Statistical Verification of Learning-Based Cyber-Physical Systems

Miroslav Pajic Yu Wang*

Department of Electrical and Computer Engineering
Department of Civil and Environment Engineering
Department of Computer Science
Department of Mechanical Engineering and Material Science
Duke University

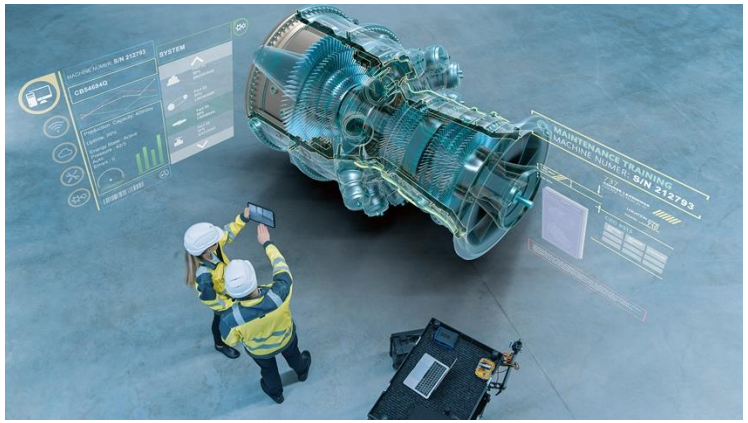
Duke
PRATT SCHOOL of ENGINEERING

UF Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

Motivation

Duke
PRATT SCHOOL of ENGINEERING

Advances in computation (particularly AI) has significantly improved our ability to handle physical systems.

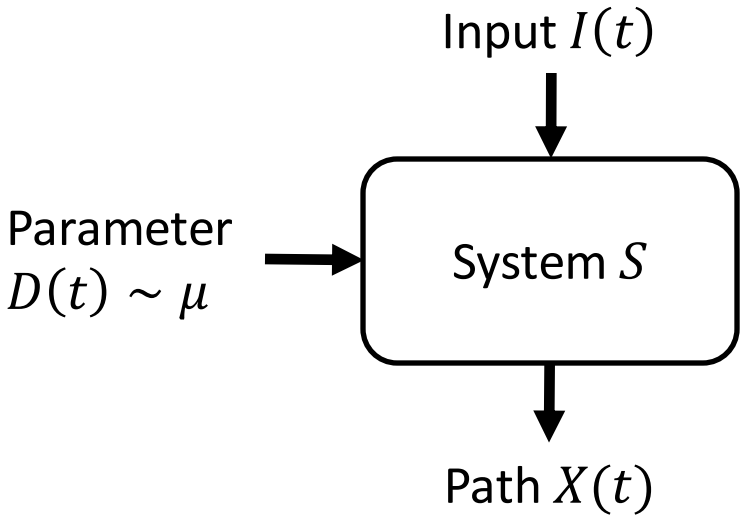


How to assure the functionality of these systems?

CPS Verification

Duke
PRATT SCHOOL of ENGINEERING

Cyber-physical systems are complex and full of unknowns for model-based analysis.



Question: If for any allowed input $I(t)$,

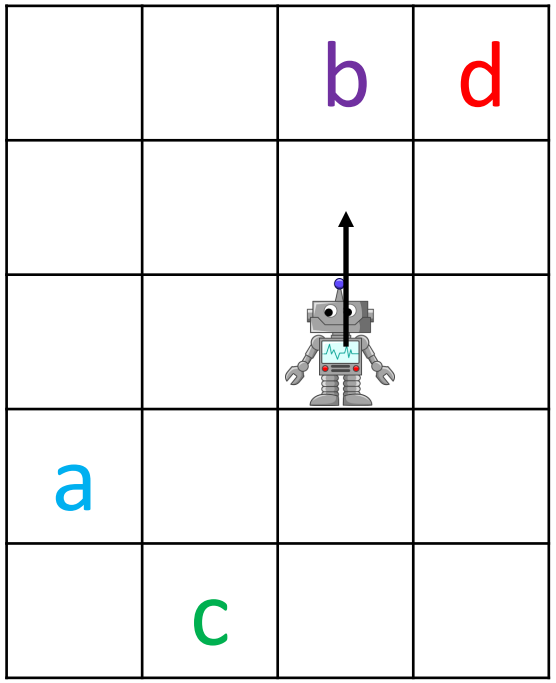
$\Pr_{D(t)\sim\mu}(S \text{ satisfies the design specification}) > 0.99$

How to develop (***statistical***) verification algorithms for these systems?

What is missing?

Duke
PRATT SCHOOL of ENGINEERING

There are many different “design specifications”.



- a: adult
- b: baby
- c: charger
- d: danger

Specifications:

(1) Avoid danger,

(2) No return to baby before visiting adult or charger,

(3) Go for baby after leaving adult,

(4) Go for charger after baby asleep,

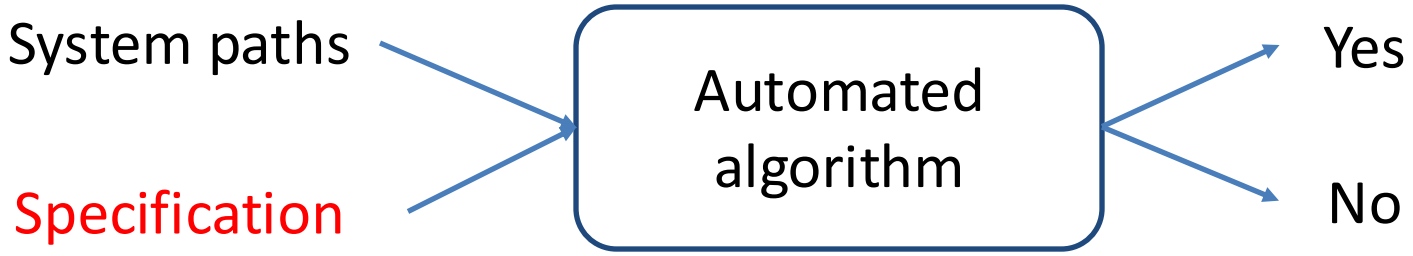
(5) After leaving charger, return to baby, and

(6) Go for adult after baby awake.

The testing/verification algorithm for one specification is not directly usable for another.

Our Goal

Develop an automated algorithm that can deal with a class of specifications.

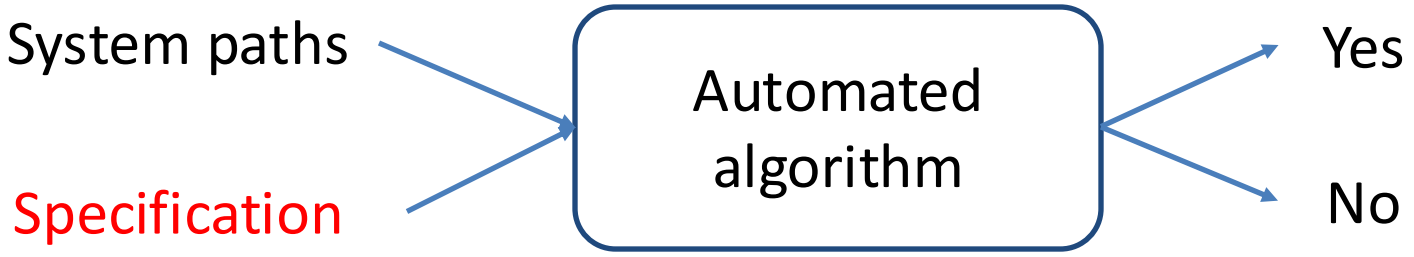


How to express specification in an understandable way to computer algorithms?

