# Statistical Model Checking for SWARMS

Jeffery Hansen (PI)
Sagar Chaki
Scott Hissam
James Edmondson
David Kyle

# Problem Statement

Military is interested in autonomy

- Cooperating unmanned systems
- Uncertain environments
- Adapt to change autonomously

Problem: Need systematic techniques for estimating the probability of mission success.

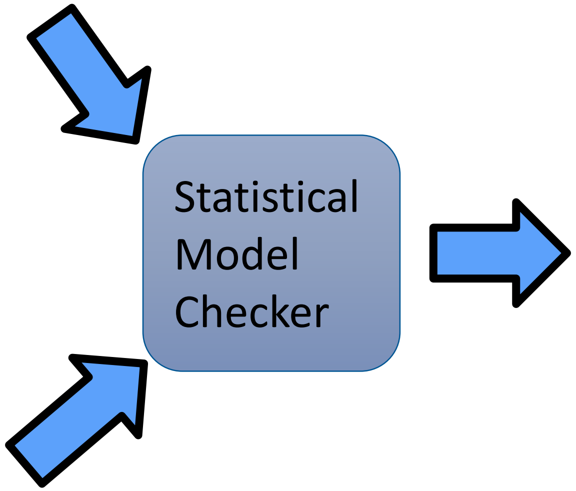
- Systems are large and complex
- Too large for formal models
- Stochastic/uncertain environment

But…. Is a simple estimate of mission success probability good enough?

- Why did you get 0.85 probability of success?
- What factors influence that result?
- What can you do to improve that result?

## Statistical Model Checking

System $\mathcal{M}$ with random inputs (e.g., collection of cooperating UAS performing a mission)




Estimated Probability that $\mathcal{M} \models \boldsymbol{\phi}$ with relative error $\boldsymbol{RE}$

Predicate $\boldsymbol{\phi}$ to be tested (e.g., “mission success”)

$$\boldsymbol{Relative\ Error} = \frac{\boldsymbol{Std.Dev.}}{\boldsymbol{Mean}}$$

# Motivating Example

## Pursuer/Evader Example

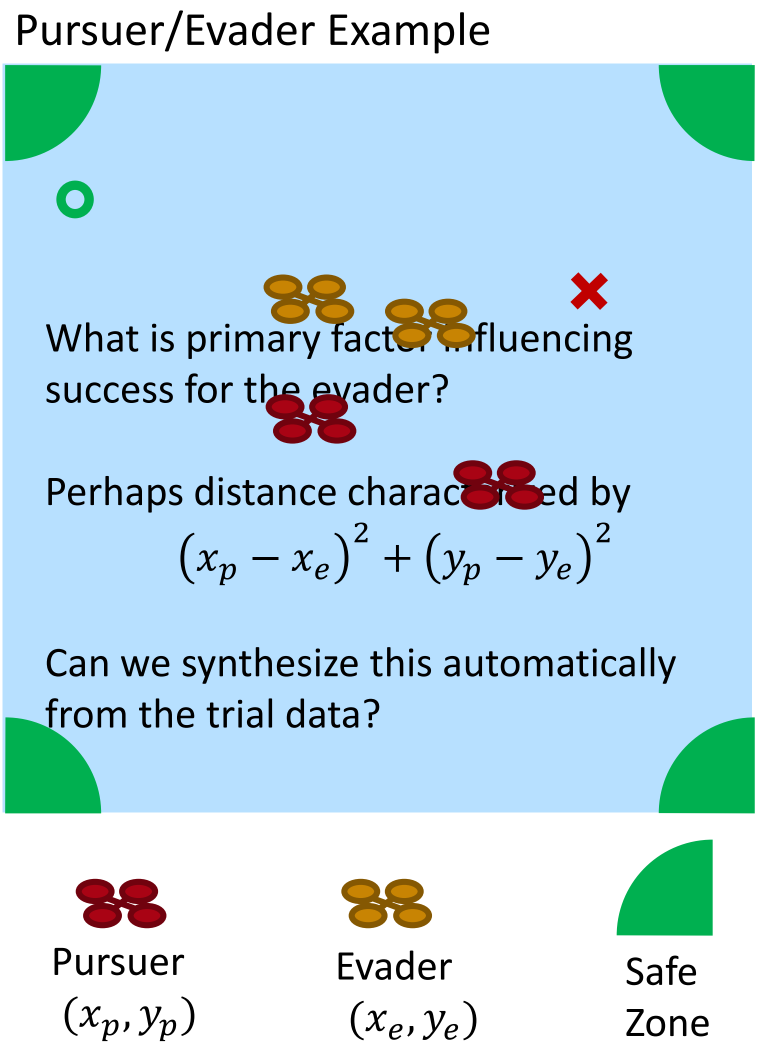
- Random initial positions $(x_p, y_p)$ and $(x_e, y_e)$ near center.
- Evader attempts to reach safe zone in corner.
- Faster moving pursuer attempts to catch evader.

