Adaptable yet Provably Correct

Autonomous Systems

Ufuk Topcu
University of Texas

u-t-autonomous.info

Partly supported by NSF, Air Force, ONR, ARO and DARPA.
Integration of Unmanned+Autonomous Systems

Limited to simple tasks

- Low-level, pre-programmed capabilities for fixed conditions

Limited gains in manpower efficiency

- Humans are “deep” in the loop
  - Blind separation → limited acceptance, automation surprises

Hard to establish trust

- Safety critical...
  - ... yet impossible to exhaustively test

High and unsustainably increasing costs

- Affordability in time, manpower, monetary costs is not clear (yet)
A Synthesis Problem
[adapted to an autonomous navigation scenario]

Given:

• **System model**
  - both continuous & discrete evolution
  - actuation limitations
  - modeling uncertainties & disturbances

• **Specifications**
  - high-level requirements & goals
  - assumptions about the a priori unknown, dynamic environment
    
    \[(\varphi_{\text{init}} \land \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \land \varphi_{\text{goal}})\]

Automatically synthesize a control protocol that
• manages the system behavior;
• reacts to changes in allowable external environment; and
• is provably correct with respect to the specifications.
“Specify+Synthesize+Implement” Demonstrations

Quadrotor testbed
(with Kumar Lab at Penn)

Abstract Models + High-Level, LTL Specifications

Correct-By-Construction Synthesis

Executable Code Generation

Hardware Implementation (or high-fidelity simulation)

Unified locomotion and manipulation
(with Ye Zhao at Texas)

Electric power network testbed
Joint **Control + Learning** in Operator-Autonomy Interaction

(with proofs w.r.t. “rich specifications”)

Conventional Synthesis

- Environment: $\phi_{env}$
- Requirements: $\phi_{req}$
- Performance: $C$

Known/fixed temporal logic constraints + reward structure

Synthesis + Learning

1. No a priori knowledge about environment. But, online (and offline) data available.
2. Operators do not “speak” temporal logic or cannot express complex reward structures.
3. Controller itself may be learned from data or examples.
Adaptable yet Provably Correct?

Adaptable but no clear semantics for “correctness”

Unambiguous proofs but fragile

PROBABLY APPROXIMATELY CORRECT
Nature’s Algorithms for Learning and Prospering in a Complex World

THE TEMPORAL LOGIC OF PROGRAMS*
Amir Pnueli
University of Pennsylvania, Pa. 19104
and
Tel-Aviv University, Tel Aviv, Israel

Turing Award: 1996

Turing Award: 2010
Joint Control + Learning in Operator-Autonomy Interaction
(with proofs w.r.t. “rich specifications”)

1+2 Integrate temporal logic constraints into learning
(with Min Wen and Jie Fu)

- Probably approximately correct learning and control with temporal logic specifications
  - Markov decision processes (RSS 2014)
  - Stochastic, two-player, quantitative games (IJCAI 2016)

2 Constrain learning to provably correct operation envelopes
(with Min Wen, Ruediger Ehlers, Nils Jansen, Sebastian Junges and Joost-Pieter Katoen)

- Correct-by-synthesis reinforcement learning with temporal logic constraints
  - Two-player games (IROS 2015)
  - Markov decision processes (TACAS 2016)

3 Automaton learning to solve safety games over infinite graphs
(with Daniel Neider)
Problem Statement

**Given:** MDP \( M = \langle Q, \Sigma, q_0, P, AP, L \rangle \) with unknown transition function (its structure is known)

LTL specification \( \varphi \)

**Synthesize a policy** that is...

**Correct in temporal logic sense:**
A policy \( f: Q^* \longrightarrow \Sigma \) induces a Markov chain \( M^f \) and a probability distribution over the infinite runs \( \rho \) in \( Q^\omega \).

\[
\max_f \sum_{\rho \models \varphi} \Pr(\rho, M^f)
\]

**Efficient in PAC (probably approximately correct) sense:**
With high confidence,

- learn
- a near-optimal policy

with "polynomial" time and sample complexity

**Related earlier work:**
- Statistically-oriented
  - infinite amount of data
- Restricted class of spec’s and models
  (Legay, Henriques, Chen, Mao,…)

**OR**
- Temporal logic → history dependence
- Markovian-reward assumptions violated
  (Kearns, Brafman…)}
Some Definitions and (Standard) Facts

Given: partially known MDP $M = \langle V, \Sigma, v_0, P, \mathcal{AP}, L \rangle$ + specification $\varphi$

- Express $\varphi$ as a deterministic Rabin automaton $A_{\varphi}$
- Account for memory by passing to the product automaton $\mathcal{M} = M \times A_{\varphi}$
- $C$: union of accepting end components in $\mathcal{M}$

