Safe Reinforcement Learning via Formal Methods

Nathan Fulton and André Platzer

Carnegie Mellon University
Safety-Critical Systems

"How can we provide people with cyber-physical systems they can bet their lives on?" - Jeannette Wing
How can we provide people with autonomous cyber-physical systems they can bet their lives on?
Model-Based Verification  Reinforcement Learning

\( \varphi \)
pos < stopSign
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

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
- Aomputational 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
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

Model-Based Verification

Goal: Provably correct reinforcement learning
1. Learn Safely
2. Learn a Safe Policy
3. Justify claims of safety

Reinforcement Learning

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

Accurate, analyzable models often exist!

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

**Accurate**, analyzable models often exist!

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

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

\[\ast\] Continuous motion

Discrete control
Model-Based Verification

**Accurate**, analyzable models often exist!

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

Continuous motion

discrete, *non-deterministic* control
Model-Based Verification

Accurate, analyzable models often exist!

\[
\text{init} \rightarrow \{ \{
\{ \text{?safeAccel}; \text{accel} \cup \text{brake} \cup \text{?safeTurn}; \text{turn} \};
\{ \text{pos'} = \text{vel}, \text{vel'} = \text{acc}, \text{t'}=1 \}\}^* \}\text{pos < stopSign}
\]
Model-Based Verification

**Accurate, analyzable** models often exist!

Formal verification gives strong safety guarantees

\[ \text{init} \rightarrow \{ \{ \text{?safeAccel, accel} \cup \text{brake} \cup \text{?safeTurn, turn} \}; \{ \text{pos}' = \text{vel}, \text{vel}' = \text{acc}, t'=1 \} \}^* \text{pos} < \text{stopSign} \]
Model-Based Verification

Accurate, analyzable models often exist!

formal verification gives strong safety guarantees

\[
= \quad \bullet \text{Computer-checked proofs of safety specification.}
\]
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?

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

\{ \text{pos’} = \text{vel}, \text{vel’} = \text{acc} \}

\}^* 

Only accurate sometimes
Model-Based Verification Isn’t Enough

**Perfect**, analyzable models don’t exist!

How to implement?

\[
\{ \)
\{ ?safeAccel; accel \cup brake \cup ?safeTurn; turn \};
\{ dx’=w*y, dy’=-w*x, \ldots \}
\}^* \\

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 $\cup$ brake $\cup$ turn

Observe & compute reward
Learning to Resolve Non-determinism

\{\text{accel}, \text{brake}, \text{turn}\}

Observe & compute reward
Learning to Resolve Non-determinism

\{\text{accel}, \text{brake}, \text{turn}\}

Observe & compute reward

\rightarrow

Policy
Learning to Resolve Non-determinism

{accel, brake, turn}

Observe & compute reward

⇒

(safe?)

Policy
Learning to **Safely** Resolve Non-determinism

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

Observed & compute reward

Safety Monitor

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

Use a theorem prover to prove:

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

Use a theorem prover to prove:

\[(\text{init} \rightarrow [[[\text{accel} \cup \text{brake}]; \text{ODEs}]*](\text{safe})) \leftrightarrow \phi\]
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:

\[(\text{init} \rightarrow [(\text{accel} \cup \text{brake}) ; \text{ODEs}]^*)(\text{safe})] \iff \phi\]
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 via the model monitor.

Use a theorem prover to prove:

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

Use a theorem prover to prove: \((\text{init} \rightarrow (\text{init} \rightarrow ([\{\text{accel} \cup \text{brake}\}; \text{ODEs}]^* (\text{safe})))) \leftrightarrow \varphi\)
What About the Physical Model?

{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?

{brake, accel, turn}

Model is accurate.

Observe & compute reward
What About the Physical Model?

Model is accurate.

Model is inaccurate.

{brake, accel, turn}

Observe & compute reward
What About the Physical Model?

Observe & compute reward

{brake, accel, turn}

Model is accurate.

Model is inaccurate

Obstacle!
What About the Physical Model?

