Statistical Hypothesis Testing of Controller Implementations Under Timing Uncertainties

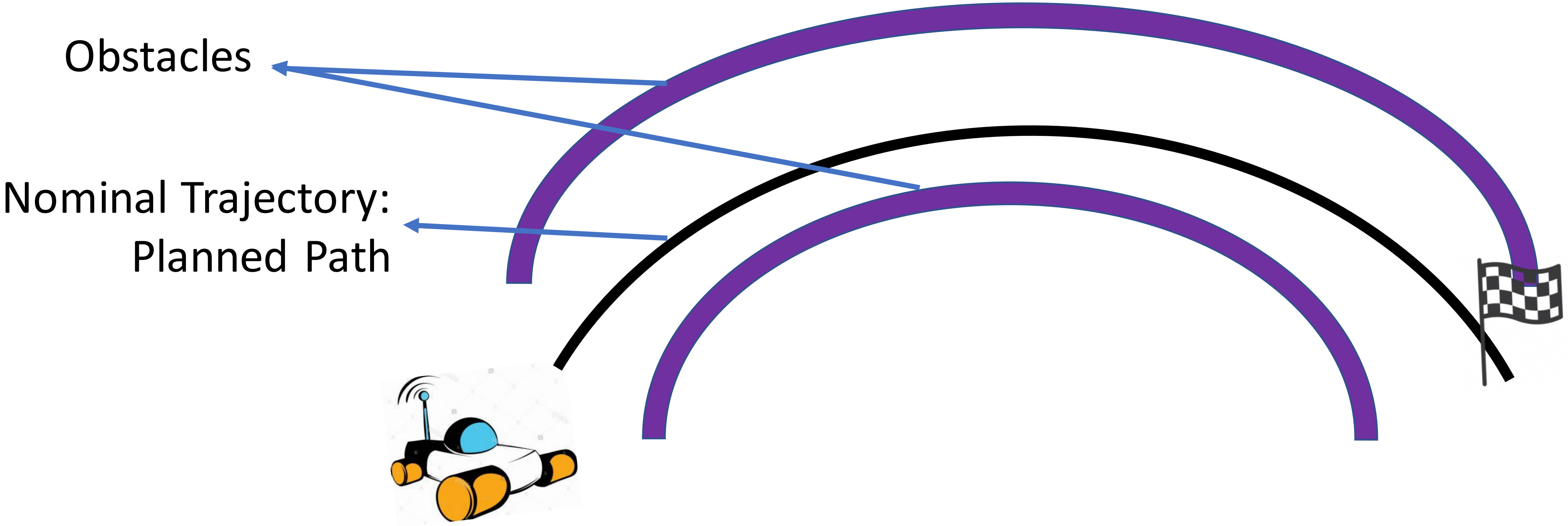
Authors: Bineet Ghosh, Clara Hobbs, Shengjie Xu, Parasara Sridhar Duggirala, James H. Anderson, P. S. Thiagarajan, Samarjit Chakraborty

RTCSA 2022

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Quantitative Safety: Robot Maneuvers

Robot trying to reach its destination, avoiding obstacles.



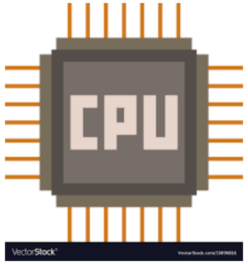
Quantitative Safety: Robot Maneuvers

But: The robot is running multiple jobs on its processor!

Responsible for moving the robot along the planned trajectory.

Multiple Jobs

- Path Follower
- ⋮
- Perception
- Heat Control



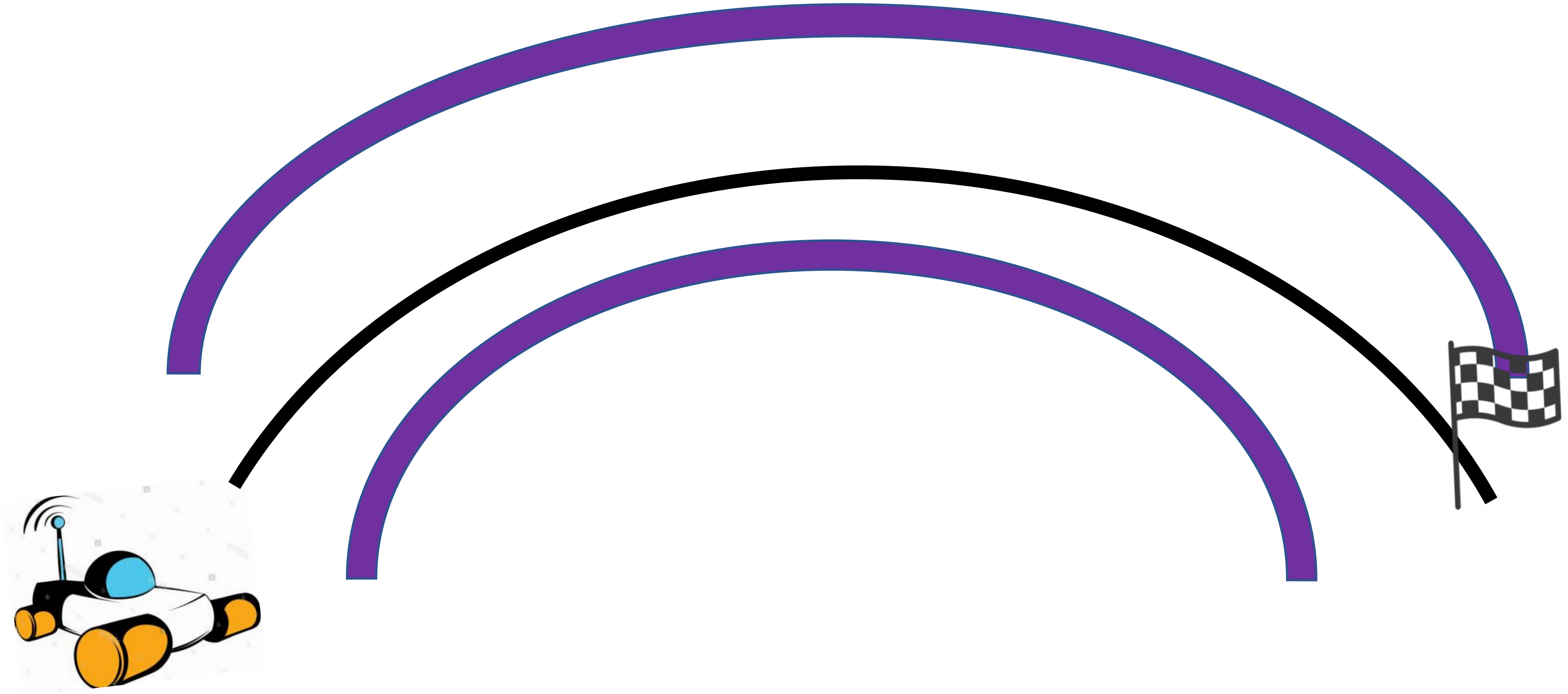
All jobs cannot always be scheduled—deadline misses!

What if the path follower misses its deadline?

Quantitative Safety: Robot Maneuvers

What if the path follower misses some deadlines?

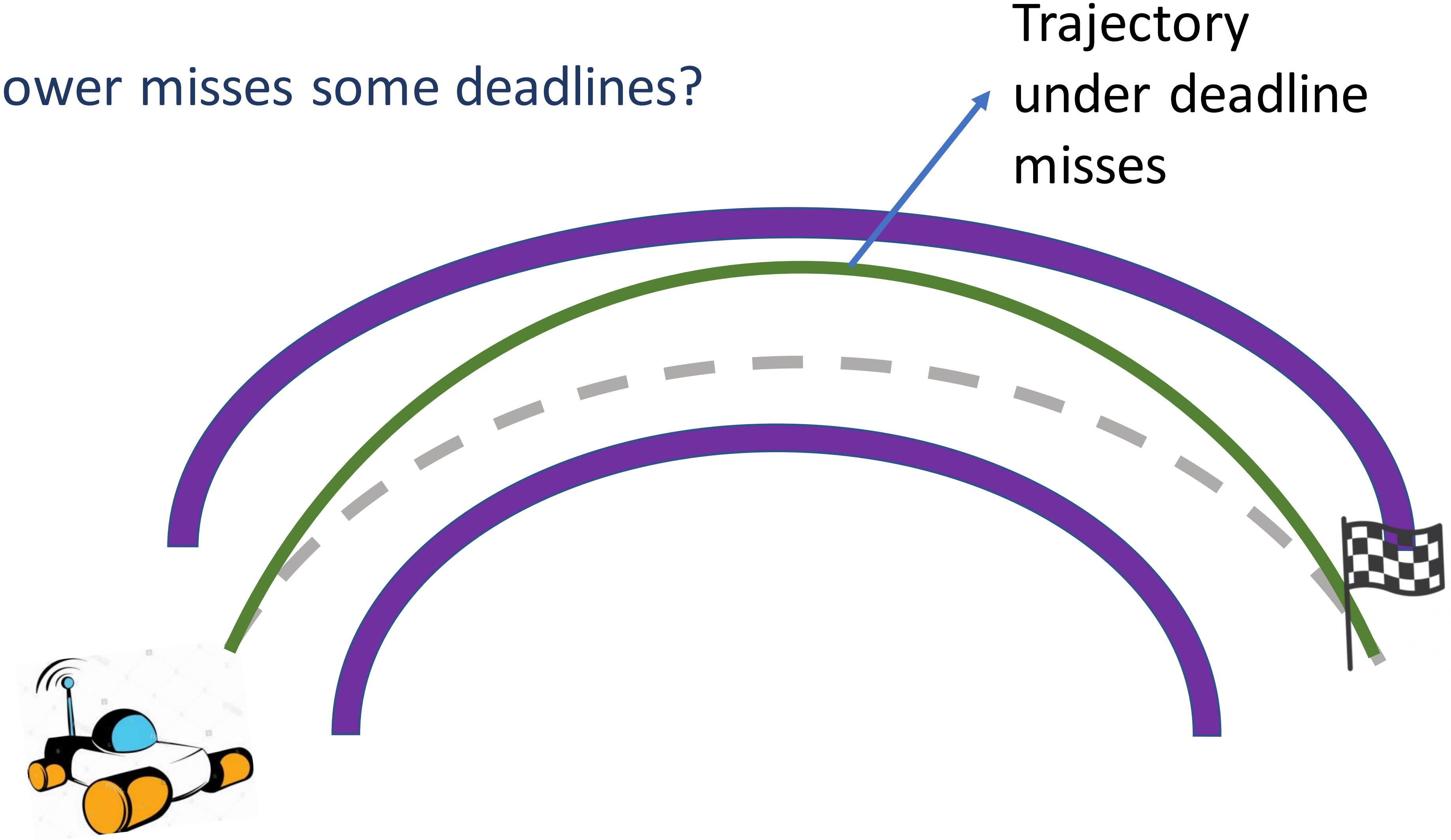
The trajectory can deviate from the nominal trajectory!



