Safe & Interactive Autonomy:
A journey starting from Formal Methods

Dorsa Sadigh

iliad
intelligent and interactive autonomous systems
What does safety mean for Robotics?
Formal methods

Mathematically based technique for specification, development, and verification of software and hardware systems.

Verification: Timing and Energy analysis
Software: Embedded code

Timing Analysis of Interrupt-Driven Programs under Context Bounds
Jonathan Kotker, Dorsa Sadigh, Sanjit A. Seshia

Sanjit Seshia
Formal methods

**Mathematically based technique for specification, development, and verification of software and hardware systems.**

Verification (model checking):

- Specification
- Model
- Verifier
- Model satisfies specification?
  - Yes / No
What does safety mean for autonomous cars?

Formal methods (verification) + Control Theory / Robotics

Can we prove that autonomous cars are safe?

Sanjit Seshia

Shankar Sastry
Formal methods

Mathematically based technique for specification, development, and verification of software and hardware systems.

Verification (model checking):

Specification → Verifier → Yes / No

Hardware

Robot

Software

Orna Kupferman

Hadas Kress-Gazit

George Pappas

Summer of 2012
Formal methods

Mathematically based technique for specification, development, and verification of software and hardware systems.

Reactive Synthesis (Church 1957):

Specification → Verifier → Yes / No

Specification → Model → Verifier

Hardware
Software
Robot
Formal methods

Mathematically based technique for specification, development, and verification of software and hardware systems.

Reactive Synthesis (Church 1957):
What does safety mean for autonomous cars?

Formal methods + Control Theory / Robotics (synthesis)

Can we *synthesize* controller for autonomous cars are provably *safe*?
Controller Synthesis from Logic Specifications

Given a formal specification, encoding the
• objective,
• environment model,
synthesize a controller that is guaranteed to satisfy the specification.
Controller Synthesis from Logic Specifications

Temporal Logic Specifications:
- At all times (something is true)
- In the future (something is true)
- safety
- some version of stability
- surveillance (infinitely often)
- fairness
- response (if then else)
Controller Synthesis from Logic Specifications

Goal
Environment Model

Controller Synthesis

Temporal Logic Specifications
Controller Synthesis from Logic Specifications

Goal Environment Model → Controller Synthesis

Temporal Logic Specifications → Non-deterministic Buchi Automaton

ExpTime
Controller Synthesis from Logic Specifications

**Goal**

Environment Model → Controller Synthesis

Temporal Logic *Specifications* → Non-deterministic Buchi Automaton

Determinize it → ExpTime

ExpTime → Controller Synthesis
Controller Synthesis from Logic Specifications

Goal
Environment Model

Controller Synthesis

Temporal Logic Specifications

Determinize it

Non-deterministic Buchi Automaton

2-player game: system (what we control) & environment (all the thing that can go wrong)

Determinize it
Controller Synthesis from Logic Specifications

Goal Environment Model \rightarrow Controller Synthesis

Temporal Logic *Specifications* \rightarrow ExpTime

Determinize it \rightarrow ExpTime

Non-deterministic Buchi Automaton \rightarrow Polynomial Time

ExpTime

2-player game: *system* (what we control) \& *environment* (all the thing that can go wrong)
Controller Synthesis from Logic Specifications

Goal
Environment Model → Controller Synthesis

Temporal Logic Specifications

ExpTime

Determinize it

ExpTime

Non-deterministic Buchi Automaton

2-player game: system (what we control) & environment (all the thing that can go wrong)

Provably correct strategy

Counter-strategy

Polynomial Time
Given a formal specification, encoding the
- objective,
- environment model,
\textit{synthesize} a controller that is guaranteed to satisfy the specification.
Controller Synthesis from Logic Specifications

Given a formal specification, encoding the
• objective,
• environment model,
• human model
synthesize a controller that is guaranteed to satisfy the specification.
Temporal Logic Spec.

- Goal
- Environment Model
- Human Model

Controller Synthesis

Realizable

Unrealizable
Unprotected Left Turn

Temporal Logic Spec.

- Goal
- Environment Model
- Human Model

\[ \psi^{new} := (\varphi \land \psi^{env}) \rightarrow \psi^{sys} \]

Mine Assumptions

Counterstrategy Graph

Controller Synthesis

Realizable

Unrealizable

Assumption Monitoring

\[ auto = false \]
Controller Synthesis from Logic Specifications

Goal
Environment Model $\rightarrow$ Controller Synthesis

Temporal Logic Specifications:

1) Need discrete time and \textit{discrete space}.

"You should not discretize space if you want to talk about robots!"

"What about bisimulation?"

It does not scale!

Me

Ruzena Bajcsy

Year: 2015
Controller Synthesis from Logic Specifications

Goal
Environment Model → Controller Synthesis

Temporal Logic Specifications:
1) Need discrete time and discrete space.