## Statistical Model Checking (SMC)

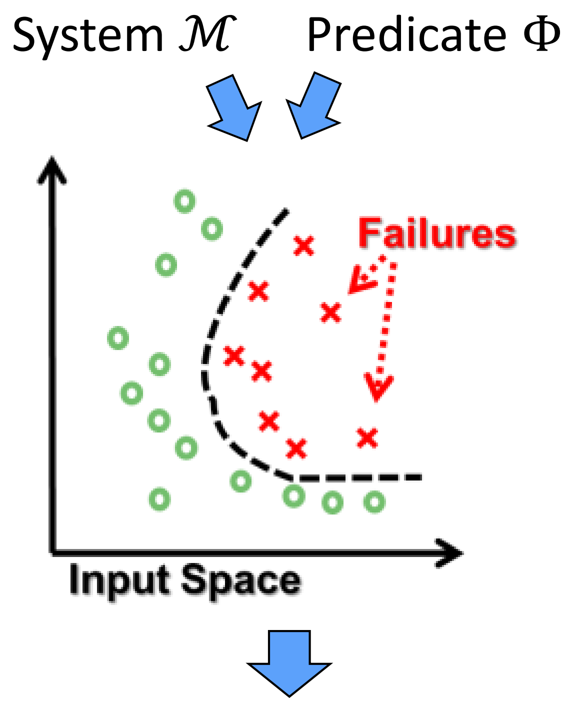
- Let $\mathcal{M}$ be the model for the pursuer/evader scenario and $\Phi$ be the property "the evader reaches safe zone".
- SMC attempts to answer the question, "What is the probability that $\mathcal{M} \vDash \Phi$?"

## Input Attribution (IA)

- Asks the question "Why do I get a particular SMC result?"
- Analog to counter-example in model checking.
- Expressed in terms of the inputs as model approximation.



# Input Attribution – The “Why” of SMC




Input Attribution

| Expression | p-Value |
|---|---|
| $0.62(a - 1.01d)^2$ | 0.0013 |
| $4.3b$ | 0.0042 |
| $1.3(2.3 - c)$^2 | 0.0172 |

Problem – Standard SMC provides an estimate on probability that a predicate is satisfied, but does not address why a particular result was obtained.

Goal – Provide investigator with informative non-redundant representation of how system inputs relate to the property being tested:

1. Describes relationship that actually exists in data
2. Is presented in a way that is quantitative and understandable
3. Gives investigator new insights
4. Is resilient to randomness in the system

Approach – Apply machine learning and feature extraction techniques.

- Use *Logistic Regression* to identify “predictors” that affect the probability that a predicate is satisfied.
- Calculate p-values for predictors to indicate significance.
- Look for sets of predictors that can be factored into larger expressions.

# Evaluating LR Results (Linear Case)

Logistic Regression Model:

$$L(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N)}}$$

$L(x)$ represents predicted probability that input $x$ will satisfy the predicate.

Positive/negative values represent increase/decrease of predicate probability.

- Measure of statistical significance
- Probability that $\beta = 0$
- >0.05 $\rightarrow$ not statistically significant

| Name | $\hat{\beta}$ | $se(\hat{\beta})$ | p-Value |
|---|---|---|---|
| – | -4.28 | 0.874 | $< 10^{-4}$ |
| $x_1$ | 0.154 | 0.0138 | $< 10^{-4}$ |
| $x_2$ | -1.91 | 0.3551 | $< 10^{-4}$ |
| $x_3$ | 0.0635 | 0.0277 | 0.0219 |
| $x_4$ | 5.05 | 2.77 | 0.0685 |

Constant Term

Predictors (input variables)

Error in estimation of $\beta$.

This predictor is not statistically significant since its p-value is greater than 0.05.