A Key Ingredient: Formal Methods

Duke
PRATT SCHOOL of ENGINEERING

Express the specification in a symbolic language to allow automated parsing, analysis, and reasoning.



Formal methods: Use mathematical logic (i.e., automated theorem proving) to system analysis.

For cyber-physical systems, we use temporal logic.

Syntax of Temporal Logic

Duke
PRATT SCHOOL of ENGINEERING

ALWAYS (1) Avoid danger,
(2) NO return to baby BEFORE visiting adult OR charger,
(3) Go for baby AFTER leaving adult,
(4) Go for charger AFTER baby asleep,
(5) AFTER leaving charger, return to baby, AND
(6) Go for adult AFTER baby awake.

All **propositional** connectives are expressible by NO ($\neg$), AND ($\wedge$).
For example, a OR b $= \neg(\neg a \wedge \neg b)$.

All **temporal** connectives are expressible by UNTIL ($\mathbf{U}$), NEXT ($\bigcirc$).
For example,
ALWAYS a $= \neg(\text{True}\ \mathbf{U}\ \neg a)$.

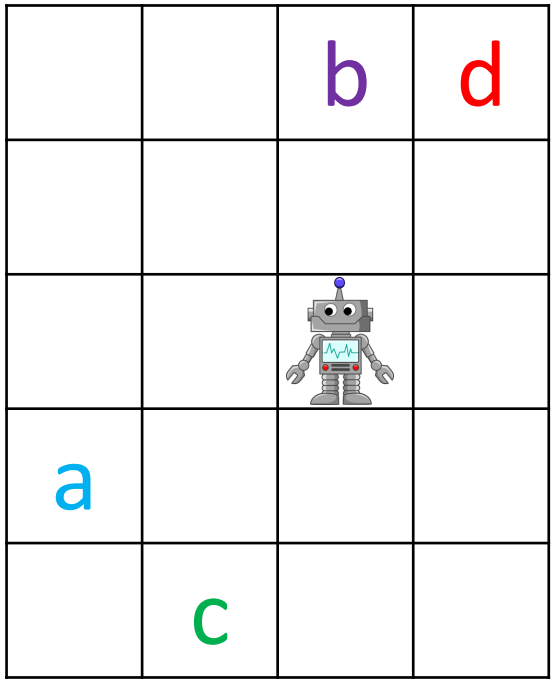
The planning tasks are formally expressible by **Linear Temporal Logic (LTL)**:

$$\varphi := a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

where a is any Boolean variable (called atomic proposition).

Temporal Logic in Action

Duke
PRATT SCHOOL of ENGINEERING



- a: adult
- b: baby
- c: charger
- d: danger

Example Control Task of a Nursing Robot:
Always (1) Avoid danger,
(2) No return to baby before visiting adult or charger,
(3) Go for baby after leaving adult,
(4) Go for charger after baby asleep,
(5) After leaving charger, return to baby, and
(6) Go for adult after baby awake.

$$\varphi = \square\Big(\underbrace{\neg d}_{(1)} \wedge \underbrace{(b \wedge \neg \bigcirc b) \rightarrow \bigcirc(\neg b \,\mathsf{U}\, (a \vee c))}_{(2)} \wedge \underbrace{a \rightarrow \bigcirc(\neg a \,\mathsf{U}\, b)}_{(3)}$$
$$\wedge \underbrace{(\neg b \wedge \bigcirc b \wedge \neg \bigcirc \bigcirc b) \rightarrow (\neg a \,\mathsf{U}\, c)}_{(4)} \wedge \underbrace{c \rightarrow (\neg a \,\mathsf{U}\, b)}_{(5)} \wedge \underbrace{(b \wedge \bigcirc b) \rightarrow \Diamond a}_{(6)} \Big)$$

□: Always ◇: Finally

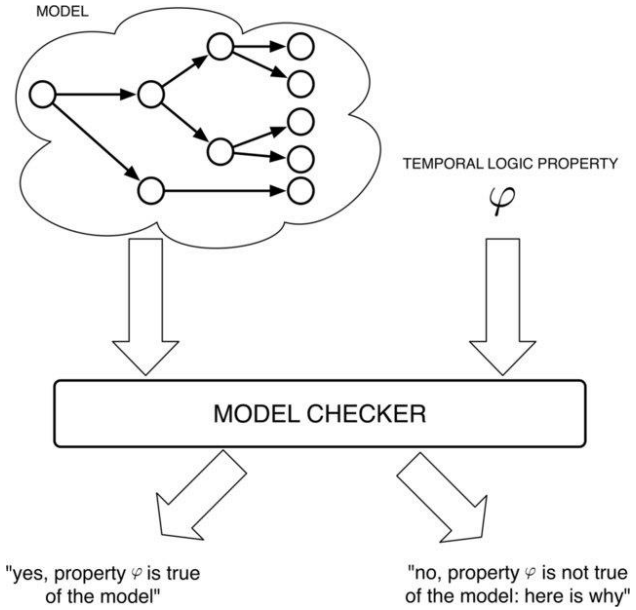
Review of Formal Methods

Duke
PRATT SCHOOL of ENGINEERING

Formal methods (particularly model checking) have been successful in ***white-box*** verification of ***simple*** systems.

Model Checking

[CAV 19] Elfar et al., Computer-Aided Verification, 2019
[ATVA20] Elfar et al., Int. Sym. on Automated Technology for Verification and Analysis, 2020
[CDC 19a] Wang et al., IEEE Conf. on Decision and Control, 2019
[CDC 19b] Wang et al., IEEE Conf. on Decision and Control, 2019



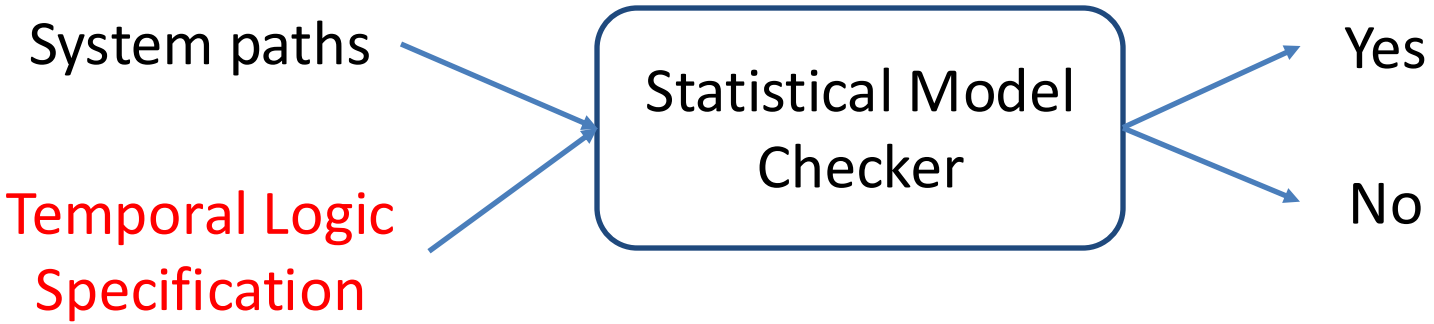
However, they are not applicable and scalable to ***complex black-box*** systems