Solution:
- Use Signal Temporal Logic Instead!
- Translate to Mixed-Integer Optimizations

“What are you doing??
This is just robust control!”

Claire Tomlin
Controller Synthesis from Logic Specifications

**Goal**
Environment Model \[\rightarrow\] Controller Synthesis

**Temporal Logic Specifications:**
1) Need discrete time and *discrete space*.
2) Our human models were totally made-up!

“Let’s study *safe and influencing interactions* in driving”

“Let’s model some humans!”

Me

Anca Dragan

[RSS’16, IROS’16, CPHS’18, AURO’18]
Learning Human's Preferences

Interaction-aware Control

Societal Impacts of Interactions
Learning Human’s Preferences

Interaction-aware Control
Learning Human’s Preferences

Interaction-aware Control
How to model *what humans want*

Write a reward function:

$$R_H(x, u_H)$$

$$u^*_R = \arg\max_{u_R} R_H(x, u_R)$$
How to model *what humans want*
How to model *what humans want*

Write a reward function:

$$R_H(x, u_H)$$

$$u_R^* = \arg\max_{u_R} R_H(x, u_R)$$
Task:

1. Reach the goal
2. Avoid the obstacle
3. Keep the arm low

\[ u^*_H = \max_{u_H} R_H(x, u_H) \]
Collect Expert Demonstrations
Inverse Reinforcement Learning

Learn Human’s reward function based on Inverse Reinforcement Learning:

\[ P(u_H | x, w) \propto \exp(R_H(x, u_H)) \]

\[ R_H(x, u_H) = w^\top \phi(x, u_H) \]

\[ u_H^* = \max_{u_H} R_H(x, u_H) \]

[Ziebart’ 08] [Levine’10]
Learned Policy from IRL
Providing Demonstrations is Difficult!

“\textit{I had a hard time controlling the robot}”

“\textit{I found the system difficult as someone who isn’t kinetically gifted}”
Leverage *comparisons* as useful observations about the desired robot reward function.
\[ P(\omega) \mid R(\xi_A) > R(\xi_B) \]
$P(\omega)$
Optimizing for **information gain**: (i) leads to *faster convergence*, and (ii) generates *easier questions.*
\[
\max I(\text{response}; \omega) = \max H(\text{response}) - H(\text{response}|\omega)
\]
Learning human preferences only from pairwise comparisons.
$w_1$ for Road Boundary
$w_2$ for Staying within Lanes
$w_3$ for Keeping Speed
$w_4$ for Heading
$w_5$ for Collision Avoidance

before

after
No prior preference

Learns *heading* preferences

Learns *collision avoidance* preferences
No prior preference

PREFERRING green basket over the red one.

Features: max altitude, final distance to the closest basket, max horizontal range, total angular displacement

[Bıyık, Sadigh. CoRL 2018]
Learning human preferences from pairwise comparisons + demonstrations.
Learning human preferences

*by asking easy questions.*
Asking Easy Questions: A User-Friendly Approach to Active Reward Learning. CoRL 2019
Accomplished
Did Wanted
Easy
Would Use Again

Rating
7
6
5
4
3
2
1
0

DemPref
IRL

I nteraction Rating Preference

* * *
“I had a hard time controlling the robot”

“I found the system difficult as someone who isn’t kinetically gifted”
Obtain offline demonstrations of high-dimensional motions
Latent Actions
Task 2: Adding Flour
Key Idea:

Learning *human models* is crucial for safe and robust control of robots.

We *actively* synthesize new comparison queries to quickly converge to human’s preference reward function.
Learning Human’s Preferences

Interaction-aware Control
An autonomous car’s actions will affect the actions of other drivers.
Interaction as a Dynamical System

direct control over $u_R$

indirect control over $u_H$
Interaction as a Dynamical System

$$u_R^* = \arg\max_{u_R} R_R(x, u_R, u_H^*(x, u_R))$$

Find optimal actions for the robot while accounting for the human response $u_H^*$. 

Interaction as a Dynamical System

\[ u^*_R = \arg\max_{u_R} R_R(x, u_R, u^*_H(x, u_R)) \]

\[ u^*_H(x, u_R) \approx \arg\max_{u_H} R_H(x, u_R, u_H) \]

Interaction as a Dynamical System

Find optimal actions for the robot while accounting for the human response $u^*_H$.

$$u^*_R = \arg\max_{u_R} R_R(x, u_R, u^*_H(x, u_R))$$

Model $u^*_H$ as optimizing the human reward function $R_H$.