Quantitative Safety: Robot Maneuvers

What if the path follower misses some deadlines?

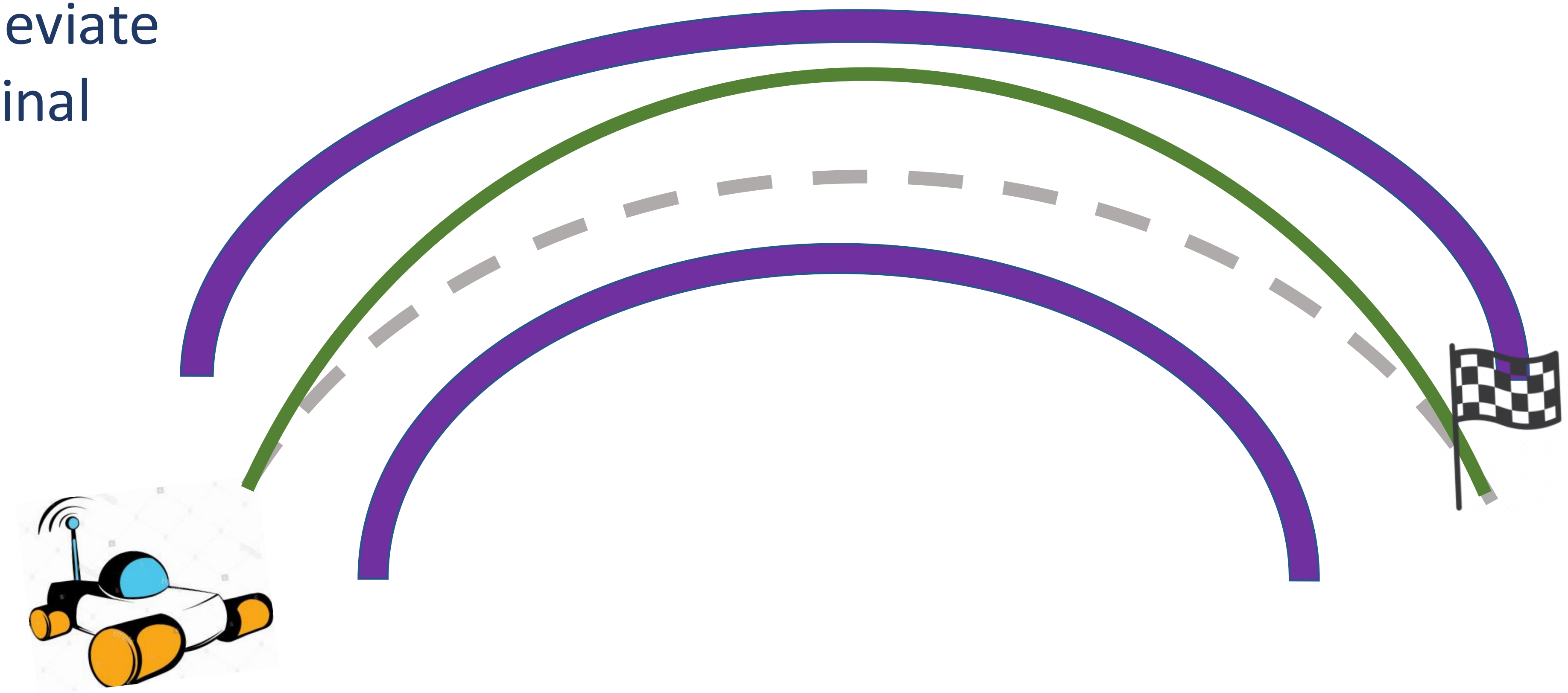
The trajectory can deviate from the nominal trajectory!



Quantitative Safety: Robot Maneuvers

What if the path follower misses deadlines **very frequently**?

The trajectory can deviate **more** from the nominal trajectory!

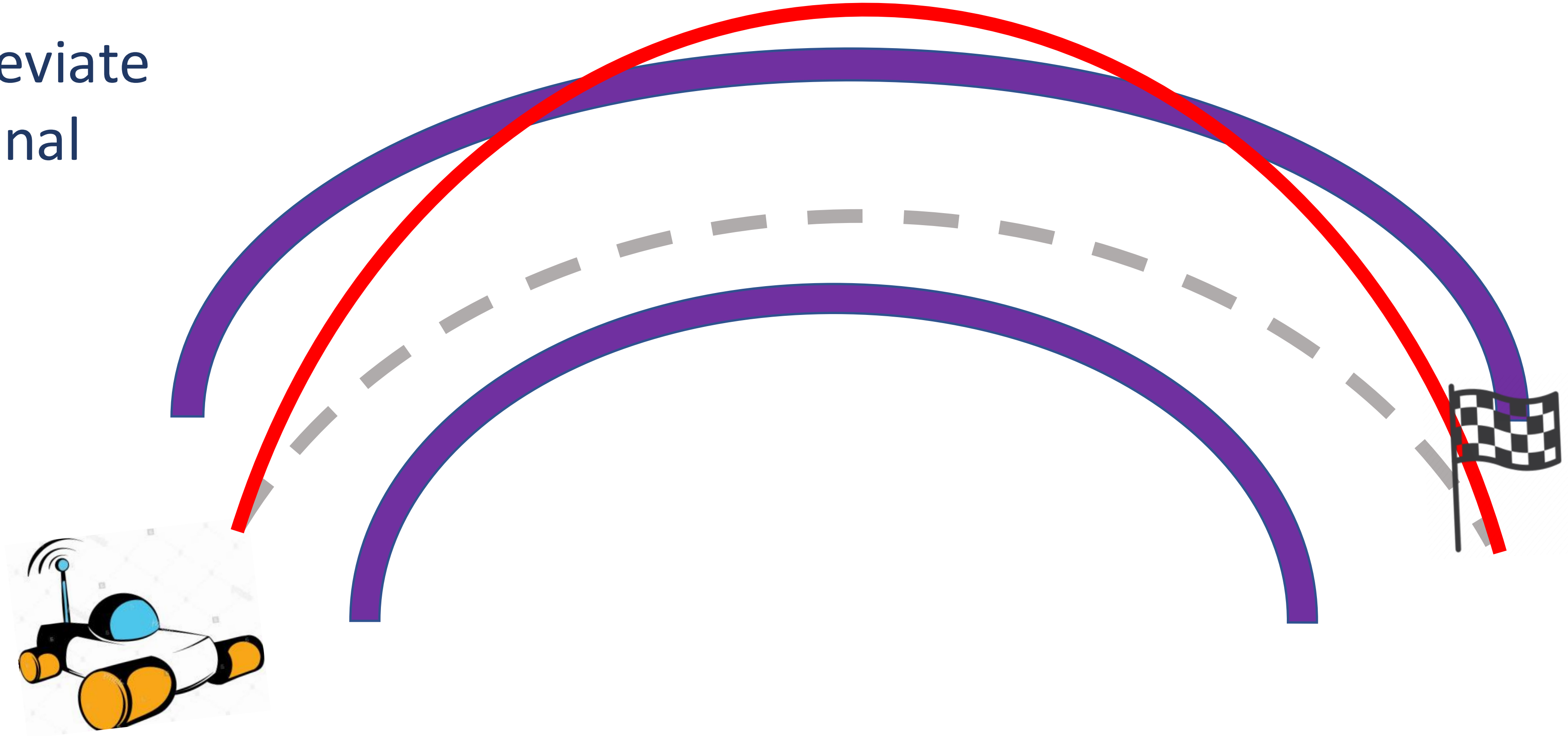


Quantitative Safety: Robot Maneuvers

What if the path follower misses deadlines **very frequently**?

The trajectory can deviate **more** from the nominal trajectory!

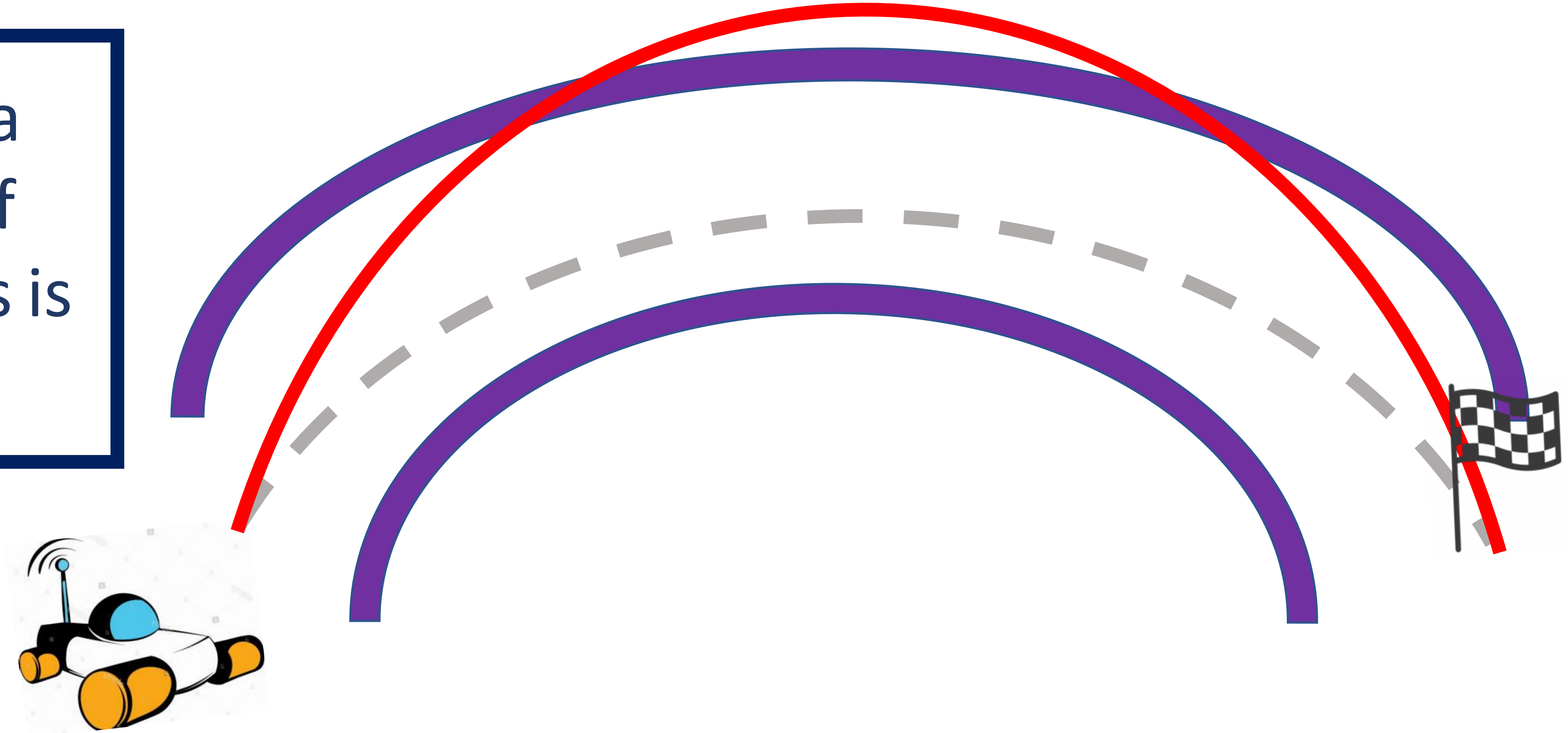
And become unsafe!