Our Approach: Statistical Model Checking

Duke
PRATT SCHOOL of ENGINEERING

Question: Check if for ***any*** allowed input $I(t)$,

$$\Pr(S \text{ satisfies } \varphi) > c$$



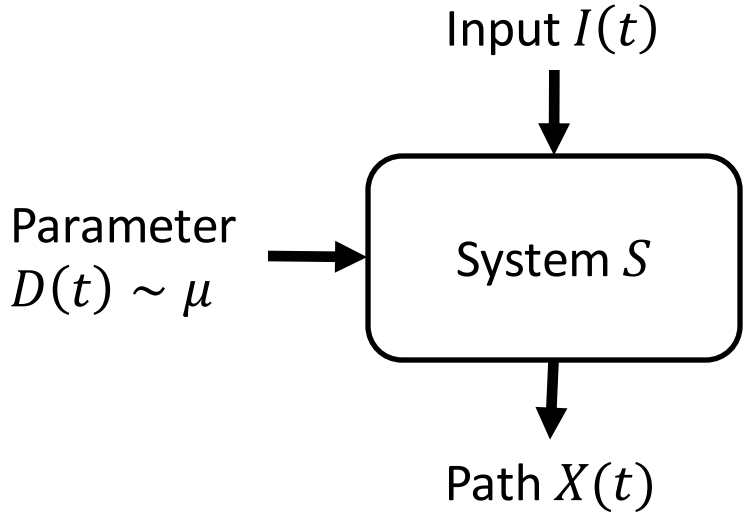
For any given $\alpha > 0$, it should return an answer with ***significance level α***, i.e.,

$$\Pr(\text{Answer is Yes} \mid \text{Specification is False}) < \alpha$$

$$\Pr(\text{Answer is No} \mid \text{Specification is True}) < \alpha$$

A Simple Case

Consider a fixed input $I(t)$.



- For LTL specification φ, S satisfies φ ***if and only if*** $X(t)$ satisfies φ.
- $P_{D(t)\sim\mu}\,(S \text{ satisfies } \varphi)$ can be estimated by sampling $X(t)$.

Statistical model checking turns into a hypothesis testing problem for the **satisfaction probability** $p = \Pr_{D(t)\sim\mu}(X_D(t) \text{ satisfies } \varphi)$:

$$H_0: p_\varphi > c, H_1: p_\varphi \leq c$$

Old Method

Duke
PRATT SCHOOL of ENGINEERING

Assume an **indifferent region** $p_\varphi \notin (c-\delta, c+\delta)$ for some known $\delta > 0$.

$$H_0: p_\varphi > c$$
$$H_1: p_\varphi \leq c$$

$$H_0: p_\varphi = c + \delta$$
$$H_1: p_\varphi = c - \delta$$

The composite hypothesis testing turns into the simple hypothesis testing, so it can be solved by standard sequential probability ratio test.

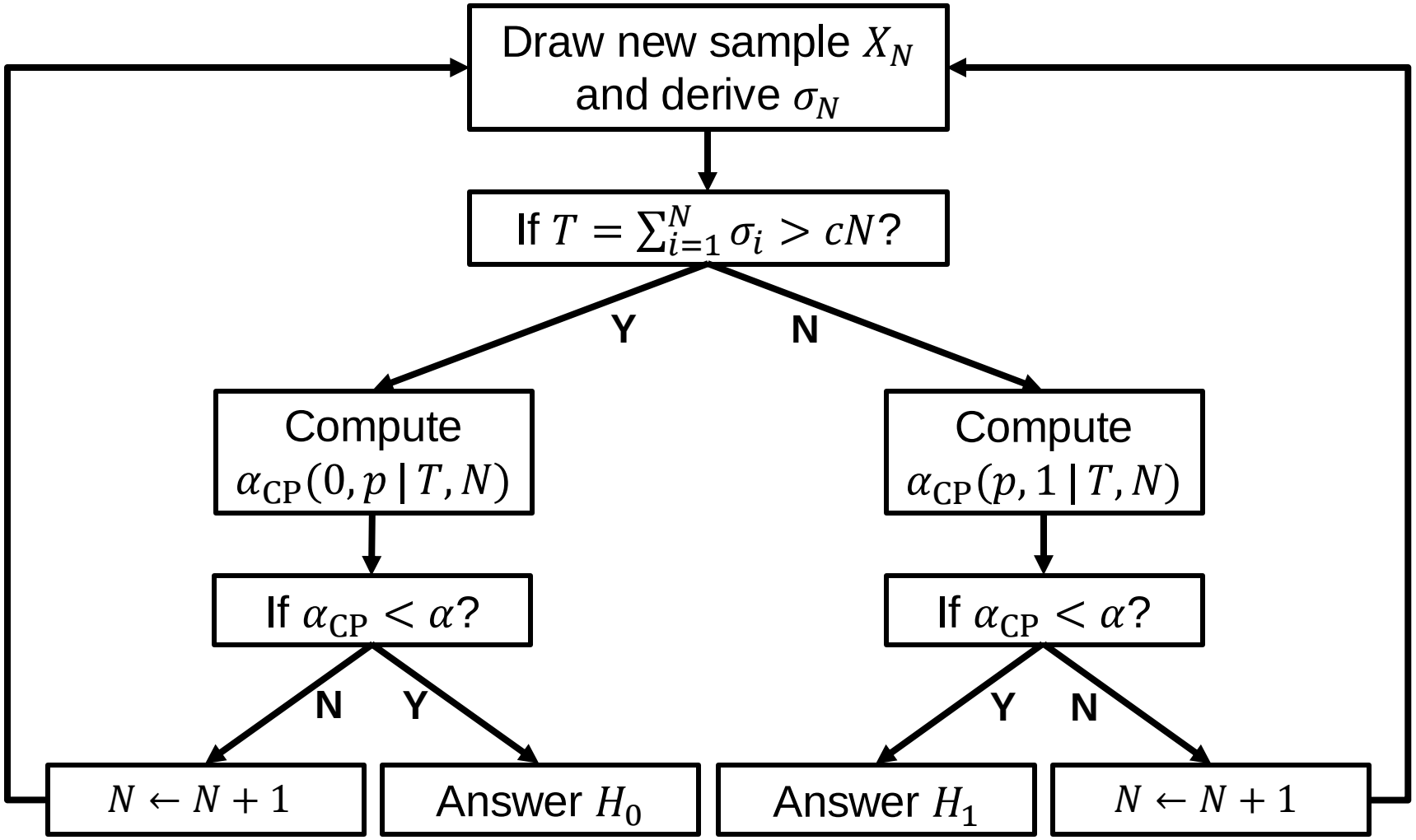
Our Method

Duke
PRATT SCHOOL of ENGINEERING

- For a sample path X_i, the event $\sigma_i = \begin{cases} 1, & \text{if } X_i \text{ satisfies } \varphi, \\ 0, & \text{otherwise.} \end{cases}$ is binary.
- The sum $\sum_{i=1}^{N} \sigma_i$ obeys the binomial distribution.
- Suppose we answer H_0 if $\frac{1}{N}\sum_{i=1}^{N} \sigma_i > c$ and H_1 if $\frac{1}{N}\sum_{i=1}^{N} \sigma_i \leq c$.
- The significant levels $\Pr\left(\frac{1}{N}\sum_{i=1}^{N} \sigma_i > c \mid p_\varphi \leq c\right)$ and $\Pr\left(\frac{1}{N}\sum_{i=1}^{N} \sigma_i \leq c \mid p_\varphi > c\right)$ can be computed by the Clopper-Pearson Exact Method.
- For a target significant level α,

  we just keep on sampling until the significance levels are below α.