$$u^*_H(x, u_R) \approx \arg\max_{u_H} R_H(x, u_R, u_H)$$

Implication: Efficiency
Implication: Efficiency
Implication: Efficiency
\[ u^*_\mathcal{H} = \arg\max_{u_\mathcal{H}} R_\mathcal{H}(x, u_\mathcal{R}, u^*_\mathcal{H}) \]

\[ u^*_\mathcal{R} = \arg\max_{u_\mathcal{R}} R_\mathcal{R}(x, u_\mathcal{R}, u^*_\mathcal{H}(x, u_\mathcal{R})) \]
\( R_{H}(x, u_H, u_R) \)
$R^\dagger_\mathcal{H}(x, u_\mathcal{H}, u_R)$

$|R^\dagger_\mathcal{H} - R_\mathcal{H}| < \delta$
How to algorithmically find these falsifying actions?
\[ \overline{u}_H = \arg \min_{u_H} R_R(x, u^*_R, u_H) \quad \text{Falsifying actions} \]

s.t. \( \exists R^+_H : u_H = \arg \max_{\overline{u}_H} R^+_H(x, u^*_R, \overline{u}_H) \)

\[ |R^+_H - R_H| < \delta \quad \text{Optimizing a perturbed version of the learned reward function.} \]
Theorem:

\[ \bar{u}_H = \arg \min_{u_H} R_R(x, u_R^*, u_H) \]

s.t. \( \exists R^+_H : u_H = \arg \max_{\bar{u}_H} R^+_H(x, u_R^*, \bar{u}_H) \)

\[ \bar{u}_H = \arg \min_{u_H} R_R(x, u_R^*, u_H) \]

s.t. \( R_H(x, u_R^*, u_H) > R_H(x, u_R^*, u_H^*) - 2\delta \)

D. Sadigh, S. Sastry, S. Seshia. HCPS 2018.
\( \overline{u}_H = \arg\min_{u_H} R_R(x, u_R^*, u_H) \)

s.t. \( R_H(x, u_R^*, u_H) > R_H(x, u_R^*, u_H^*) - 2\delta \)
\[ |R^+_H - R_H| \leq \delta \]
\[ \delta = 0 \]
\[ |R^\dagger_H - R_H| < \delta \]
\[ \delta = 0.025 \]
\[ |R_{\mathcal{H}}^\dagger - R_{\mathcal{H}}| < \delta \]

\[ \delta = 0.15 \]
\[ \Pr\left( |R_{\mathcal{H}}^+ - R_{\mathcal{H}}| < \delta \right) > 0.9 \]

\[ \delta = 0.15 \]
Key Idea:

Robot’s actions *affect* human’s actions. We want to *leverage* these effects for better safety and efficiency.

We need to *be robust to inaccuracies* of human models.
Reactive Synthesis View: Prove safety under the assumption that everyone is adversarial.
Robust Control View: Prove safety under some model of disturbance.

Safe Learning View: Accepting you don’t know things! Safely explore the things you don’t know!
`Does it matter that you don't know things?`

**Reactive Synthesis View:** Prove safety under the assumption that everyone is adversarial.

**Robust Control View:** Prove safety under some model of disturbance

**Safe Learning View:** Accepting you don’t know things! Safely explore the things you don’t know!

**Modular View:** Does it matter that you don’t know things?
Opportunities...

• More *scalable* techniques: better hardware, better software (compute), better algorithms (data), and *domain specific knowledge*.
• More *modular* systems, more *explainability*
• Better models of *humans and interactions*.
• Getting used to failing and learning from them.
Evolution of Accident Rates for Each Generation Aircraft

10 year moving average accident rate per million flights*

*Below 10 years of operation, the moving average is based on the number of years of operation.
# Fatality Rates of Autonomous Cars

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Manufacturer</th>
<th>Fatality</th>
<th>Autonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Jan 2016</td>
<td>Hebei, China</td>
<td>Tesla, S</td>
<td>Driver Fatality</td>
<td>Level 2</td>
</tr>
<tr>
<td>7 May 2016</td>
<td>Florida</td>
<td>Tesla, S</td>
<td>Driver Fatality</td>
<td>Level 2</td>
</tr>
<tr>
<td>23 March 2018</td>
<td>California</td>
<td>Tesla, X</td>
<td>Driver Fatality</td>
<td>Level 2</td>
</tr>
<tr>
<td>1 March 2019</td>
<td>Florida</td>
<td>Tesla, 3</td>
<td>Driver Fatality</td>
<td>Level 2</td>
</tr>
<tr>
<td>25 April 2019</td>
<td>Florida</td>
<td>Tesla, S</td>
<td>Pedestrian Fatality</td>
<td>Level 2</td>
</tr>
<tr>
<td>18 March 2018</td>
<td>Arizona</td>
<td>Uber</td>
<td>Pedestrian Fatality</td>
<td>Level 3</td>
</tr>
</tbody>
</table>
Sometimes we got to fail, multiple times, and that is okay!

Alison Gopnik’s work on how adult and baby rats learn

How to be aware of them. How to diagnose and repair. How to learn from them.
Learning Human’s Preferences

Interaction-aware Control

Societal Impacts of Interactions