Quantitative Safety: Robot Maneuvers

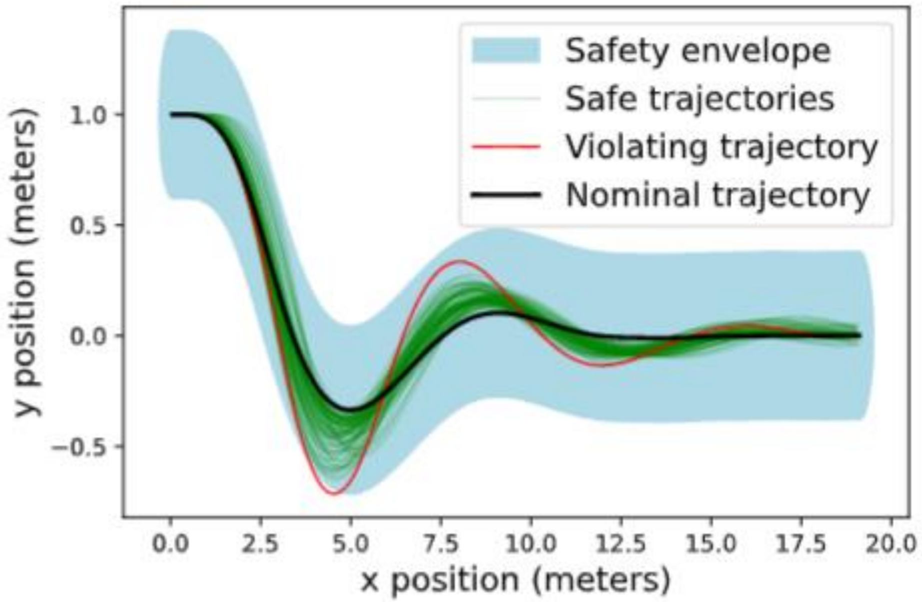
In Conclusion: Not all *patterns* of deadline misses are ***safe***!

Goal: Detect if a given *pattern* of deadline misses is safe!



Does Stable Means Safe?

F1 Tenth Simulation Case Study



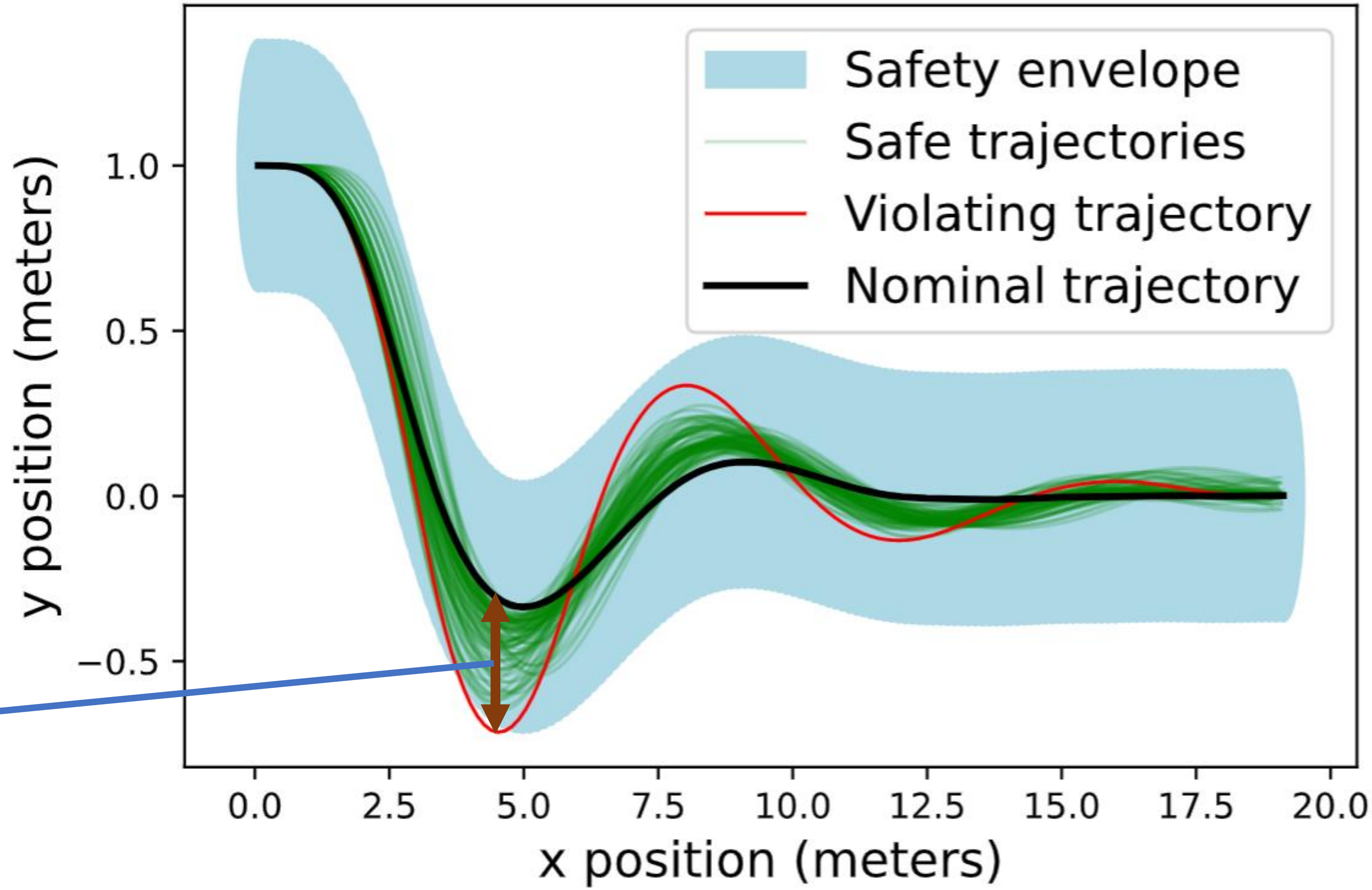
All trajectories are stable!

Yet some violate safety!

Scheduling with Deadline Misses

Goal: Compute:

Maximum Deviation



Computing Deviation: A Naïve Approach

- Given a pattern of deadline misses.
- Compute the maximum deviation up-to a bounded time H.

A Possible Behavior up-to Time H: 1 1 0 1 ⋯ 1 0 0 1

0/1-sequence of length H

0: Deadline Miss.

1: Deadline Hit (No Miss).

Computing Deviation: A Naïve Approach

- Given a pattern of deadline misses.
- Compute the maximum deviation up-to a bounded time H.
- **Naïve Approach:** Requires computing deviation of 2^H many trajectories!
- **Instead:** Compute an over-approximation of the maximum deviation.

Computing Deviation: Other Approaches

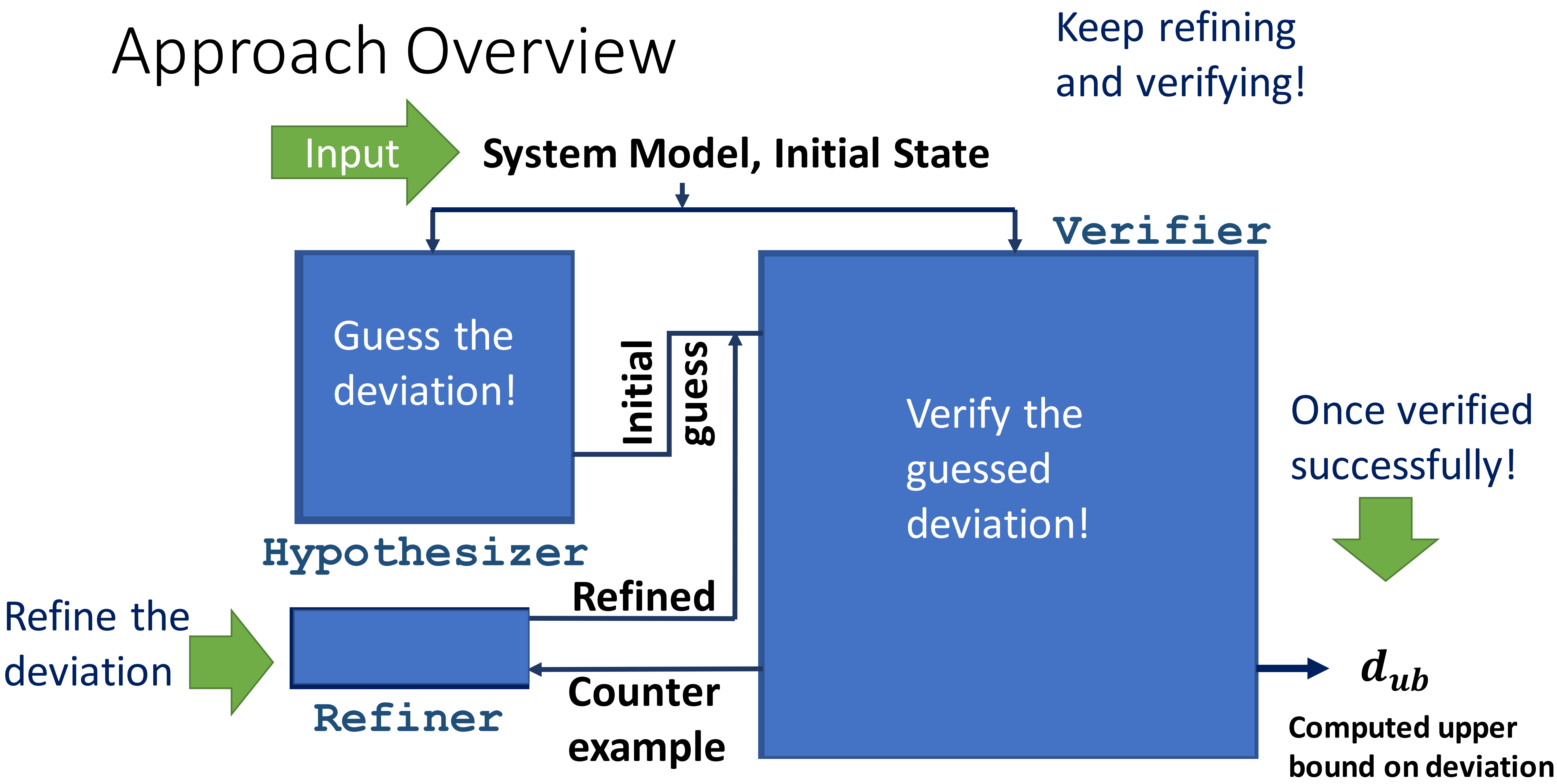
- Requires computing ***reachable sets***.