Our Algorithm

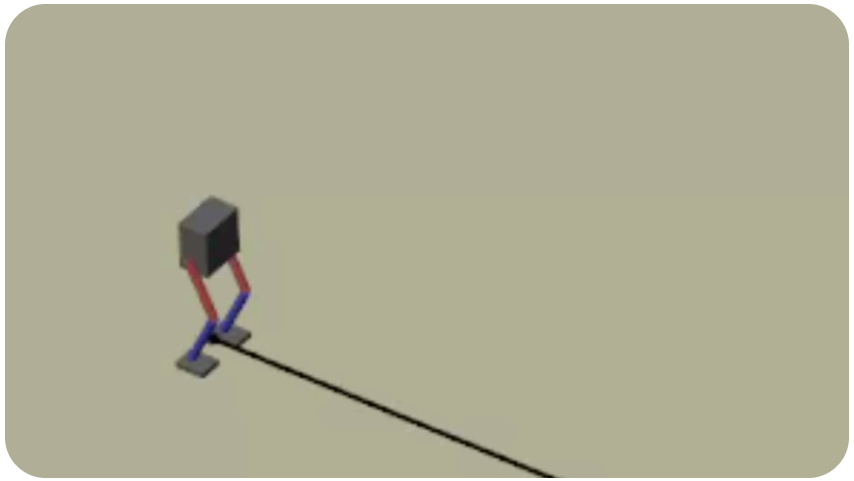


$$\alpha_{\mathrm{CP}}(a, b \mid T, N) = 1 - \begin{cases} (1-a)^N - (1-b)^N & \text{if } T = 0 \\ b^N - a^N & \text{if } T = N \\ F_{\mathrm{Beta}}(b \mid T+1, N-T) - F_{\mathrm{Beta}}(a \mid T, N-T+1) & \text{otherwise.} \end{cases}$$

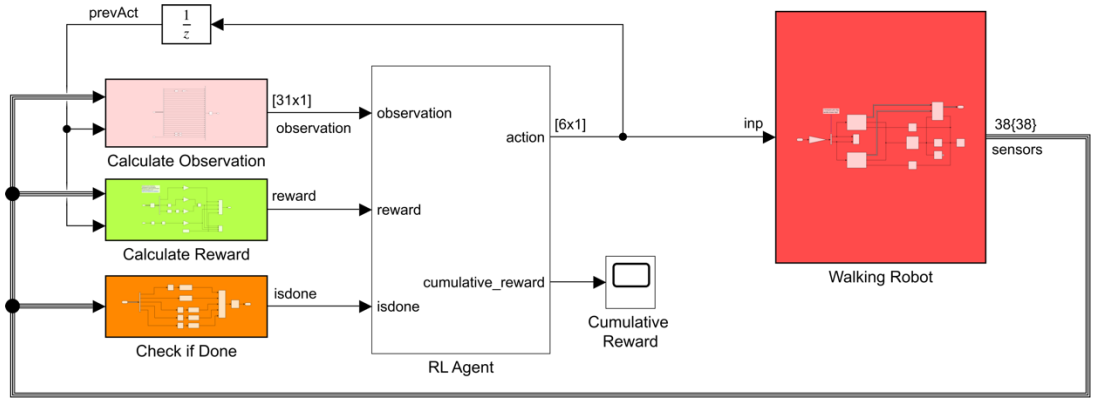
Main statistical results:

- The algorithm terminate with probability 1 if $p_\varphi \neq c$.
- The final answer has significance level α.

Application: Bipedal Robot



Simulink model of the Bipedal Robot [HSCC'20]



$$\Pr\left(\Diamond_{[0,5]}(x > \delta) \wedge \Box_{[0,5]}(\neg\eta_1 \wedge \neg\eta_2 \wedge \neg\eta_3)\right) > 1 - \varepsilon.$$

- $\boldsymbol{\eta_1}$: The lateral movement is greater than a threshold ($|y| > 0.5$),
- $\boldsymbol{\eta_2}$: The robot has fallen ($z < 0.1$),
- $\boldsymbol{\eta_3}$: The robot's roll, pitch, or yaw are greater than a threshold ($|\phi| > \frac{\pi}{4}$, $|\theta| > \frac{\pi}{4}$, $|\psi| > \frac{\pi}{4}$).
- $\varepsilon \in \{0.02, 0.12, 0.2\}$, $\delta \in \{3.0, 2.4\}$, $\alpha \in \{0.01, 0.05\}$

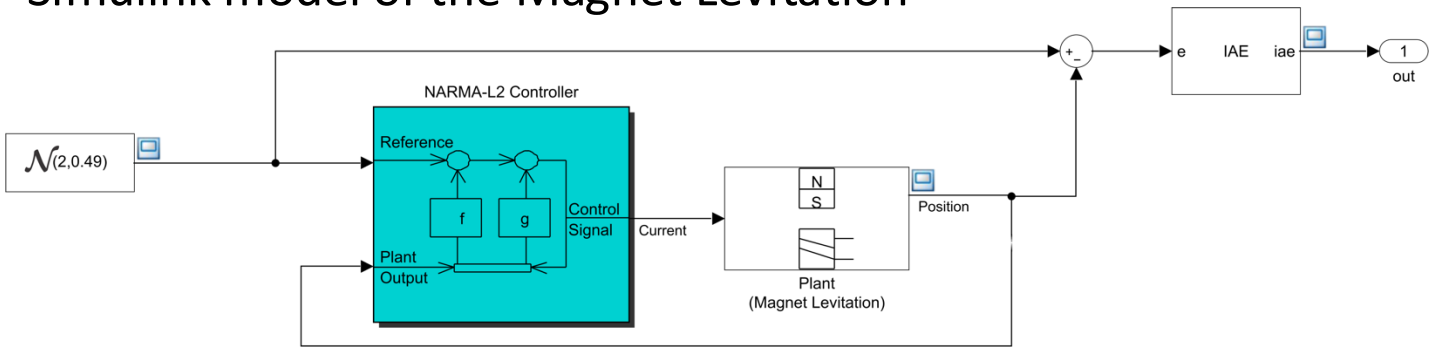
δ	$1-\varepsilon$	α	Acc.	Sam.	Time (s)	Ans.
2.4	0.02	0.01	1.00	7.4e+01	3.0e-01	False
2.4	0.02	0.05	0.99	4.4e+01	1.4e-01	False
2.4	0.12	0.01	1.00	4.2e+01	1.2e-01	True
2.4	0.12	0.05	1.00	2.1e+01	7.0e-02	True
2.4	0.20	0.01	1.00	1.3e+01	4.0e-02	True
2.4	0.20	0.05	1.00	6.7e+00	1.4e-02	True
3.0	0.02	0.01	1.00	1.1e+01	2.4e-02	False
3.0	0.02	0.05	1.00	6.5e+00	1.1e-02	False
3.0	0.12	0.01	1.00	1.4e+02	4.3e-01	False
3.0	0.12	0.05	0.98	7.0e+01	2.3e-01	False
3.0	0.20	0.01	1.00	1.6e+02	5.5e-01	True
3.0	0.20	0.05	0.98	1.0e+02	2.9e-01	True