Probability of satisfying $\varphi = \text{probability of reaching } C$

visiting certain sets infinitely often and some sets only finitely many times

After $q$, always eventually $p$
$q$, $p$: atomic propositions

"states visited infinitely often"
Some Definitions and (Standard) Facts

Given: partially known MDP $M = \langle V, \Sigma, v_0, P, AP, L \rangle$ + specification $\varphi$

$\mathcal{M} = M \times A_\varphi$

$C$: union of accepting end components in $\mathcal{M}$

At state $v$ of $\mathcal{M}$ under policy $f$: $U^f_{\mathcal{M}}(v, T) : T$-step state value (probability of reaching $C$ in $T$ steps)

$U^f_{\mathcal{M}}(v) :$ state value (probability of reaching $C$)

$\varepsilon$-state-value mixing time under policy $f$: $T = \min \{ t : \max_{v \in V} |U^f_{\mathcal{M}}(v, t) - U^f_{\mathcal{M}}(v)| \leq \varepsilon \}$
A Solution

Use standard synthesis tools for temporal logic constrained MDPs

synthesize a policy based on the currently estimated MDP

$A_\varphi$ → strategy $f$. → unknown MDP $M$. → learning → data

update maximum likelihood estimate of $P(v, \sigma, v')$

implement the strategy until “sufficient” knowledge has been accumulated

$\overline{P}(q, \sigma, q') = \frac{\theta_t^{q_0, \sigma}(q')}{\|\theta_t^{q_0, \sigma}\|_1}$

count of transitions from $q$ to $q'$ under $\sigma$

count of all transitions from $q$ under $\sigma$

Under standard assumptions, the estimate is a normal random variable with mean
A useful property

**Def:** $\overline{M} \alpha$-approximates $M$ if $\left| P(v, \sigma, v') - P(v, \sigma, v') \right| \leq \alpha$
for every $(v, \sigma, v')$.

**Lemma:** If $\overline{M} \frac{\epsilon}{N \cdot T}$-approximates $M$, then for any (synthesized) policy $f$ with $\epsilon$-state value mixing time $T$, it holds that

$$\left| U^f_M(v, T) - U^f_{\overline{M}}(v, T) \right| \leq \epsilon$$

for all states $v$ in $M$.

Bound the gap between the $T$-step state values in the estimated MDP and in the true MDP.
Main Results

A probabilistic transition \((v, \sigma, v')\) is known, if we have \(\text{Var} \cdot k \leq \epsilon / NT\) for any \(v' \in V\) with probability at least \(1 - \delta\)

Var: variance of the maximum likelihood estimate
\(k\): the critical value for the \(1 - \delta\) confidence interval

**Theorem**: Optimal policy \(f\) in the “known part” of \(\overline{M}\) either

- **exploits** — near-optimal in the value of \(U^f_M(v, T)\), or
- **explores** — runs into a *new unknown* state with non-zero probability.

**Diagram**:
- Strategy \(f\)
- Unknown MDP \(M\)
- Synthesis
- Learning
- Estimated MDP \(\overline{M}\)
- Data
Theorem: Given $\epsilon$ and $\delta$ in $\{0,1\}$. Let $T$ be the $\epsilon$-state-value mixing time of the optimal policy.

With probability $1-\delta$,

the resulting policy $f$ satisfies

$$\left| \Pr(v, M^f \models \varphi) - \max_g \Pr(v, M^g \models \varphi) \right| \leq 2\epsilon \quad \forall v$$

with a number of steps polynomial in $|M|, |A_\varphi|, T, 1/\delta, 1/\epsilon$. 
A Case Study

Motion planning over a grid world
Infinitely often visiting green blocks $R_1$, $R_2$, and $R_3$ in sequence while avoiding all red cells.

Safety?
Probability of safety violation minimized at convergence.
Learning in two-player, stochastic, Buchi games—
problem statement

Given

• a turn-based stochastic Buchi game $G = (S, S_S, S_E, I, A, T, R, F)$ satisfying
  - the game is fully observable for both players;
  - the system knows the correct list of all possible successors for all state-action pairs, but does not know the exact transition distributions a priori;
  - the reward function is unknown a priori, but is upper bounded by the specified positive number $R_{\text{max}}$;
• a discount factor $\gamma \in (0; 1)$; and
• a sub-optimality bound $\epsilon > 0$.