- **Disadvantages:**
  - Computationally slower (generally).
  - The computed bounds on the maximum deviation are not tight (generally).

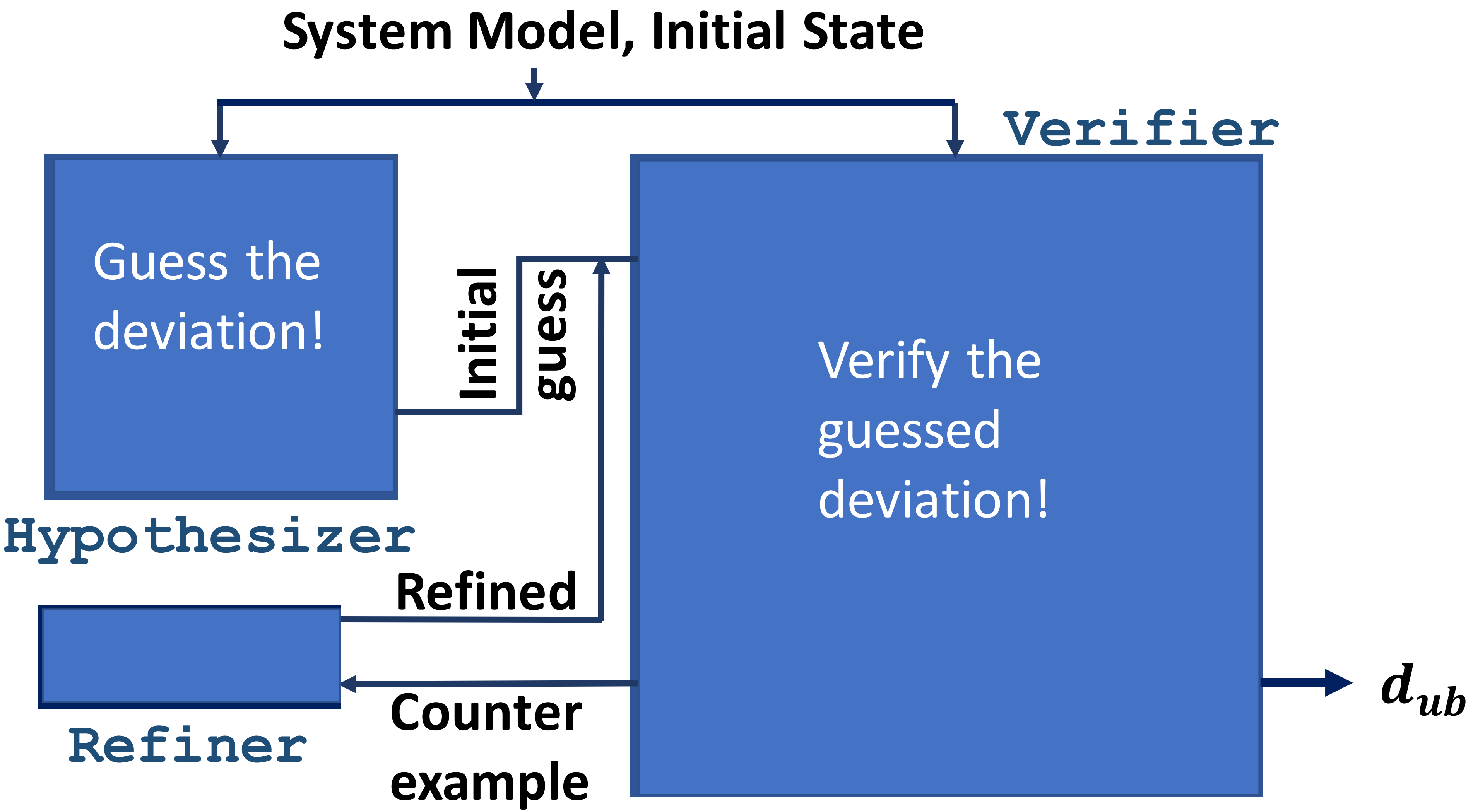
Contribution

- Compute an **upper bound** of the **maximum deviation under a *pattern* of deadline misses**.
- **Statistical Approach:** guarantees are probabilistic.
- **Advantages**:
  - Computationally faster than non-probabilistic approaches.
  - Tighter bounds on the computed maximum deviation.