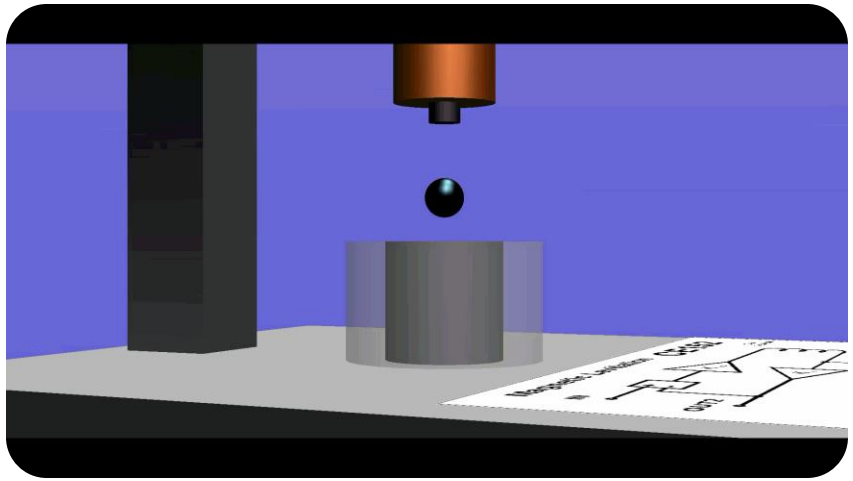
Application: Magnet Levitation

Duke
PRATT SCHOOL of ENGINEERING

[HSCC'20]

Simulink model of the Magnet Levitation





Metrics of interest: *Integral Absolute Error (IAE)*

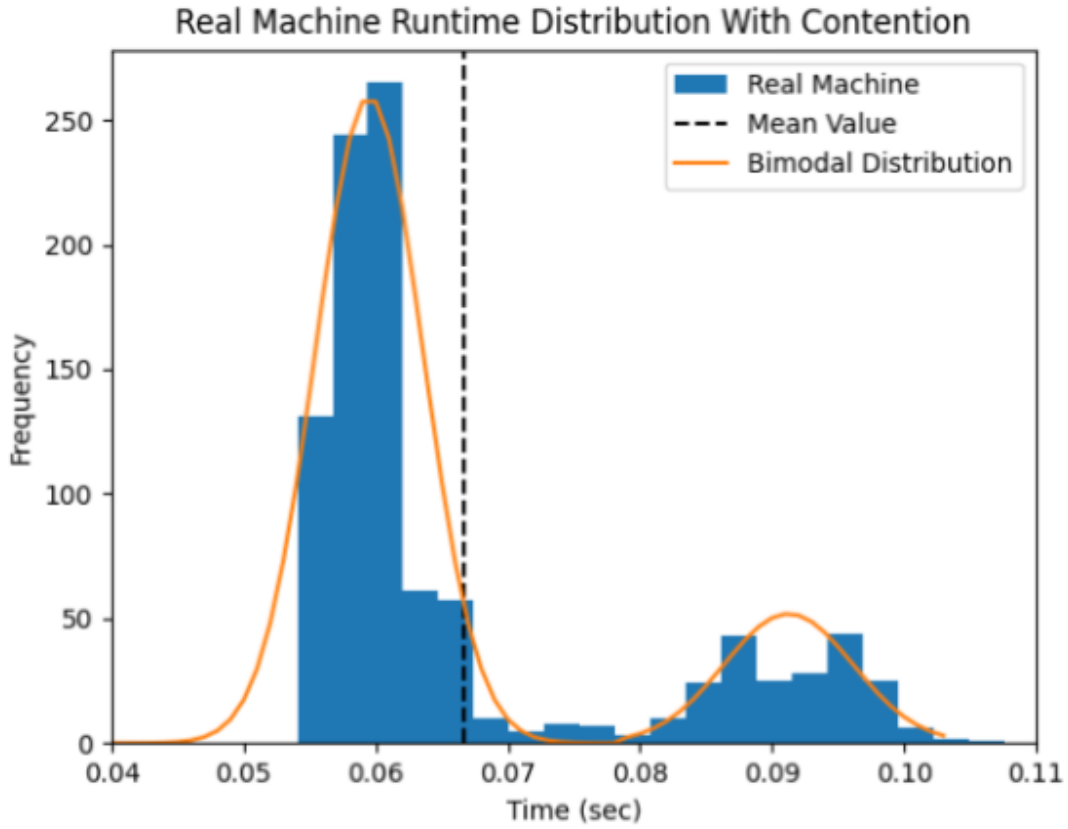
$$\mathbf{Pr}(\square_{[0,5]}(IAE < \delta)) > 1 - \varepsilon$$