Learn

• a memoryless system strategy $\sigma_{s, \epsilon}$ that satisfies
  - the almost-sure winning objective and
  - the discounted-sum objective

\[ V_{\sigma_{s, \epsilon}}(s) \geq \max_{\sigma' \in \Sigma_S} V_{\sigma'}(s) - \epsilon \]

V$_{\sigma}(s)$: worst-case, expected, $\gamma$-discounted reward from state $s$ under the system strategy $\sigma$. 

a system strategy $\sigma$ is almost-sure winning at $s$, if a run beginning at $s$ visits $F$ infinitely often with probability one regardless of the environment strategy.
Learning in two-player, stochastic, Buchi games — algorithm

1: Compute the almost-sure winning region $W_{as}^\text{in}$ and a memoryless almost-sure winning strategy $\sigma_s$ for the system in $G^\text{in}$.
2: $G := G^\text{in} \upharpoonright W_{as}^\text{in} = (S, S_a, S_e, I, A, T, R, F)$.
3: Initialize the game model $\hat{G} := (S, S_a, S_e, I, A, T, R, F)$ such that $\hat{G}$ shares the same set of transitions with $G$ and all transition distributions in $\hat{T}$ are uniform; $\hat{R}(s, a, s') \sim R_{\max}$ for all transition $(s, a, s')$.
4: $\sigma_{s,e} \leftarrow \sigma_s$.
5: For all existing transition $(s, a, s')$ in $\hat{G}$, $k(s, a, s') \leftarrow 0$, $L(s, a) \leftarrow 0$, $\hat{Q}^*(s, a) \leftarrow \frac{R_{\max}}{1 - \gamma}$ for all transition $(s, a, s')$.
6: $\delta \leftarrow \frac{\varepsilon(1 - \gamma)^2 \log(\gamma)}{6 R_{\max} |S| \log(\varepsilon(1 - \gamma)^2/6 R_{\max})}$, $K \leftarrow \frac{1}{2\delta^2} \log \frac{4|A||S|^2}{\delta^2}$.
7: $\varepsilon_1 \leftarrow \frac{\varepsilon}{12}, P_{\varepsilon_1} \leftarrow \frac{\varepsilon(1 - \gamma)^2}{12 R_{\max} - \varepsilon(1 - \gamma)^3}$.
8: while TRUE do
  9:    if $s \in S_a$ then
  10:      Take $\sigma_{s,e}$ for one step.
  11:    else
  12:      Environment takes a transition.
  13:    end if
  14:    Observe the transition $(s, a, s')$ and the reward $r$.
  15:    if $L(s, a) = 0$ then
  16:      $k(s, a, s') \leftarrow k(s, a, s') + 1, R(s, a, s') \leftarrow r$.
  17:    if $\sum_{s' \in S} k(s, a, s') \geq K$ or $|E(\hat{G}(s, a))| = 1$ then
  18:      $L(s, a) \leftarrow 1$.
  19:    For all $s' \in S, T(s, a)(s') = \sum_{s' \in S} k(s, a, s') / k(s, a, s')$.
  20:    Update the optimal action value function $\hat{Q}^*$.
  21:    Construct $\hat{G} \leftarrow \text{HatGame}(G, \hat{Q}^*, \varepsilon_1, P_{\varepsilon_1})$.
  22:    Compute a memoryless almost-sure winning strategy $\hat{\sigma}_s$ for the system in $\hat{G}$.
  23:    $\sigma_{s,e} \leftarrow \text{RecoverHatStrategy}(\hat{G}, \hat{\sigma}_s)$.

Restrict the game to the systems winning region and compute a memoryless, almost-sure winning strategy — sufficient to know the structure.

Apply steps (updates) similar to those in the MDP case (generalization to so-called R-max algorithm).

Construct a new game $G$-hat from $G$ such that...

... for every almost-sure winning strategy in $G$-hat, we can construct an almost-sure winning strategy in $G$; and

... every memoryless system strategy in $G$-hat can be used to construct a memoryless, $\varepsilon$-optimal strategy in $G$.

Repeat as new knowledge is discovered.
Learning in two-player, stochastic, Buchi games — main result

Given

• a turn-based stochastic Buchi game $G = (S, S_s, S_e, I, A, T, R, F)$ satisfying
• a discount factor $\gamma \in (0; 1)$;
• a sub-optimality bound $\varepsilon > 0$; and
• a confidence lower bound $\delta$ such that $1-\delta \in (0,1)$.

Then

... with probability no less than $1-\delta$,
... the system strategy $\sigma_{s,\varepsilon}$ in the algorithm (previous slide)
... is memoryless, almost-sure winning and behaves $\varepsilon$-optimally
... except for a number of steps that is polynomial in $|S|, |A|, 1/\varepsilon$, and $1/\delta$.

Open (and interesting): Can we do with Rabin games?