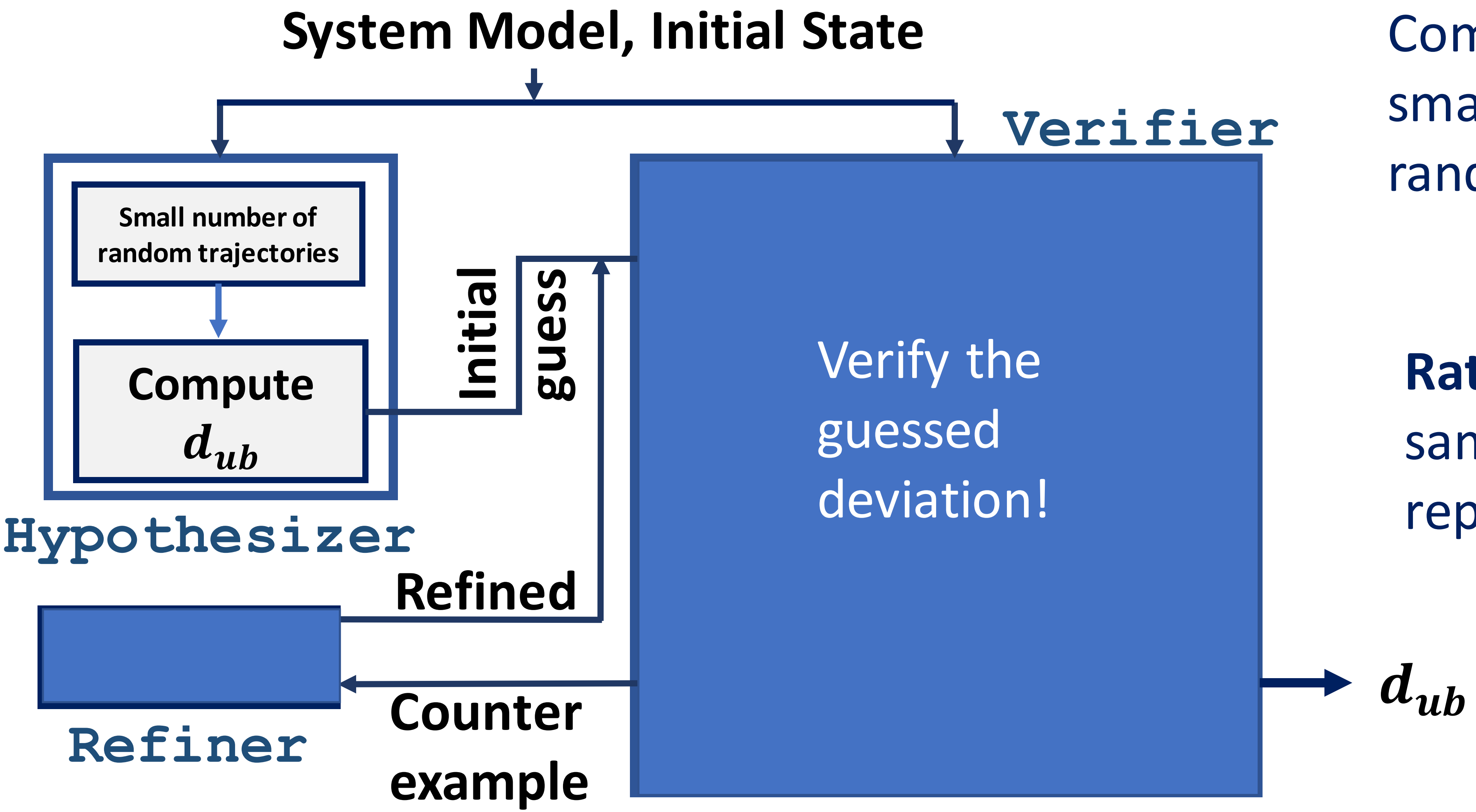
Approach Overview



Approach Overview



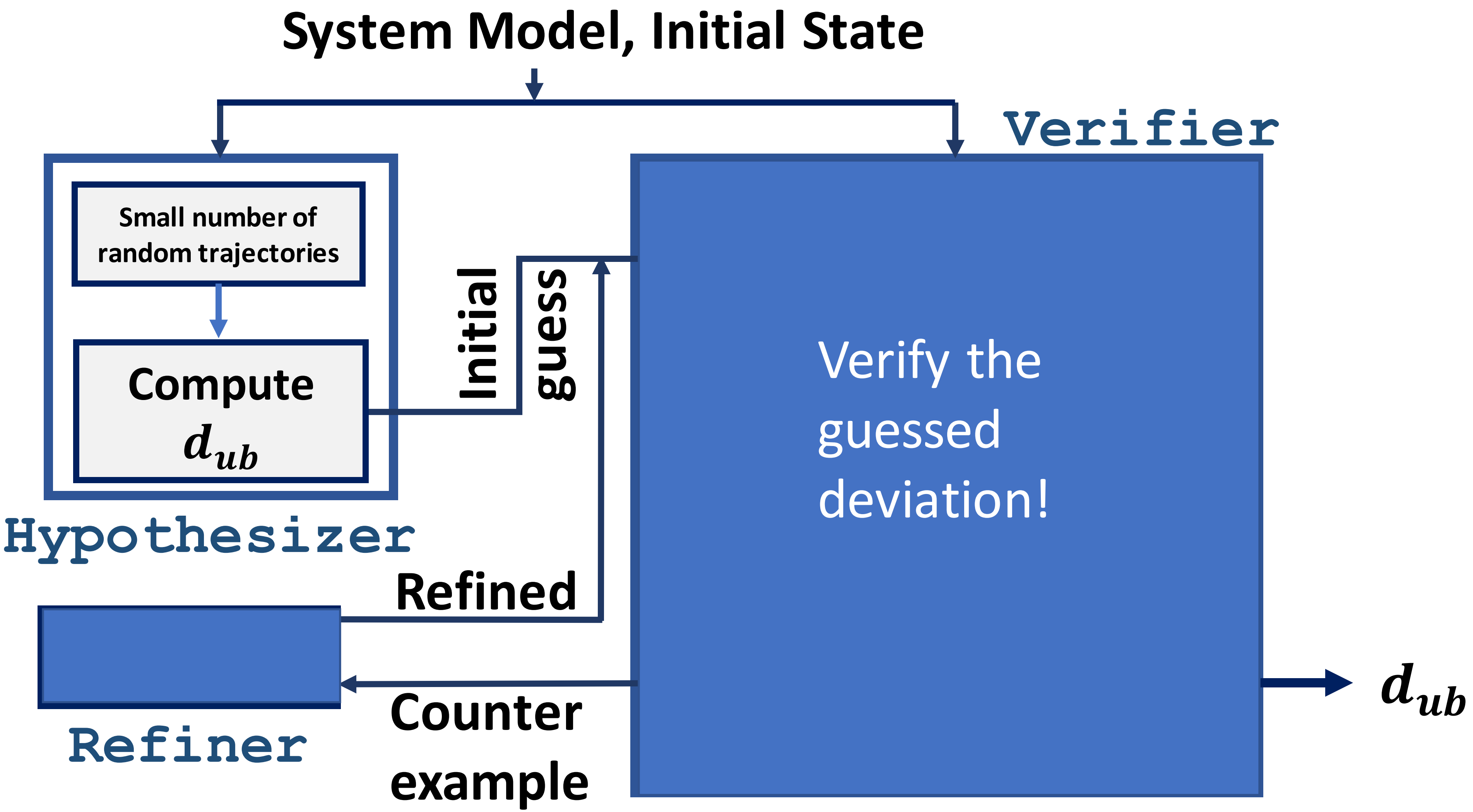
Approach Overview: Hypothesizer



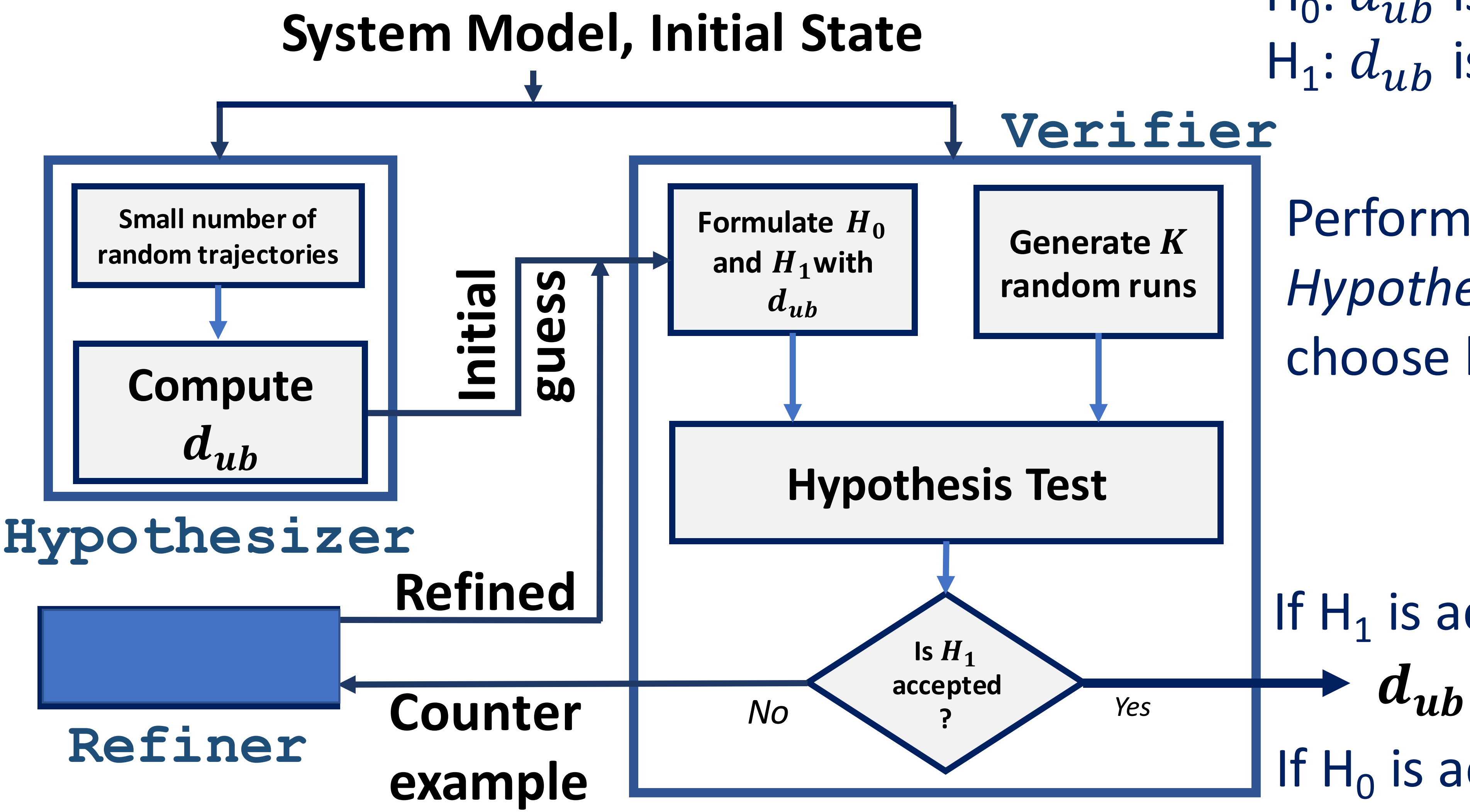
Compute d_{ub} using a small number of random trajectories.

Rationale: Small sample set might represent the *reality!*

Approach Overview



Approach Overview: Verifier



Formulate Hypotheses

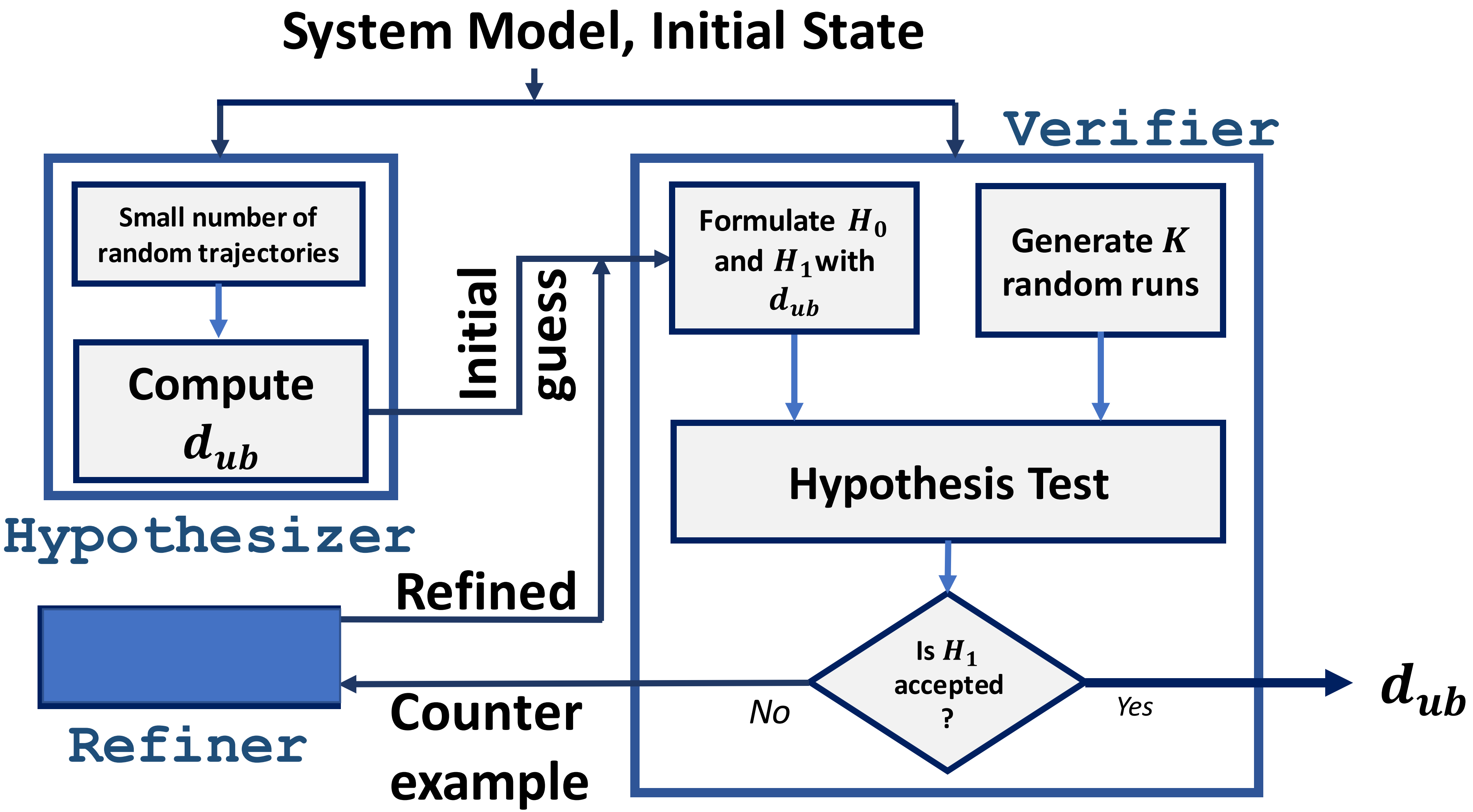
H_0: d_{ub} is ***not*** *acceptable!*
H_1: d_{ub} is *acceptable!*

Perform *Statistical Hypothesis Testing* to choose between H_0 and H_1.

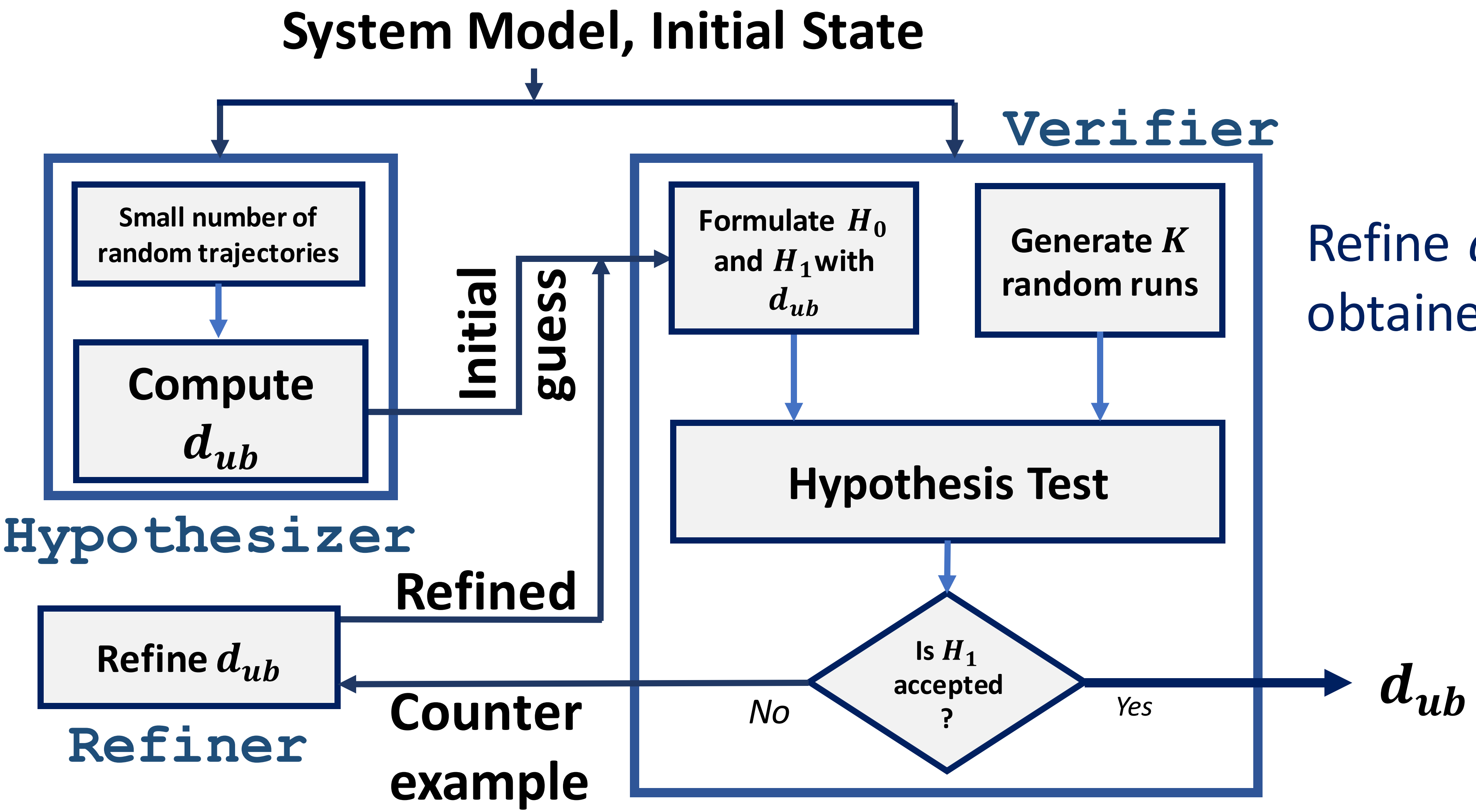
If H_1 is accepted: **return** d_{ub}.

