Safe Reinforcement Learning via Formal Methods

Nathan Fulton and André Platzer

Carnegie Mellon University
"How can we provide people with cyber-physical systems they can bet their lives on?" - Jeannette Wing
Autonomous Safety-Critical Systems

How can we provide people with autonomous cyber-physical systems they can bet their lives on?
Model-Based Verification  Reinforcement Learning
pos < stopSign
Model-Based Verification

Reinforcement Learning

pos < stopSign
**Approach**: prove that control software achieves a specification with respect to a model of the physical system.
**Approach**: prove that control software achieves a specification with respect to a model of the physical system.

Model-Based Verification  
Reinforcement Learning
Model-Based Verification

Reinforcement Learning

Benefits:

● Strong safety guarantees
● Automated analysis
Model-Based Verification

Benefits:

- Strong safety guarantees
- Automated analysis

Drawbacks:

- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
Benefits:
- Strong safety guarantees
- Automated analysis

Drawbacks:
- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
- Assumes accurate model
Benefits:

- Strong safety guarantees
- Automated analysis

Drawbacks:

- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
- Assumes accurate model.
Model-Based Verification

**Benefits:**
- Strong safety guarantees
- Automated analysis

**Drawbacks:**
- Control policies are typically non-deterministic: answers "what is safe", not "what is useful"
- Assumes accurate model.

Reinforcement Learning

**Benefits:**
- No need for complete model
- Optimal (effective) policies
Model-Based Verification

Benefits:
- Strong safety guarantees
- Automated analysis

Drawbacks:
- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
- Assumes accurate model.

Reinforcement Learning

Benefits:
- No need for complete model
- Optimal (effective) policies

Drawbacks:
- No strong safety guarantees
- Proofs are obtained and checked by hand
- Formal proofs = decades-long proof development
Model-Based Verification

Benefits:
- Strong safety guarantees
- Assumptions on model

Drawbacks:
- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
- Assumes accurate model

Reinforcement Learning

Benefits:
- No need for complete model
- Optimal (effective) policies

Drawbacks:
- No strong safety guarantees
- Proofs are obtained and checked by hand
- Formal proofs = decades-long proof development

Goal: Provably correct reinforcement learning
Model-Based Verification

Benefits:
- Strong safety guarantees
- Computational aids (ATP)

Drawbacks:
- Control policies are typically non-deterministic: answers “what is safe”, not “what is useful”
- Assumes accurate model

Reinforcement Learning

Benefits:
- No need for complete model
- Optimal (effective) policies

Drawbacks:
- No strong safety guarantees
- Proofs are obtained and checked by hand
- Formal proofs = decades-long proof development

Goal: Provably correct reinforcement learning

1. Learn Safety
2. Learn a Safe Policy
3. Justify claims of safety
Model-Based Verification

Accurate, analyzable models often exist!

\{
  \{?safeAccel; accel \cup brake \cup ?safeTurn; turn\};
  \{pos' = vel, vel' = acc\}
}\*
Model-Based Verification

**Accurate**, analyzable models often exist!

\[
\{ \\
\{\text{?safeAccel};\text{accel} \cup \text{brake} \cup \text{?safeTurn};\text{turn}\}; \\
\{\text{pos'} = \text{vel}, \text{vel'} = \text{acc}\}
\}^* \text{ Continuous motion}
\] 

\text{discrete control}
Model-Based Verification

Accurate, analyzable models often exist!

\[
\{ \text{pos'} = \text{vel, vel'} = \text{acc} \}^{\ast}
\]

\[
\{ \text{?safeAccel;} \text{accel} \cup \text{brake} \cup \text{?safeTurn;} \text{turn} \}^{\ast}
\]

Continuous motion  \quad \text{discrete, non-deterministic control}
Model-Based Verification

**Accurate, analyzable** models often exist!

\[ \text{init} \rightarrow \{
\begin{align*}
\{ & \text{?safeAccel;} \text{accel} \cup \text{brake} \cup \text{?safeTurn;} \text{turn}\}; \\
\{ & \text{pos}' = \text{vel}, \text{vel}' = \text{acc} \}
\}
\]* \text{pos} < \text{stopSign} \]
Model-Based Verification

**Accurate, analyzable** models often exist!

formal verification gives strong safety guarantees

\[
\text{init} \rightarrow \left\{ \left\{ \begin{array}{l}
\text{?safeAccel; accel } \cup \text{ brake } \cup \text{?safeTurn; turn} \\
\text{pos'} = \text{vel}, \text{vel'} = \text{acc}
\end{array} \right\} \right\} \text{pos < stopSign}
\]
Model-Based Verification

Accurate, analyzable models often exist!
formal verification gives strong safety guarantees

VERIFIED

= • Computer-checked proofs of safety specification.
Model-Based Verification

**Accurate, analyzable** models often exist!

formal verification gives strong safety guarantees

=  
  - Computer-checked proofs of safety specification
  - Formal proofs mapping model to runtime monitors
Model-Based Verification Isn’t Enough

**Perfect**, analyzable models don’t exist!
Model-Based Verification Isn’t Enough

Perfect, analyzable models don’t exist!

