation ::= } \text{affine\_function} = 0 \\
\text{inequality ::= } \text{convex\_function} \text{ relation } 0 \\
\text{relation ::= } < | \leq
\]
Applications: Controller Synthesis

Obstacle Avoidance

\[ \phi :: = \text{initial partition} \]
\[ \land \text{goal partition} \]
\[ \land \text{adjacency partition} \]
\[ \land \text{robot dynamics} \]
\[ \land \text{input constraints} \]
\[ \land \text{initial state} \]
\[ \land \text{region constraints} \]

LTL Motion Planning

\[ \phi :: = \text{LTL BMC encoding} \]
\[ \land \text{adjacency partition} \]
\[ \land \text{robot dynamics} \]
\[ \land \text{input constraints} \]
\[ \land \text{initial state} \]
\[ \land \text{region constraints} \]

Multi-robot Motion Planning

\[ \phi :: = \text{LTL BMC encoding} \]
\[ \land \text{adjacency partition} \]
\[ \land \text{robot dynamics} \]
\[ \land \text{input constraints} \]
\[ \land \text{initial state} \]
\[ \land \text{region constraints} \]
\[ \land (x_{\text{robot 1}} - x_{\text{robot 2}} > \epsilon) \]
\[ \lor (x_{\text{robot 2}} - x_{\text{robot 1}} > \epsilon) \]
\[ \lor (y_{\text{robot 1}} - y_{\text{robot 2}} > \epsilon) \]
\[ \lor (y_{\text{robot 2}} - y_{\text{robot 1}} > \epsilon) \]
Applications: CPS Security

Secure State Estimation

Secure Localization

Secure Traffic Routing

\[ \phi(b, X) := \bigwedge_{i=1}^{p} \left( -b_i \Rightarrow \|Y_i - \mathcal{O}_i X\|_2^2 \leq 0 \right) \]
\[ \bigwedge \left( \sum_{i=1}^{p} b_i \leq s \right). \]

\[ (i, j) \in \mathcal{E} \]
\[ -b_{ij} \Rightarrow \|g_{ij}^T Z g_{ij} - m_{ij}^2\| \leq 0 \]
\[ Z = \begin{bmatrix} Y & X^T \\ 0 & 1 \end{bmatrix} \succeq 0 \]
\[ \bigwedge (\sum_{(i, j) \in \mathcal{E}} b_{ij} \leq s) . \]

\[ b_i = (0, 0) \quad \text{ith car is honest} \]
\[ b_i = (0, 1) \quad \text{ith car physically exist but dishonest} \]
\[ b_i = (1, 0) \quad \text{ith car is Sybil} \]

if car \( i \) does not report car \( j \)

then both \( i \) and \( j \) can not be honest

for the cluster \( \mathcal{I} \) of honest cars

\[ \frac{1}{N} \sum_{k=1}^{N} (v_{\mathcal{I}}(k) - v_{\text{loop}}(k))(v_{\mathcal{I}}(k) - v_{\text{loop}}(k))^T \]
\[ \succeq \frac{1}{N} \sum_{k=1}^{N} (v_{\mathcal{I}}(k) - v_{\text{loop}}(k))(v_{\mathcal{I}}(k) - v_{\text{loop}}(k))^T \]
The satisfiability of the monotone SMC formula can always be cast as a feasibility problem for a finite disjunction of convex constraints.

Theorem:
Any feasibility problem for a finite disjunction of convex programs can be posed as a satisfiability problem for a monotone SMC formula.
Complexity of Solving SMC

Complexity = \#\text{Iterations} \times (\text{Time}_{\phi(b)} + \text{Time}_{g(\pi) \leq 0})

\((g_1(x) \leq 0) \land \phi_1(b)\)

\lor (g_2(x) \leq 0) \land \phi_2(b)\)

\lor ...

\lor (g_k(x) \leq 0) \land \phi_k(b)\)

Monotone SMC Formula

Reduce the number of iterations?
Solving SMC

Key idea: abstraction-based search

Complexity = \#Iterations \times (\text{Time}_{\phi(b)} + \text{Time}_{g(x) \leq 0})

Monotone SMC Formula:

\[(g_1(x) \leq 0) \land \phi_1(b) \lor (g_2(x) \leq 0) \land \phi_2(b) \lor \ldots \lor (g_k(x) \leq 0) \land \phi_k(b)\]
Boolean Expansion

• Step 0:
  • Given a monotone SMC formula $\phi(b, x)$, construct its Boolean expansion:

$$\phi'(a, b, x) = \phi_B \land \bigwedge_{i=1}^{\vert C \vert} (a_i \implies (g_i(x) \leq 0))$$

where $\phi_B$ is obtained from $\phi(b, x)$ by replacing each convex constraint $g_i(x) \leq 0$ with a Boolean variable $a_i$

Lemma:
$\phi(b, x)$ and $\phi'(a, b, x)$ are equi-satisfiable.
Solver Operation

• Step I: Solve the “Boolean abstraction” of the problem.