If H_0 is accepted: **refine** d_{ub}.

Approach Overview



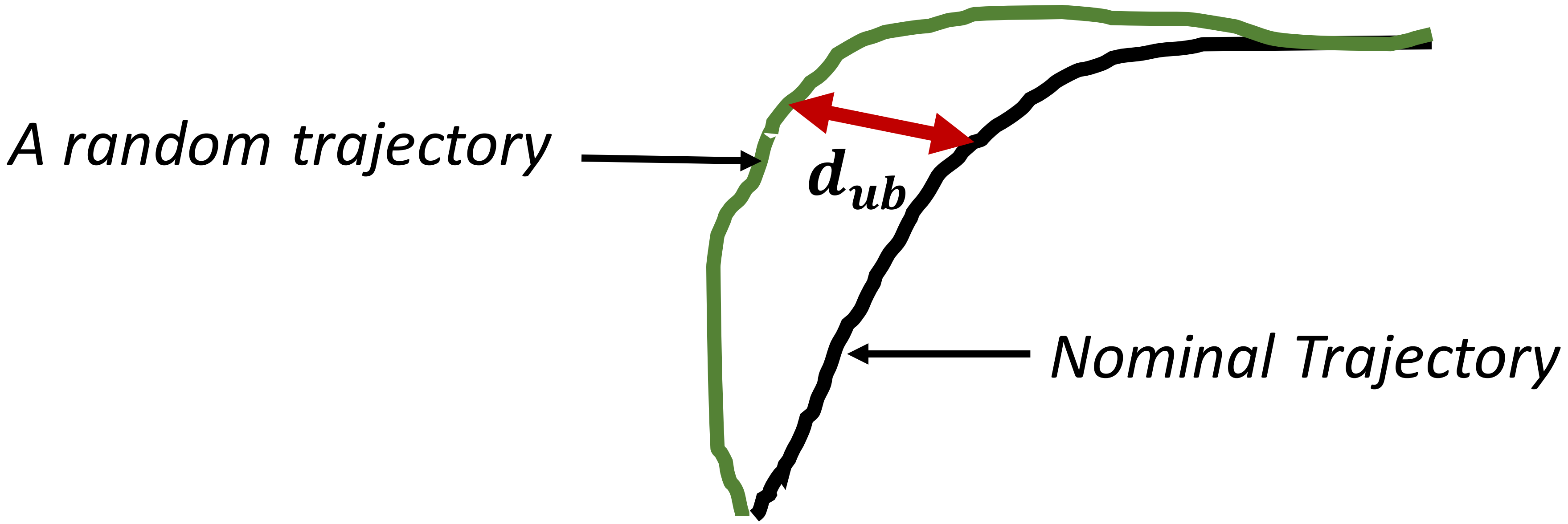
Approach Overview: Refiner



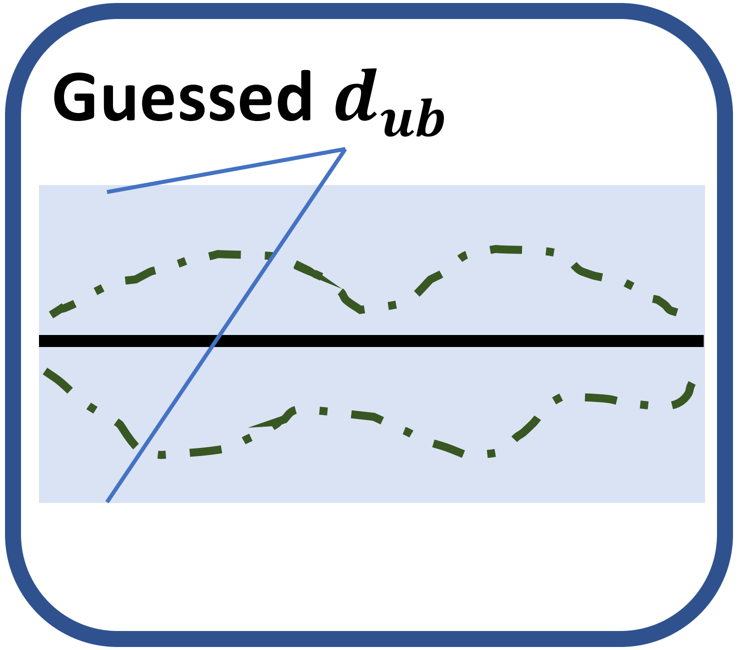
Refine d_{ub} using the obtained counter example.

Hypotheses (H_0 & H_1)

- H_0: **With a most probability *c***, *any* trajectory (random) will have a deviation bounded by d_{ub}.
- H_1: **With at least probability *c***, *any* trajectory (random) will have a deviation that is bounded by d_{ub}.



Approach Overview: Steps

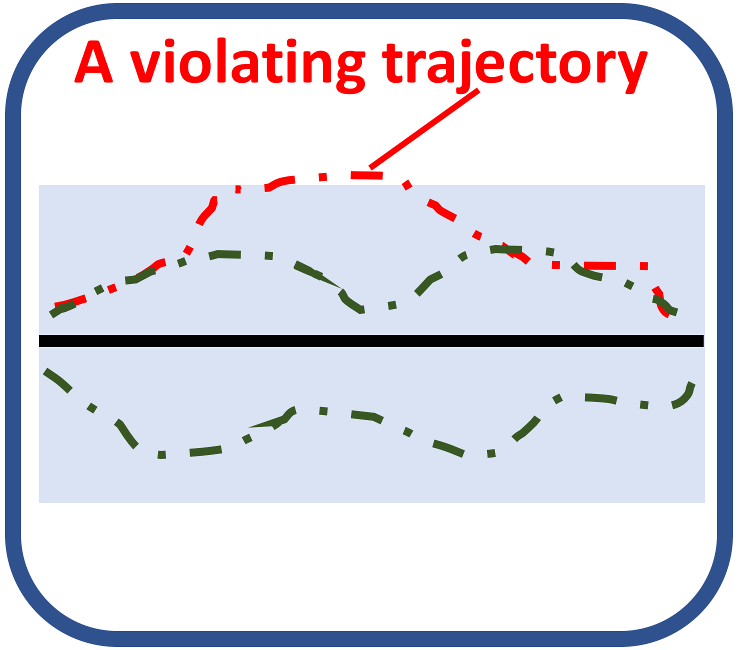


Step 1: Guess the deviation bound

Hypothesizer: *Generate few random trajectories and compute the maximum deviation.*

Black: Nominal Trajectory.
Green: Random Trajectories.
Light Blue: d_{ub}.

Approach Overview: Steps



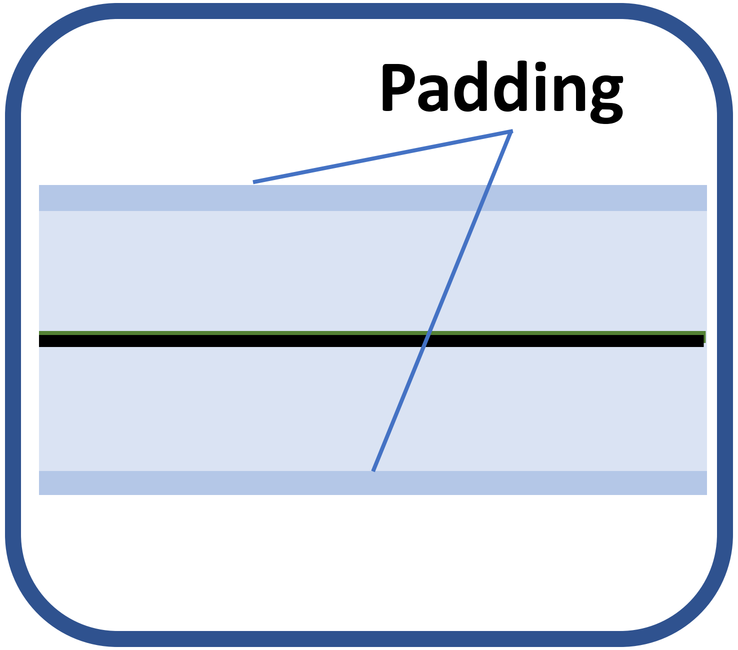
Step 2: Statistically verify the guessed bound

`Verifier:` *Verify d_{ub} by generating K random trajectories.*

K is computed using Jefferey's Bayes Factor based method.

If a *violating trajectory* is found (counter example), use it to refine d_{ub} (and re-verify)!