How to implement?

\[
\begin{align*}
\{ & ?\text{safeAccel};\text{accel} \cup \text{brake} \cup ?\text{safeTurn}; \text{turn} \\
& \{ \text{pos'} = \text{vel}, \text{vel'} = \text{acc} \} \}
\end{align*}
\]

Only accurate sometimes
Model-Based Verification Isn’t Enough

Perfect, analyzable models don’t exist!

How to implement?

\[
\{ 
\{ \text{?safeAccel;} \text{accel} \cup \text{brake} \cup \text{?safeTurn;} \text{turn}\}; \\
\{dx'=w*y, dy'=-w*x, \ldots\}
\}^* \\
\text{Only accurate sometimes}
\]
Our Contribution

Justified Speculative Control is an approach toward provably safe reinforcement learning that:
1. learns to resolve non-determinism without sacrificing formal safety results
Our Contribution

Justified Speculative Control is an approach toward provably safe reinforcement learning that:

1. learns to resolve non-determinism without sacrificing formal safety results
2. allows and directs speculation whenever model mismatches occur
Learning to Resolve Non-determinism

Act

Observe & compute reward
Learning to Resolve Non-determinism

accel U brake U turn

Observe & compute reward
Learning to Resolve Non-determinism

Observe & compute reward

\{accel, brake, turn\}
Learning to Resolve Non-determinism

{accel, brake, turn}

Observe & compute reward

Policy
Learning to Resolve Non-determinism

{accel, brake, turn}

Observe & compute reward

(safe?) Policy
Learning to **Safely** Resolve Non-determinism

- Safety Monitor
- Observe & compute reward
- (safe?) Policy
Learning to **Safely** Resolve Non-determinism

- Safety Monitor
- Observe & compute reward

≠ “Trust Me”
Learning to **Safely** Resolve Non-determinism

Use a theorem prover to prove:

\[(\text{init} \rightarrow \left[\{\text{accel} \cup \text{brake}\};\text{ODEs}\right]^{*}(\text{safe})) \leftrightarrow \varphi\]
Learning to **Safely** Resolve Non-determinism

Use a theorem prover to prove:

\[
\text{(init} \rightarrow [[[\text{accel} \cup \text{brake}];\text{ODEs}]^*](\text{safe})) \iff \phi
\]
Learning to **Safely** Resolve Non-determinism

**Main Theorem:** If the ODEs are accurate, then our formal proofs transfer from the non-deterministic model to the learned (deterministic) policy

Use a theorem prover to prove:

\[
\text{(init} \rightarrow [\{{\text{accel}} \cup \text{brake}\};\text{ODEs}*](\text{safe})) \leftrightarrow \varphi
\]
Learning to **Safely** Resolve Non-determinism

**Main Theorem:** If the ODEs are accurate, then our formal proofs transfer from the non-deterministic model to the learned (deterministic) policy via the model monitor.

Use a theorem prover to prove:

\[
(init \rightarrow [\{\text{accel} \cup \text{brake}\}; \text{ODEs}]^*)(\text{safe}) \leftrightarrow \varphi
\]
What about the physical model?

Use a theorem prover to prove: \((\text{init} \implies (\text{safe})) \iff \varphi\)
What About the Physical Model?

Observe & compute reward

{brake, accel, turn}
What About the Physical Model?

Model is accurate.

{brake, accel, turn}

Observe & compute reward
What About the Physical Model?

Model is accurate.

{brake, accel, turn}

Observe & compute reward
What About the Physical Model?

Observe & compute reward

{brake, accel, turn}

Model is accurate.

Model is inaccurate
What About the Physical Model?

Model is accurate.

Model is inaccurate

Obstacle!
What About the Physical Model?

{brake, accel, turn}

Observe & compute reward

Expected Reality
Speculation is Justified

Observe & compute reward

\{brake, accel, turn\}

Expected (safe)

Reality (crash!)
Leveraging Verification Results to Learn Better

{brake, accel, turn}

Observe & compute reward

Use a real-valued version of the model monitor as a reward signal
Conclusion

Justified Speculative Control provides the best of logic and learning:
Conclusion

Justified Speculative Control provides the best of logic and learning:

- Formally model the control system (**control + physics**)

![Diagram](image-url)
Conclusion

**Justified Speculative Control** provides the best of logic and learning:

- Formally model the control system (**control + physics**)
- Learn how to resolve non-determinism in models.
Conclusion

**Justified Speculative Control** provides the best of logic and learning:

- Formally model the control system (**control + physics**)
- Learn how to resolve non-determinism in models.
- Leverage theorem proving to transfer **proofs** to learned policies.
Conclusion

Justified Speculative Control provides the best of logic and learning:

- Formally model the control system (**control + physics**)
- Learn how to resolve non-determinism in models.
- Leverage theorem proving to transfer **proofs** to learned policies.
- Unsafe **speculation is justified** when model deviates from reality
Conclusion

Justified Speculative Control provides the best of logic and learning:

- Formally model the control system (control + physics)
- Learn how to resolve non-determinism in models
- Leverage theorem proving to transfer proofs to learned policies
- Unsafe speculation is justified when model deviates from reality, but verification results can still be helpful!
Conclusion

Justified Speculative Control provides the best of logic and learning:

- Formally model the control system (control + physics)
- Learn how to resolve non-determinism in models
- Leverage theorem proving to transfer proofs to learned policies
- Unsafe speculation is justified when model deviates from reality, but verification results can still be helpful!