• Step II: Extract which convex constraints are active.

• Step III: Check the satisfiability of:

\[
\min_{a \in \mathbb{R}^n} 1 \\
\text{s.t. } g_j(x) \leq 0 \\
\quad j \in \{1, \ldots, |C|, a_j = 1\}
\]

• Step IV: Generate UNSAT certificate:

\[
\phi_{\text{trivial-ce}} = \bigvee_{j \in \text{supp}(a)} \neg a_j
\]
Solver Operation

\[ \phi'(a, b, x) = \phi_B \land \bigwedge_{i=1}^{\left| \mathcal{C} \right|} (a_i \implies (g_i(x) \leq 0)) \]

Performance?

Complexity = \#Iterations \times (Time_{\phi(b)} + Time_{g(w) \leq 0})
Minimal UNSAT Certificate

• Finding a minimal UNSAT certificate is equivalent to finding an Irreducible Infeasible Set (IIS) of:

$$\min_{x \in \mathbb{R}^n} 1 \quad \text{s.t.} \quad g_j(x) \leq 0 \quad j \in \{1, \ldots, |C|, a_j = 1\}$$

• NP-hard in general.

• However, our problem has more structure than general IIS problems.

• Can we exploit the structure of the Boolean constraints to help find the IIS of the convex program?
### Summary of UNSAT certificates

<table>
<thead>
<tr>
<th>UNSAT Certificate</th>
<th>Minimal</th>
<th>Complexity (number of convex problems)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial</td>
<td>No</td>
<td>Constant</td>
</tr>
<tr>
<td>IIS</td>
<td>Yes</td>
<td>Exponential</td>
</tr>
<tr>
<td>Sum of Slacks</td>
<td>Yes*</td>
<td>Linear</td>
</tr>
<tr>
<td>Minimum Prefix</td>
<td>Yes*</td>
<td>Constant</td>
</tr>
</tbody>
</table>

* under reasonable technical assumptions

\[
(g_1(x) \leq 0) \land \phi_1(b) \\
\lor (g_2(x) \leq 0) \land \phi_2(b) \\
\lor \ldots \\
\lor (g_k(x) \leq 0) \land \phi_k(b)
\]

**Monotone SMC Formula**

**Complexity** = \#Iterations × \((\text{Time}_{\phi(b)} + \text{Time}_{g(x) \leq 0})\)

“small” polynomial
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning

\[ g_1(x) \leq 0 \]
\[ g_2(x) \leq 0 \]
\[ g_3(x) \leq 0 \]
\[ g_4(x) \leq 0 \]
\[ g_5(x) \leq 0 \]
\[ g_6(x) \leq 0 \]
\[ g_7(x) \leq 0 \]
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad 1 \\
\text{s.t.} & \quad (g_1(x) \leq 0) \land (g_3(x) \leq 0) \land (g_2(x) \leq 0) \land (g_4(x) \leq 0) \land (g_6(x) \leq 0) \\
& \quad (g_5(x) \leq 0) \land (g_7(x) \leq 0)
\end{align*}
\]
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} 1 \quad &\text{s.t.} \quad (g_1(x) \leq 0) \land (g_3(x) \leq 0) \land (g_2(x) \leq 0) \land (g_4(x) \leq 0) \land (g_6(x) \leq 0) \\
&\quad \land (g_5(x) \leq 0) \land (g_7(x) \leq 0)
\end{align*}
\]
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning

Key idea: Find the "shortest" UNSAT certificate (e.g., the shortest prefix witness of a safety property violation).
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning

Key idea: Find the “shortest” UNSAT certificate (e.g., the shortest prefix witness of a safety property violation).
## Minimum Prefix Certificates

**Definition: Positively Ordered Unate (POU) Function**

A function is said to be POU with respect to an ordering of its variables if for all values of $b_i$, we have

\[
\begin{align*}
    u(0, \ldots, 0) &\leq u(b_1, 0, \ldots, 0) \leq u(b_1, b_2, 0, \ldots, 0) \leq \ldots \\
    \ldots &\leq u(b_1, b_2, \ldots, b_{m-1}, 0) \leq u(b_1, b_2, \ldots, b_{m-1}, b_m).
\end{align*}
\]

**Theorem:**

Consider a monotone SMC formula $\phi(b, x)$ and its Boolean abstraction $\phi_B$.