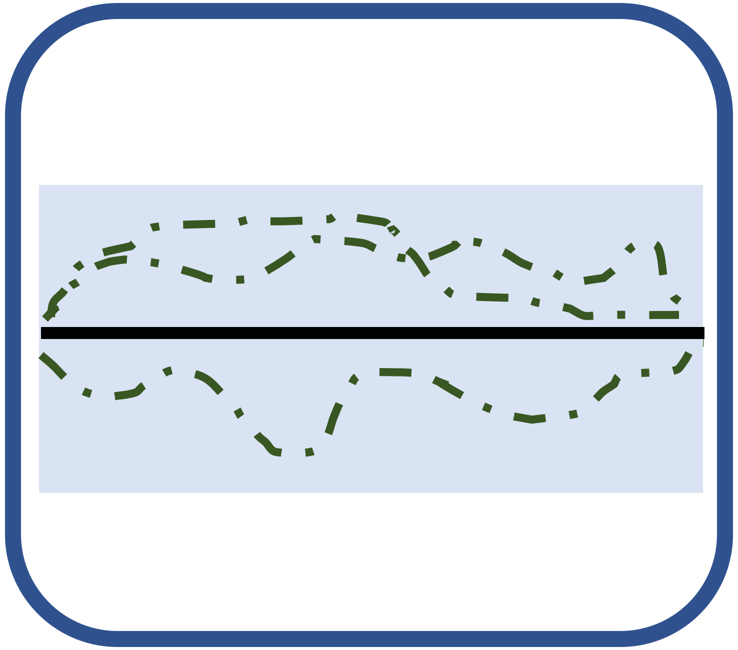
Approach Overview: Steps



Step 3: Refine the guessed bound

`Refiner:` *Pads the deviation bound obtained from the counterexample with slack ϵ.*

Approach Overview: Steps



Step 4: Statistically re-verify the guessed bound

Step 5: Return the accepted bound

Case Studies: Comparison with Benchmark Approaches

- RC Network

- Comparable upper bounds, and computation time.

- Electric Steering
- Unstable Second Order System
- F1 Tenth

- Computed significantly tighter bounds on the deviation.
- Significantly less computation time.

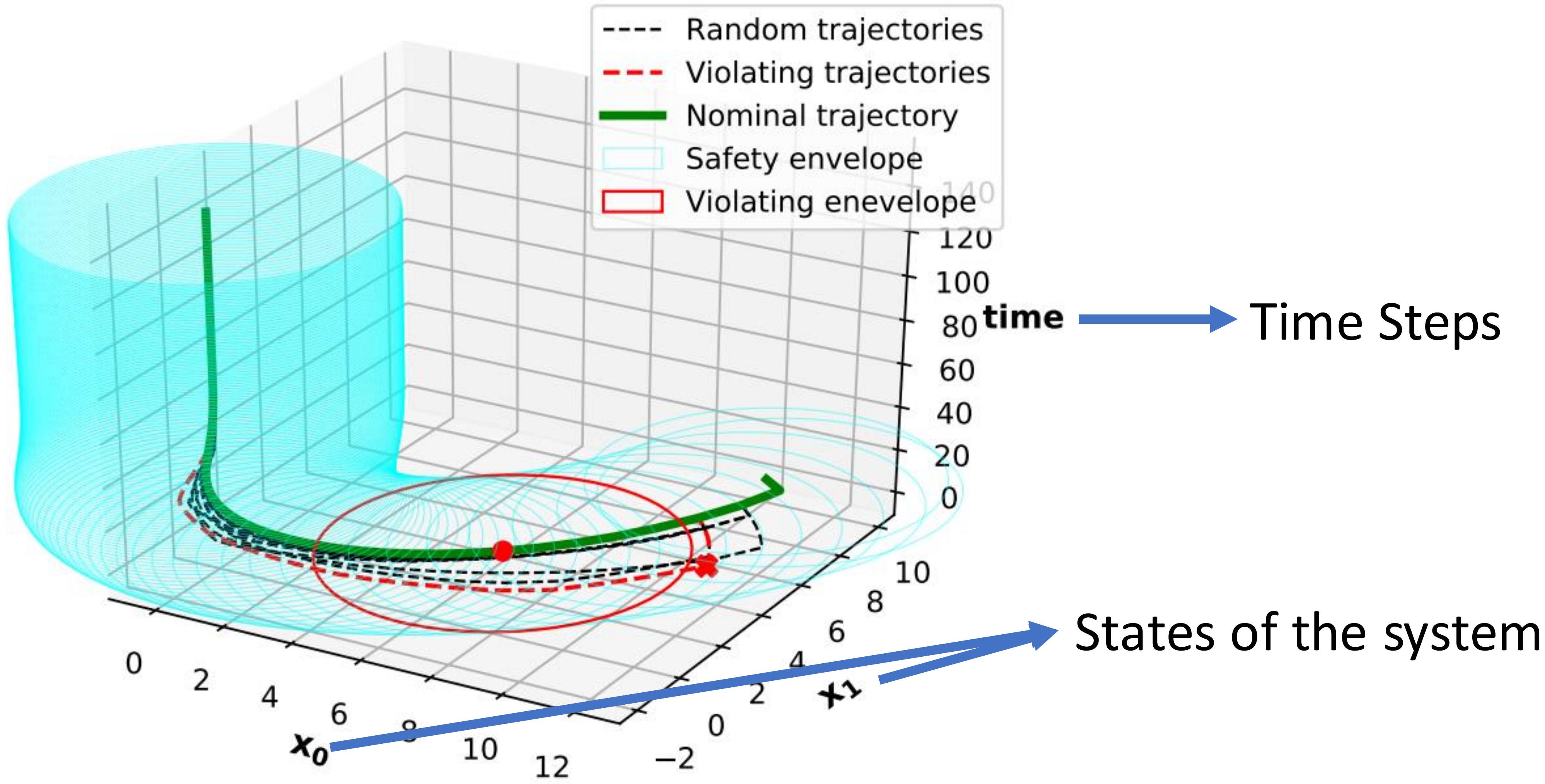
Case Studies: Comparison with Benchmark Approaches

- RC Network
- Electric Steering

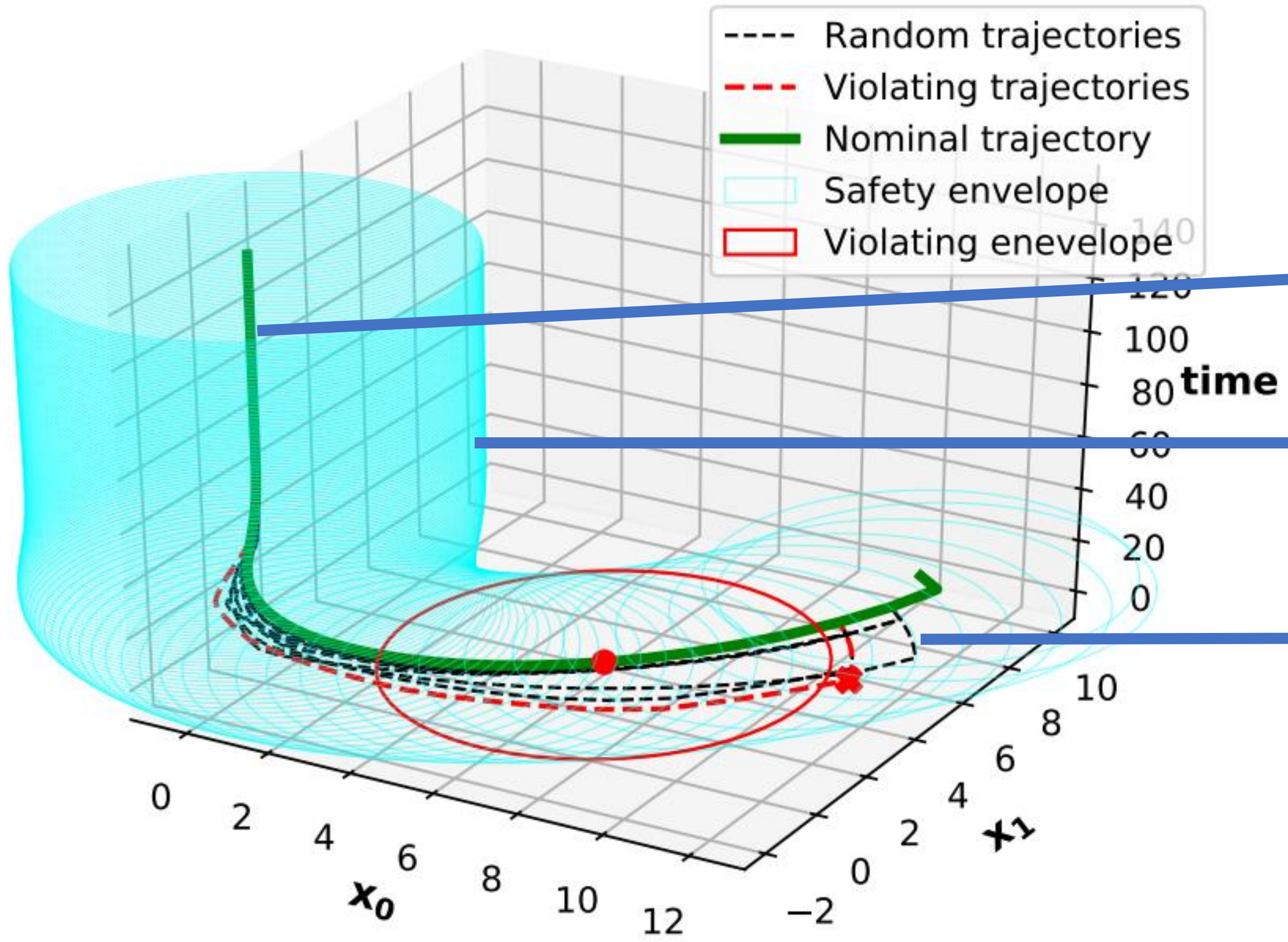
Discuss in this presentation!