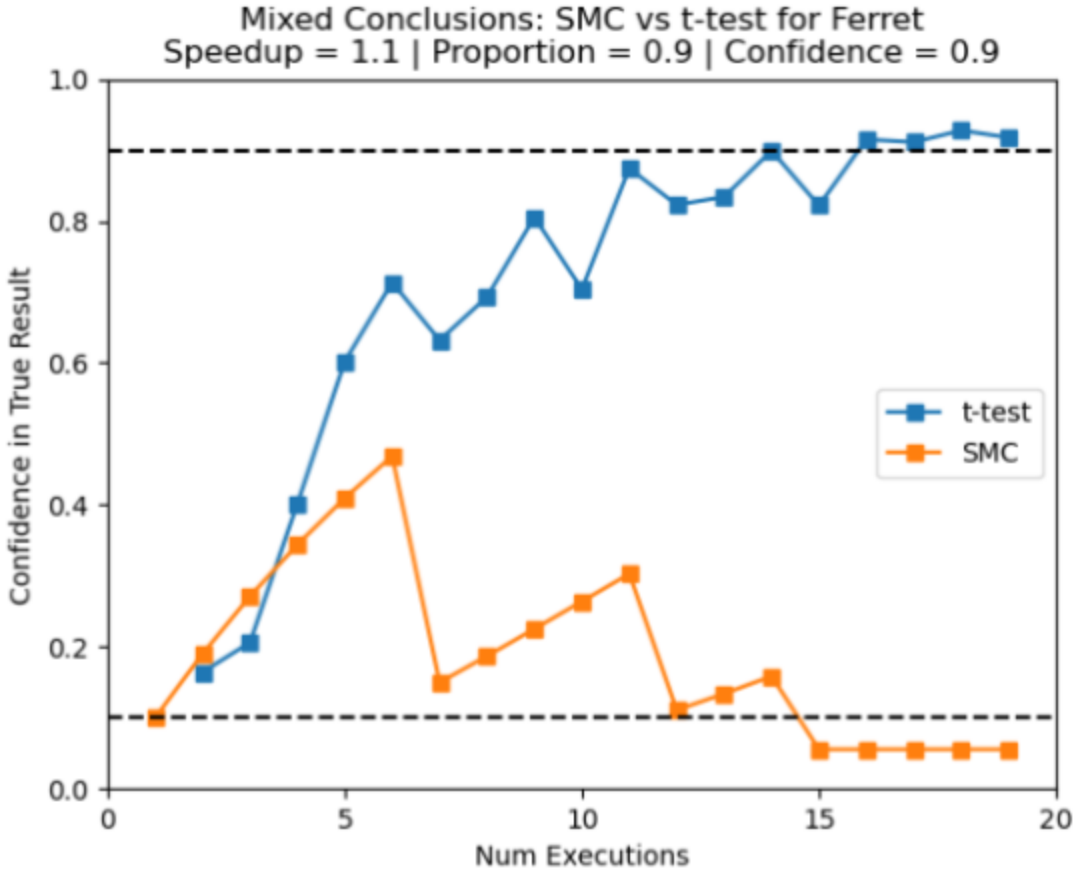
δ	$1-\varepsilon$	α	Acc.	Sam.	Time (s)	Ans.
1.00	0.50	0.01	1.00	1.6e+01	2.6e+00	True
1.00	0.50	0.05	1.00	1.1e+01	1.7e+00	True
1.00	0.70	0.01	1.00	6.6e+00	1.3e+00	True
1.00	0.70	0.05	1.00	4.9e+00	7.8e-01	True
1.00	0.90	0.01	1.00	3.6e+00	6.5e-01	True
1.00	0.90	0.05	1.00	2.5e+00	4.4e-01	True
1.05	0.50	0.01	1.00	3.6e+01	5.9e+00	True
1.05	0.50	0.05	0.99	1.9e+01	3.8e+00	True
1.05	0.70	0.01	1.00	9.7e+00	1.8e+00	True
1.05	0.70	0.05	1.00	6.3e+00	1.3e+00	True
1.05	0.90	0.01	1.00	4.6e+00	8.1e-01	True
1.05	0.90	0.05	1.00	3.1e+00	5.6e-01	True
1.10	0.50	0.01	1.00	3.3e+02	5.5e+01	False
1.10	0.50	0.05	**0.92**	1.2e+02	2.3e+01	False
1.10	0.70	0.01	1.00	5.6e+01	9.6e+00	True
1.10	0.70	0.05	0.99	2.9e+01	5.0e+00	True
1.10	0.90	0.01	1.00	8.9e+00	1.5e+00	True
1.10	0.90	0.05	1.00	6.6e+00	1.1e+00	True

Application: Microarchitecture Performance and *Security*

Duke
PRATT SCHOOL of ENGINEERING

[MICRO'23]

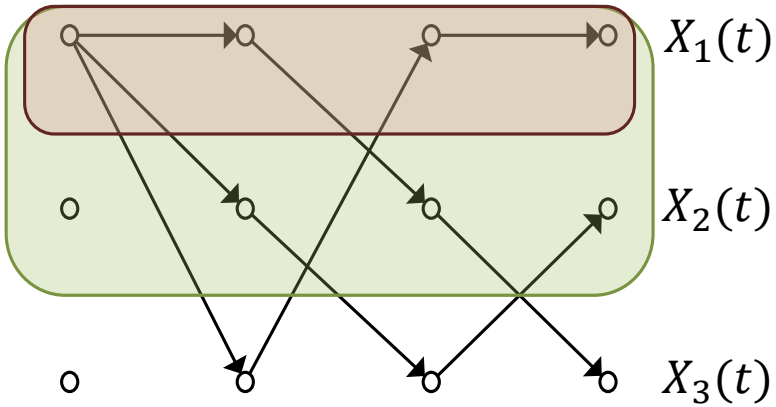


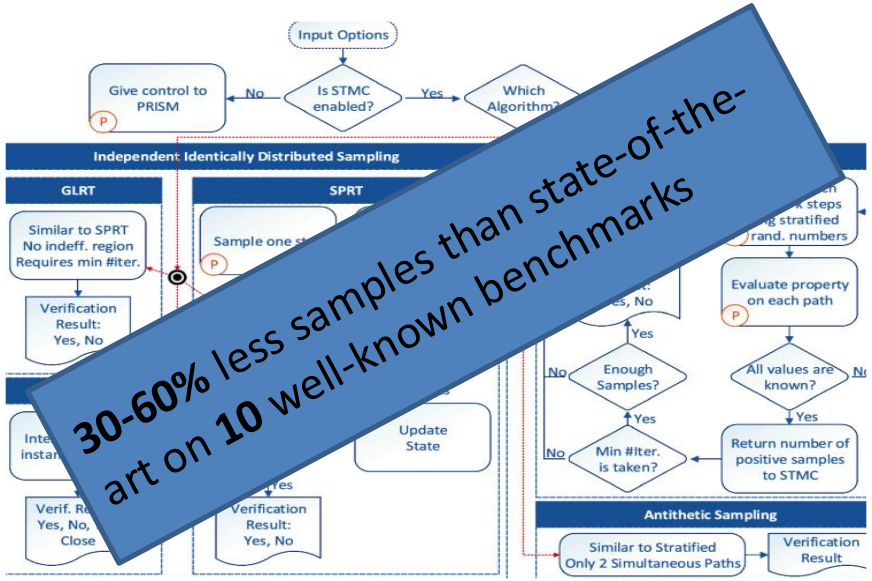


Beyond I.I.D. Samples

For a class of LTL specifications, step-wise anticorrelation of paths X_i and X_j leads to anticorrelation in satisfaction, i.e.,

$$\Pr(X_i \text{ satisfies } \varphi \mid X_j \text{ satisfies } \varphi) \leq \Pr(X_i \text{ satisfies } \varphi)$$





https://github.com/nima-roohi/STMC

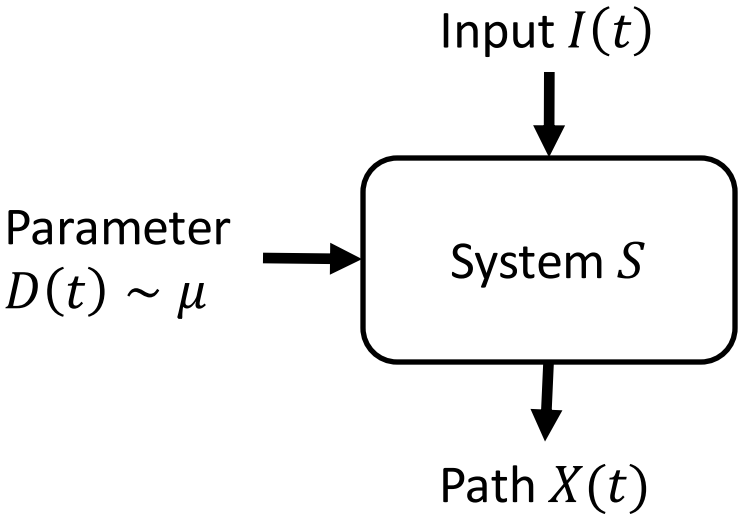
[FMSD'19, CAV'20]

Anticorrelation leads to faster SMC.

Generalization to Hyper Specifications

Duke
PRATT SCHOOL of ENGINEERING

Many system properties cannot be determined from single system path from a fixed input.