If:

1. $\phi_B$ is positively ordered unate
2. the domain of the real variables is bounded

then minimal UNSAT certificates exist and can be computed in constant time.
Minimum Prefix Certificates

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad 1 \quad \text{s.t.} \quad g_j(x) \leq 0 \quad j \in \{1, \ldots, |C|, \alpha_j = 1\} \\
\min_{s_1, \ldots, s_L \in \mathbb{R}, x \in \mathbb{W} \subset \mathbb{R}^n} & \quad \sum_j |s_j| \\
\text{s.t.} & \quad g_j(x) \leq s_j, \quad j \in \{1, \ldots, |C|, \alpha_j = 1\} \\
& \quad \beta \left( \sum_{k=1}^{i-1} |s_k| \right) \leq |s_i|, \quad j \in \{1, \ldots, |C|, \alpha_j = 1\}
\end{align*}
\]
**Summary of UNSAT certificates**

<table>
<thead>
<tr>
<th>UNSAT Certificate</th>
<th>Minimal</th>
<th>Complexity (number of convex problems)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial</td>
<td>No</td>
<td>Constant</td>
</tr>
<tr>
<td>IIS</td>
<td>Yes</td>
<td>Exponential</td>
</tr>
<tr>
<td>Sum of Slacks</td>
<td>Yes*</td>
<td>Linear</td>
</tr>
<tr>
<td>Minimum Prefix</td>
<td>Yes*</td>
<td>Constant</td>
</tr>
</tbody>
</table>

* under reasonable technical assumptions

\[
\left( (g_1(x) \leq 0) \land \phi_1(b) \right) \\
\lor \left( (g_2(x) \leq 0) \land \phi_2(b) \right) \\
\lor \ldots \\
\lor \left( (g_k(x) \leq 0) \land \phi_k(b) \right) \\
\text{Monotone SMC Formula}
\]

**Complexity** = \#Iterations \times \left( \text{Time}_{\phi(b)} + \text{Time}_{g(x) \leq 0} \right) \\
\text{“small” polynomial}
Scalability Results

Increase the number of Boolean constraints
#Boolean variables = 4800
#Real variables = 100

http://yshoukry.bitbucket.io/SatEX

Increase the number of Real variables
#Boolean variables = 4800
#Boolean constraints = 7000
Results(1): Single Robot, Reach-Avoid

Syclop (Synergistic Combination of Layers Of Planning):
- High level planner + low level RRT/EST
- outperform traditional sampling-based algorithms by orders of magnitude

Z3 and MILP (LTL OPT) times out
  time out =1 hour
Result (2): LTL Motion Planning

![Graph and Table]

\[ \phi ::= \Box \Diamond \pi_1 \land \Box \Diamond \pi_2 \land \Box \neg \pi_{obstacle} \]
Result (3): Multi-Robot, LTL

- \( \neg \text{obstacles} \)
- \( \bigwedge \bigcirc \bigcirc \text{at least one robot patrols the middle} \)
- \( \bigwedge \bigcirc \bigcirc \text{1st robot visits charging station } \#1 \)
- \( \bigwedge \bigcirc \bigcirc \text{2nd robot visits charging station } \#2 \)
- \( \bigwedge \bigcirc \text{1.5m separation} \)
- \( \bigwedge \bigcirc \text{velocity } \leq 2.5m/s \)
- \( \bigwedge \bigcirc \text{acceleration } \leq 1m/s^2 \)
Result (4): Secure CPS

Under attack - no protection

Under attack - with protection
Challenges

• Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control

• Implementation in software on top of networked robotics platforms

• Dealing with untrusted components, uncertainty in environment, and ML-based perception
Robotics Software Stack

Challenges

- High-Level Programming of Event-Driven Software
- Reliable Networked Behavior
- Verification in the Presence of Black/Gray-box Components
- Guaranteeing Safety via Run-time assurance
Drona: A Software Framework for Distributed Mobile Robotics

Contributions:
1. *Provably correct* decentralized motion planner that can perform on-the-fly collision-free planning *robust* against network clock-synchronization errors.
2. A *verified* software stack for distributed mobile robotics.
3. High-level ModP language, back-end model checker, and code generation to ROS/other SDKs.

Achievements:
- Programmed multiple drone platforms (Astec-Firefly, CrazyFlie, ROS, etc.).
- Framework efficiently scales for large workspaces with 128 robots.

Drona scalability experiment: 64 drones in simulation
Challenges

- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control

- Implementation in software on top of networked robotics platforms

- Dealing with untrusted components, uncertainty in environment, and ML-based perception
Simplex Architecture for Run-Time Assurance of Periodic Real-Time Systems

[Anderson et al., RTSS’98]

- Advanced Controller (AC)
- Safe Controller (SC)
- Decision Module (DM)
  (sampling period $\Delta$
- Plant or Robot

- RTA module to ensure property $\phi$

Already used in fault-tolerant avionics systems
Our Approach: Controlled Reachable Sets for Reach-Avoid Objectives

**Theorem:** The following is an inductive invariant:

\[
\text{Mode} = SC \land s \in \varphi_{safe} \\
\lor \\
\text{Mode} = AC \land \text{Reach}(s, *, \Delta) \in \varphi_{safe}
\]

Results: Drone Navigating in an Unknown Workspace (reach-avoid problems)
Summary

- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control → SMC

- Implementation in software on top of networked robotics platforms → Drona

- Dealing with untrusted components, uncertainty in environment, and ML-based perception → Run-time assurance + more... See tomorrow’s talk!