- Unstable Second Order System
- F1 Tenth

Case Study: Electric Steering



Case Study: Electric Steering

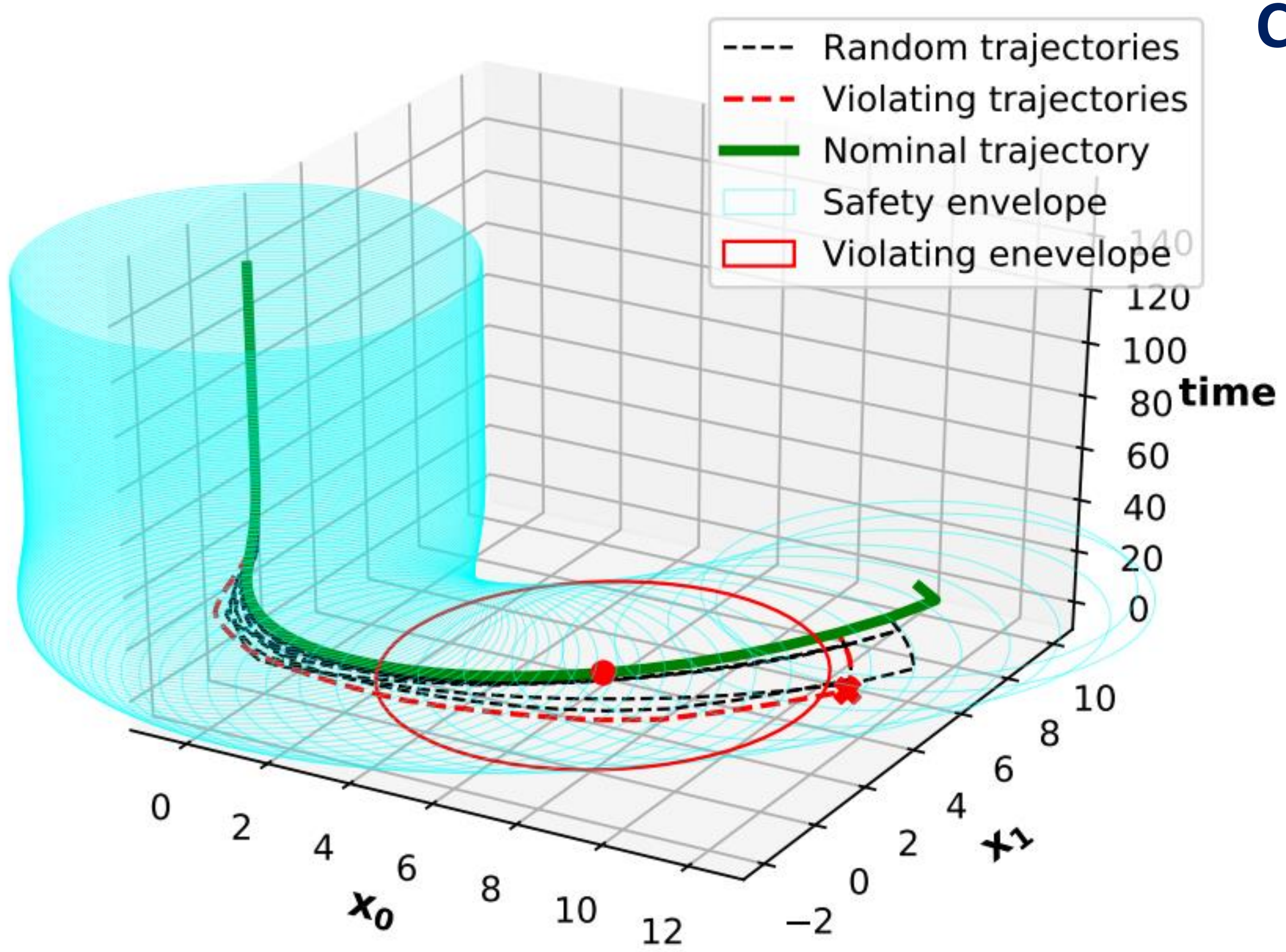


Nominal Trajectory (no deadline misses)

Safety Envelope

Few random trajectories with deadline misses (*at most 3 consecutive misses*)

Case Study: Electric Steering



Computed d_{ub}

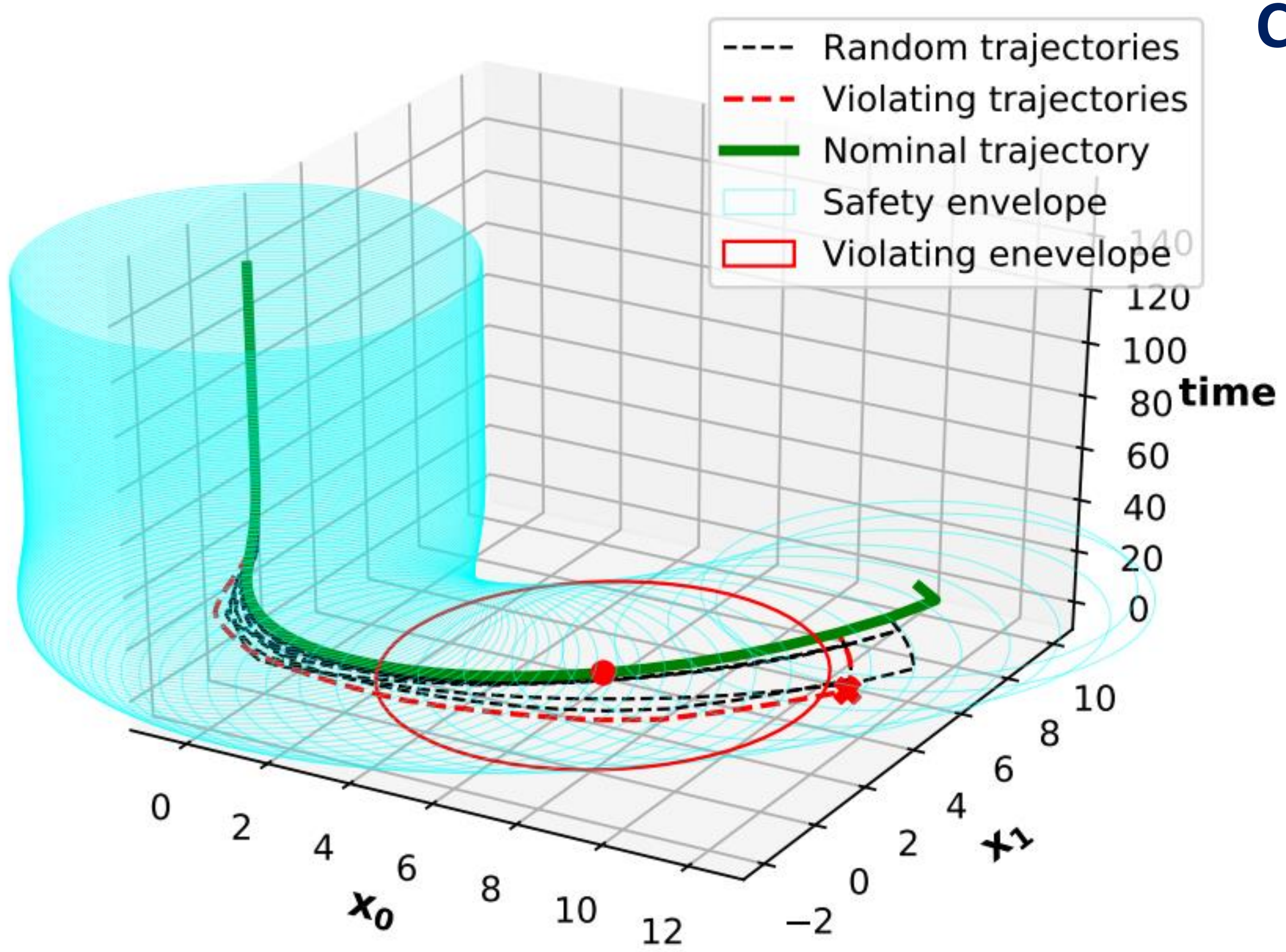
- Our Approach: 3.8
- Benchmark Approach: 12.37

Computation Time

- Our Approach: 1.7 s
- Benchmark Approach: 31 s

Case Study: Electric Steering

Our approach clearly outperforms the benchmark approach!



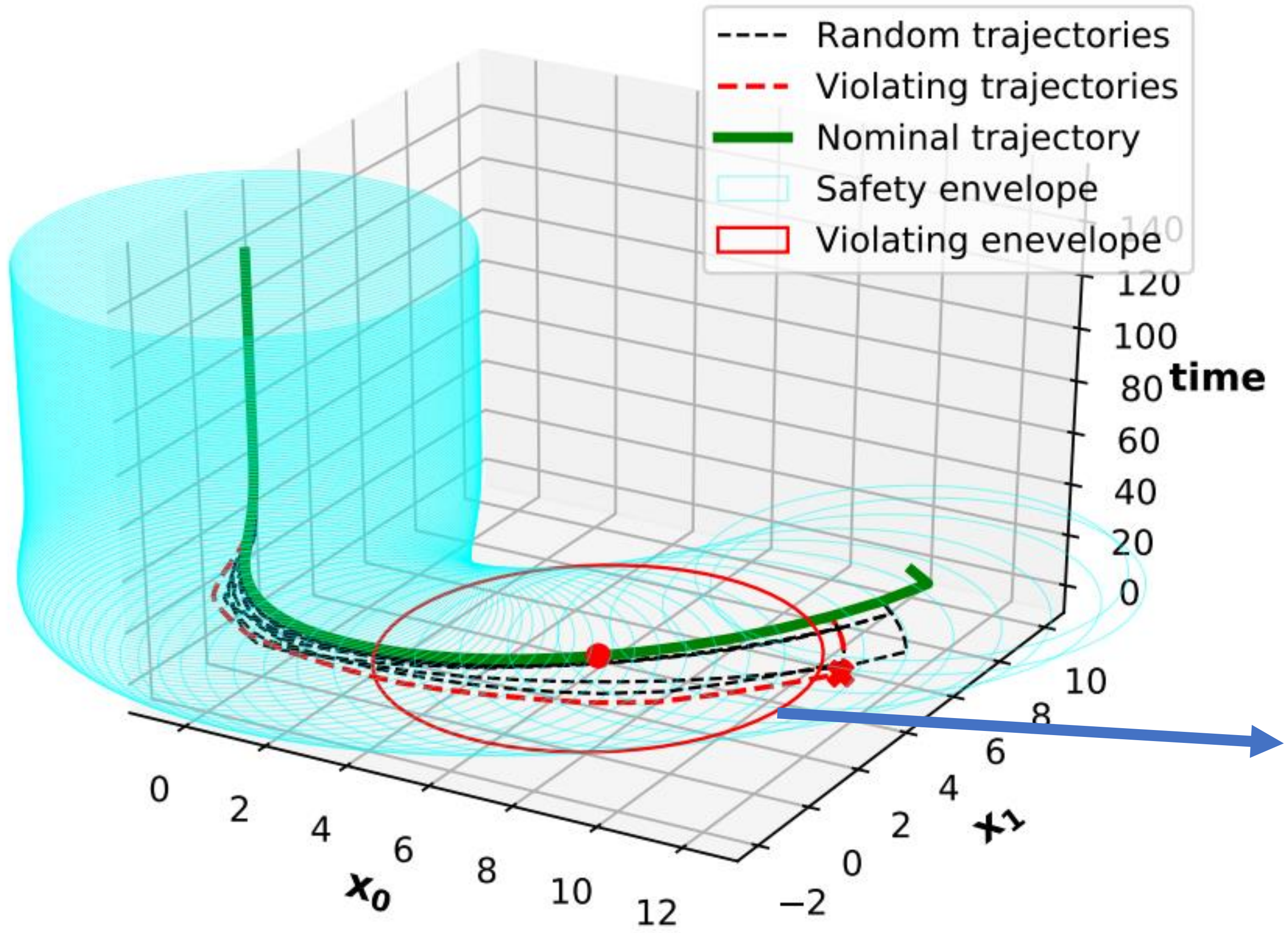
Computed d_{ub}

- **Our Approach: 3.8**
- Benchmark Approach: 12.37

Computation Time

- **Our Approach: 1.7 s**
- Benchmark Approach: 31 s

Case Study: Electric Steering



No unsafe behavior with "at most 3 consecutive deadline misses"!

Unsafe behavior with *"at most **4** consecutive deadline misses"!*

Conclusion

- Statistical approach to compute maximum deviation under deadline misses!
- Our approach computes tighter upper bounds with less computation time.
- **Future Work:** Complicated deadline miss patterns.

Scan this QR code on your phone

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

The open-source prototype tool, **`StatDev`**, is available at:
github.com/bineet-coderep/StatJitteryScheduler

Thank You!