For example, resiliency in specification φ requires that

$$\Pr(X_1 \text{ satisfies } \varphi \mid \text{Input } \mathrm{I}_1) > \frac{1}{2}\Pr(X_2 \text{ satisfies } \varphi \mid \text{Input } \mathrm{I}_2)$$

Questions:

1. How to express these hyper specifications in temporal logic?
2. How to develop statistical algorithms to check them?

New Hyper Temporal Logic

Duke
PRATT SCHOOL of ENGINEERING

New Hyper Temporal Logic:

$$\varphi ::= \mathrm{a}^{\pi} \mid \varphi^{\pi} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \boldsymbol{U_T} \, \varphi \mid p \sim p$$

$$p ::= \mathbb{P}^{\Pi}\varphi \mid \mathbb{P}^{\Pi}p \mid f(p, \ldots, p)$$

- a replaced by a^{π}, π is a path variable,
- $\mathbb{P}$ replaced by $\mathbb{P}^{\Pi}$, Π is a set of path variables,
- $\mathbb{P}_{\sim \boldsymbol{p}} \, \varphi$ replaced by a set of rules $p ::= \mathbb{P}^{\Pi}\varphi \mid \mathbb{P}^{\Pi}p \mid f(p, \ldots, p)$ and $p \sim p$

The new logic is well-defined.

New Statistical Algorithms

Duke
PRATT SCHOOL of ENGINEERING

The new hyper features requires new statistical techniques.

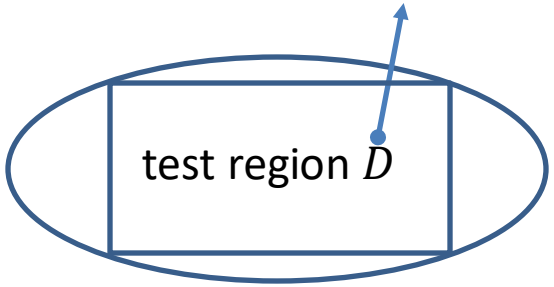
Based on analysis of the hyper logic:

- **Multiple** paths $\mathbb{P}^{(\pi_1,\pi_2)}\varphi^{(\pi_1,\pi_2)} < p$
- **Nested** probabilities $\mathbb{P}^{\pi_1}\left(\mathbb{P}^{\pi_2}\varphi^{(\pi_1,\pi_2)} < p_2\right) < p_1$
- **Joint** probabilities $(\mathbb{P}^{\pi_1}\varphi_1, \mathbb{P}^{\pi_2}\varphi_2) \in D$

Joint Probabilities

For $(\mathbb{P}^{\pi_1}\varphi_1, \mathbb{P}^{\pi_2}\varphi_2) \in D$, we need to generalize the Clopper-Pearson Exact Method to the multi-dimensional case.

$$\left(\frac{T_1}{N_1}, \frac{T_2}{N_2}\right) \in [a_1, b_1] \times [a_2, b_2] \subset D$$



Compute an upper bound for a **simple** region $[a_1, b_1] \times [a_2, b_2] \subset D$

The confidence level is at most

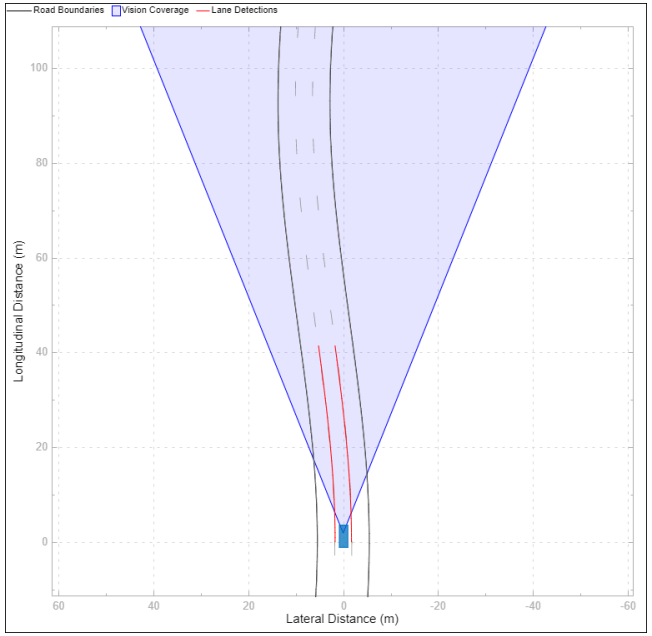
$$1 - \prod_{i=1}^{2} \alpha_{\mathrm{CP}}\left(a_i, b_i \mid T_i, N_i\right)$$

[EMSOFT 19, Best Paper Finalist]

Further Generalization to Conformance

Consider parametrized specifications φ_λ. Conformance requires

$$\forall \lambda > 0, |\Pr(X_1 \text{ satisfies } \varphi_\lambda \mid \text{Input } \mathrm{I}_1) - \Pr(X_2 \text{ satisfies } \varphi_\lambda \mid \text{Input } \mathrm{I}_2)| < c$$



For example, we want to check if

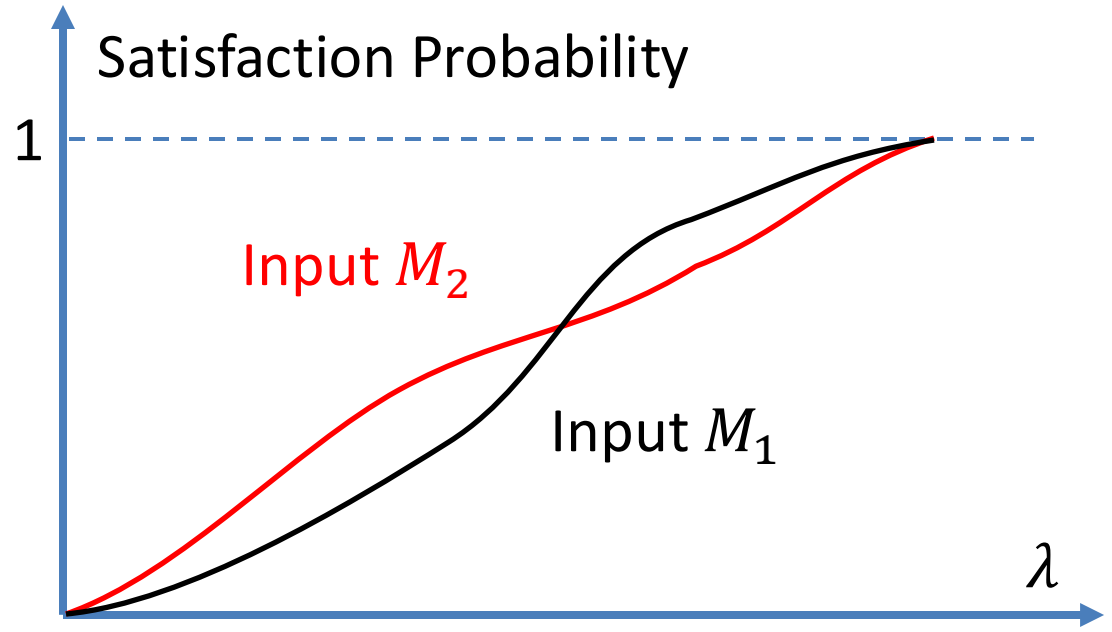
a traditional and learning-based controller have similar performance for any speed λ

to provide post-hoc explainability.