• Rabin games + optimality require finite-memory strategies.
• The proof techniques on the learning side “seem to break with memory”.
Known turn-based game + unknown reward structure

Uncontrolled

Environment

Controlled

System

Temporal logic specification

Environment assumption, $\varphi_{env}$

System requirement, $\varphi_{sys}$

$$\begin{align*}
\theta_{i_{\text{init}}}^{e} \land \bigwedge_{i \in I_{e}} \square \psi_{i}^{e} \land \bigwedge_{k \in K_{e}} \Diamond J_{k}^{e} & \rightarrow \theta_{i_{\text{init}}}^{s} \land \bigwedge_{i \in I_{s}} \square \psi_{i}^{s} \land \bigwedge_{k \in K_{s}} \Diamond J_{k}^{s}
\end{align*}$$

Human operator

Performance criterion

Reward function for the system: $S \times A \rightarrow \mathbb{R}$.

- Human preferences vary over operators
- Hard to model a priori

Compute a finite-memory, optimal, winning strategy for the system
(optimal w.r.t. unknown reward → learning)
The problem is harder than its pieces

Compute a finite-memory, optimal, winning strategy for the system

- Realizability of LTL specifications
- Turn-based deterministic games
- Optimal winning strategies may need infinite memory
- Deterministic memoryless strategy
- Deterministic memoryless strategy
- Discounted reward

A game with no environment states

Initial state: $s_0$

$\varphi = \Diamond b_2$

$R(s_0, a_0) = 1$, $R(s_0, a_1) = R(s_1, a_2) = 0$
Solution overview

- **Permissive strategy:**
  - non-deterministic
  - is winning for the system,
  - “includes” all memoryless winning system strategies
  (inclusion in terms of the sets of runs generated by the strategies)

- All allowable system executions in the new game satisfy the specifications.

- The choice of reinforcement learning algorithm is independent of the specification.

- Learning/optimality is decoupled from correctness.
Known turn-based game + unknown reward structure (main result)

The algorithm converges to a (possibly suboptimal) solution to the problem.

- **Permissive** strategies exist for all finite-state games (if the spec is realizable).

If a **maximally** permissive strategy exists, the algorithm converges to an optimal solution to the problem.

A **maximally** permissive strategy includes all winning system strategies.
A note on the existence and computation of (maximally) permissive strategies

[Ehlers & Finkbeiner, 2011] Characterize the family of specifications for which maximally permissive strategies exists (as reactive safety) and provide a method for constructing.

[Bernet et al., 2002] For safety formulas, there always exists a maximally permissive strategy.

When the LTL formula is of the form $\varphi_0 \land \Box \varphi_1$ (with $\varphi_0$ and $\varphi_1$ atomic propositions — one $\bigcirc$ allowed), a maximally permissive strategy can be computed in linear time.

**Toolbox:** [Ehlers] SLUGS.
Example

 NxN grid world with one controlled and one environment robot.

 Assumptions and guarantees to describe allowable system and environment moves.

 Unknown reward is to encourage the system robot to reach positions diagonal to the environment robot’s position as often as possible.

 **Safety**: Always avoid collision.
 - Maximally permissive strategy exists.
 - Correct + optimal strategy computed.

 Add **liveness**: Both players to visit certain cells infinitely often.
 ⇒ Results may be suboptimal.

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>8.1</td>
</tr>
<tr>
<td>7.29</td>
</tr>
</tbody>
</table>

 $\delta_i \in \hat{S}_g$

 Iteration (x 10^4)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>9.5</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>8.5</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

 Sub-optimality decreases monotonically with increasing memory (or permissiveness).

 (N=4)

 (N=3)

 a measure of memory
More generally: known model + unknown reward structure

- **design time**
  - a priori known models & specifications
  - strategy synthesis
    - two-player (deterministic) game + temporal logic
    - Markov decision process + safety

- **run time**
  - permissive strategy
  - learning
    - data
    - minimax Q-learning
    - Q-learning

optimal + correct decisions
Summary

Integrate temporal logic constraints into learning
(with Min Wen and Jie Fu)

Probably approximately correct learning and control with
 temporal logic specifications
  • Markov decision processes (RSS 2014)
  • Stochastic, two-player, quantitative games (IJCAI 2016)

Constrain learning to provably correct operation envelopes
(with Min Wen, Ruediger Ehlers, Nils Jansen, Sebastian Junges and Joost-Pieter Katoen)

Correct-by-synthesis reinforcement learning with temporal logic constraints
  • Two-player games (IROS 2015)
  • Markov decision processes (TACAS 2016)

Automaton learning to solve safety games over infinite graphs
(with Daniel Neider)
References

http://arxiv.org/abs/1404.7073

http://arxiv.org/abs/1503.01793

http://arxiv.org/abs/1510.05880

http://arxiv.org/abs/1601.01660