Observe & compute reward

{brake, accel, turn}

Expected

Reality
Speculation is Justified

{brake, accel, turn}

Observe & compute reward

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
An Example
An Example: The System

\[\text{init} \rightarrow \{\{\text{?safeAccel;accel} \cup \text{brake} \cup \text{?safeMaint; maintVel}\}; \{\text{pos'} = \text{vel}, \text{vel'} = \text{acc}, \text{t'}=1\}\}^*\text{safe}\]
An Example: The Monitor

init → [{

{?safeAccel; accel ∪ brake ∪ ?safeMaintain; maintainVel};

{pos' = vel, vel' = acc, t' = 1}

}]*safe

(t_{post} >= 0 ∧ a_{post} = acc ∧ v_{post} = acc t_{post} + v ∧ p_{post} = acc t_{post}^2/2 + v t_{post} + p) ∨
(t_{post} >= 0 ∧ a_{post} = 0 ∧ v_{post} = v ∧ p_{post} = vt_{post} + p) ∨ Etc.
An Example: The Monitor

init → [{

{safeAccel; accel ∪ brake ∪ safeMaintain; maintainVel};

{pos' = vel, vel' = acc, t' = 1}

}]*safe

(t_{post} >= 0 ∧ a_{post} = acc ∧ v_{post} = acc t_{post} + v ∧ p_{post} = acc t_{post}^2/2 + v t_{post} + p) ∨

(t_{post} >= 0 ∧ a_{post} = 0 ∧ v_{post} = v ∧ p_{post} = vt_{post} + p) ∨ Etc.
An Example: The Monitor

\[\text{init} \rightarrow [\{\]

\[?\text{safeAccel};\text{accel} \cup \text{brake} \cup ?\text{safeMaintain};\text{maintainVel}\};\]

\[\{\text{pos'} = \text{vel}, \text{vel'} = \text{acc}, t' = 1\}\]

\[\text{]*safe}\]

\[(t_{\text{post}} \geq 0 \land a_{\text{post}} = \text{accel} \land v_{\text{post}} = \text{acc} t_{\text{post}} + v \land p_{\text{post}} = \text{acc} t_{\text{post}}^2/2 + v t_{\text{post}} + p) \lor \]

\[(t_{\text{post}} \geq 0 \land a_{\text{post}} = 0 \land v_{\text{post}} = v \land p_{\text{post}} = vt_{\text{post}} + p) \lor \text{Etc.}\]
An Example: The Monitor

init → [{

{?safeAccel;accel U brake U ?safeMaintain; maintainVel};

{pos' = vel, vel' = acc, t'=1}

}]safe

(t_{post} >= 0 ∧ a_{post} = acc ∧ v_{post} = accel t_{post} + v ∧ p_{post} = acc t_{post}^2/2 + v t_{post} + p) ∨

(t_{post} >= 0 ∧ a_{post} = 0 ∧ v_{post} = v ∧ p_{post} = vt_{post} + p) ∨ Etc.
An Example: The Monitor

\[ \text{init} \rightarrow \{ \]

\{ ?\text{safeAccel};\text{accel} \cup \text{brake} \cup ?\text{safeMaintain};\text{maintainVel} \}; \]

\{ \text{pos}' = \text{vel}, \text{vel}' = \text{acc}, t' = 1 \} \]

\} \text{safe} \]

\( (t_{\text{post}} \geq 0 \land a_{\text{post}} = \text{acc} \land v_{\text{post}} = \text{accel} t_{\text{post}} + v \land p_{\text{post}} = \text{acc} t_{\text{post}}^2/2 + v t_{\text{post}} + p) \lor \]

\( (t_{\text{post}} \geq 0 \land a_{\text{post}} = 0 \land v_{\text{post}} = v \land p_{\text{post}} = vt_{\text{post}} + p) \lor \text{Etc.} \]
An Example: The Monitor

init → [{

{?safeAccel; accel ∪ brake ∪ ?safeMaintain; maintainVel};

{pos' = vel, vel' = acc, t'=1}

}]*safe

(t_{post} >= 0 ∧ a_{post} = acc ∧ v_{post} = accel ∧ p_{post} = acc t_{post}^2/2 + v t_{post} + p) ∨

(t_{post} >= 0 ∧ a_{post} = 0 ∧ v_{post} = v ∧ p_{post} = v t_{post} + p) ∨ Etc.
An Example: The Reward Signal (simplified)

\[x \geq 0 \land v \geq 0 \land A \geq 0 \rightarrow \{x' = v, v' = A\}]x \geq 0\]
An Example: The Reward Signal (simplified)

\[ x \geq 0 \land v \geq 0 \land A \geq 0 \rightarrow \left\{ x' = v, \ v' = A \right\} x \geq 0 \]

Minimize \( \max(v_{\text{Error}}, x_{\text{Error}}) \) where

\[
v_{\text{Error}} = \max(v_{\text{post}} - (A t_{\text{post}} + v), A t_{\text{post}} + v - v_{\text{post}})\]

\[
x_{\text{Error}} = \max(x_{\text{post}} - (A t_{\text{post}}^2/2 + v t_{\text{post}} + x), (A t_{\text{post}}^2/2 + v t_{\text{post}} + x) - x_{\text{post}})\]
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*)
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!

Policy $\phi$