Questions: How to develop statistical algorithms to check them?

Further Generalization to Conformance

- In general, checking conformance is hard.
- But it is possible if the satisfaction probability of φ_λ is monotone in λ.



We can use non-parametric Kolmogorov-Smirnov test their similarity.

Simulation Results

Duke
PRATT SCHOOL of ENGINEERING

Conformance in Track Error of MPC/NN-based Lane-Keeping Controllers.

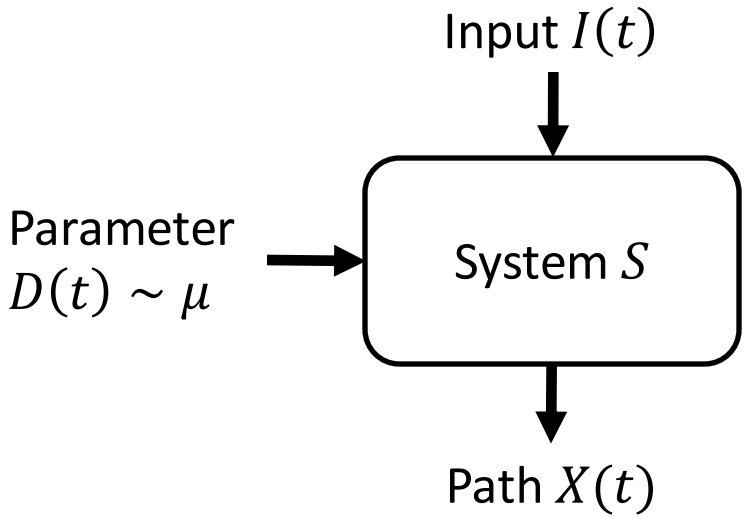
$$\forall \tau \geq 0.\ \left| \mathbf{Pr}_{\sigma_1 \sim \mathcal{M}_{\mathrm{NN}}}(\sigma_1 \vDash \Diamond_{[0,\tau]}(|e_y^{\mathrm{NN}}| < \gamma)) - \mathbf{Pr}_{\sigma_2 \sim \mathcal{M}_{\mathrm{MPC}}}(\sigma_2 \vDash \Diamond_{[0,\tau]}(|e_y^{\mathrm{MPC}}| < \gamma)) \right| < c$$

c	Confidence	Samples	Time (s)	Result
0.40	0.99	1.0e+04	9.6e+00	T
0.40	0.95	3.6e+03	2.0e+00	T
0.25	0.99	9.5e+02	3.2e-01	F
0.25	0.95	2.5e+02	5.9e-02	F
0.10	0.99	2.1e+02	4.2e-02	F
0.10	0.95	1.2e+02	2.2e-02	F
0.05	0.99	1.3e+02	2.5e-02	F
0.05	0.95	7.3e+01	1.4e-02	F

Dealing with Non-fixed Input using Reinforcement Learning

Duke
PRATT SCHOOL of ENGINEERING

Check if for *any* input $I(t)$, $\Pr(S \text{ satisfies } \varphi) > c$.



How to find the worst input $I(t)$ that minimizes the satisfaction probability?

- The problem is hard for general systems.
- For Markov Decision Processes, we can use reinforcement learning algorithm to find worst-case $I(t)$.

Case Study: Sum of two dice

Duke
PRATT SCHOOL of ENGINEERING

$\|S\|$	$\|A\|$	H	p	PRISM	Ans.	Iter.	Time (s)
3	3	4	0.25	0.35	False	208.5	0.02
		4	0.45		True	3528.7	0.19
4	2	4	0.09	0.19	False	171.8	0.01
		4	0.29		True	3671.7	0.22
5	2	4	0.05	0.11	False	441.7	0.04
		4	0.21		True	4945.5	0.42
10	2	4	0.04	0.09	False	544.7	0.14
		4	0.19		True	5193.3	1.45
15	2	3	0.04	0.09	False	873.1	0.28
		3	0.19		True	4216.7	1.28
20	4	5	0.12	0.22	False	337.6	0.99
		5	0.32		True	9353.3	28.06
25	5	10	0.09	0.19	False	270.5	7.57
		10	0.29		True	25709.8	728.49
30	5	10	0.08	0.17	False	355.6	14.35
		10	0.27		True	27161.7	1085.77
35	5	10	0.09	0.18	False	328.9	18.82
		10	0.28		True	27369.6	1529.84
40	5	10	0.08	0.16	False	390.0	26.79
		10	0.26		True	30948.8	2122.24

TABLE I

CHECKING $\mathbf{P}^{\max}_{<p}(\alpha_1 \mathbf{U}_H \alpha_2)$. "ANS." IS RETURN OF ALGORITHM 2. "ITER." IS AVERAGE NUMBER OF ITERATIONS. "PRISM" IS PRISM'S ESTIMATION OF SATISFACTION PROBABILITY.

$\|S\|$	$\|A\|$	p	PRISM	Ans.	H_1	H_2	Iter.	Time (s)
3	3	0.25	0.35	False	14.7	15.8	126.0	0.03
		0.45		True	12.9	13.0	111.3	0.01
4	2	0.09	0.19	False	13.0	27.2	103.7	0.01
		0.29		True	9.6	19.2	73.0	0.01
5	2	0.05	0.11	False	12.4	59.3	239.9	0.06
		0.21		True	6.9	24.3	92.7	0.00
10	2	0.04	0.09	False	3.2	225.6	891.4	0.24
		0.19		True	1.1	21.5	79.6	0.00
15	2	0.04	0.09	False	1.3	117.8	1862.0	0.61
		0.19		True	1.0	11.0	161.3	0.01
20	4	0.12	0.22	False	1.0	1.0	1336.2	0.27
		0.32		True	1.0	1.8	16843.8	3.02
25	5	0.09	0.19	False	1.0	1.0	1619.2	0.64
		0.29		True	1.0	2.0	154621.6	56.56
30	5	0.08	0.17	False	1.0	1.0	2246.8	1.05
		0.27		True	1.0	1.3	86617.0	42.53
35	5	0.09	0.18	False	1.0	1.0	1925.1	0.82
		0.28		True	1.0	1.0	13219.0	5.79
40	5	0.08	0.16	False	1.0	1.0	2401.5	1.35
		0.26		True	1.0	1.0	12158.6	6.66

TABLE II

CHECKING $\mathbf{P}^{\max}_{<p}(\alpha_1 \mathbf{U} \alpha_2)$. H_1 AND H_2 ARE THE TIME STEPS NEEDED TO MODEL CHECK THE FORMULA AND ITS NEGATION, UPDATED BY (20).

Results

RL can be used to find worst-case input $I(t)$ for Markov decision processes **(TAC24)**.

Case Study I:

Safe Absorbing States

Case Study II:

Nursery Scenario

CSRL -- **C**ontrol **S**ynthesis by **R**einforcement **L**earning (https://gitlab.oit.duke.edu/cpsl/csrl)

Thank you


AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE


DEPARTMENT OF THE NAVY
USN
ONR
Science & Technology


NSF


U.S. DEPARTMENT OF HOMELAND SECURITY