# Polynomial Input Attribution

## Non-Linear Predictors

- By including non-linear predictors, it may be possible to find a statistically significant solution when linear only terms fail.
- In our work to date, we have focused on quadratic terms (e.g., $x^2, y^2, xy$)
- Higher order or non-polynomial terms could be useful for some systems.

## Factoring

- Factored polynomials are easier for humans to understand.
- Since coefficients are approximated, perfect factorings may not be possible.
- Look for approximate factorings which do not adversely affect original coefficients.

Find variable pairs with squares and cross terms

| Name | $\hat{\beta}$ | $se(\hat{\beta})$ | p-Value |
|---|---|---|---|
| ⋮ | ⋮ | ⋮ | ⋮ |
| $x^2$ | 1.01 | 0.0148 | $< 10^{-4}$ |
| $xy$ | -2.04 | 0.0362 | $< 10^{-4}$ |
| $y^2$ | 1.02 | 0.0193 | 0.0219 |
| ⋮ | ⋮ | ⋮ | ⋮ |

Complete square to create candidate factoring

$$1.01(x - 1.01y)^2$$

$$1.01x^2 - 2.04xy + 1.03y^2$$

Re-expand and accept approximation if error is within set factor of std. error.

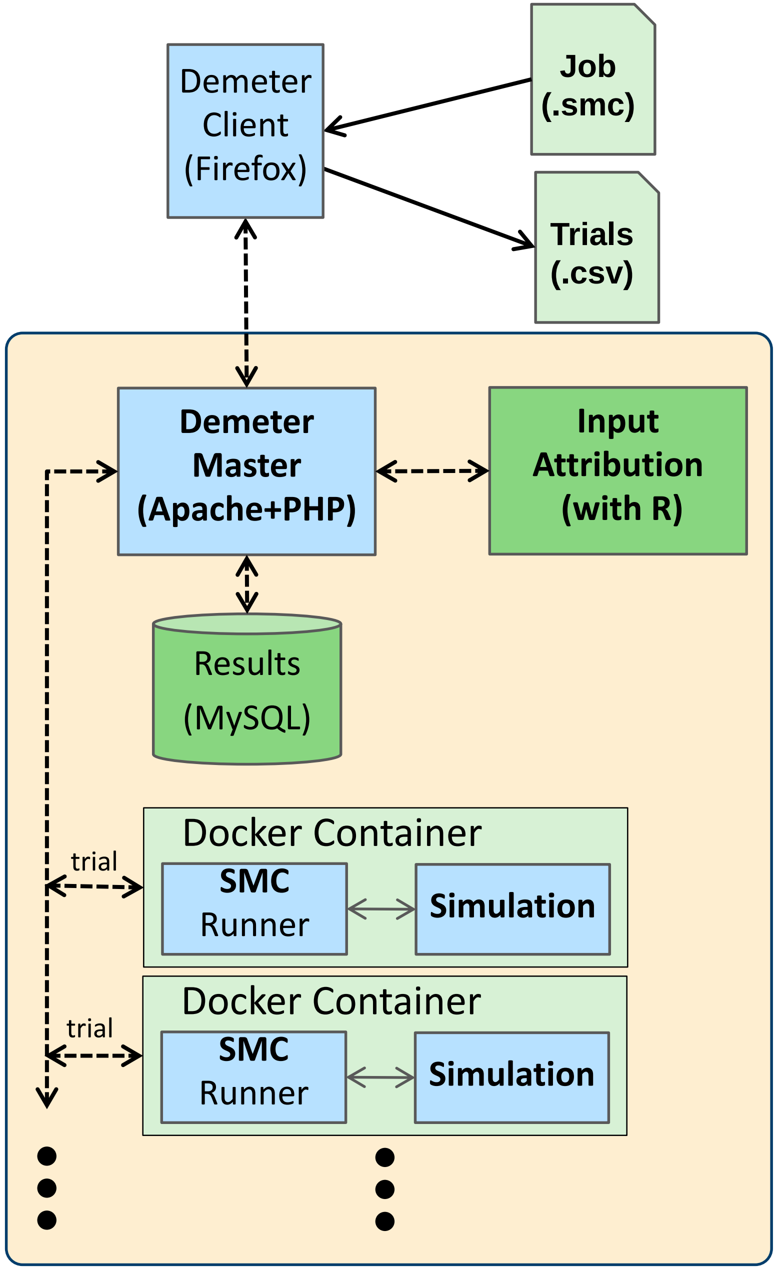
# Implementation – Demeter

## Demeter Goals

- Parallel infrastructure for SMC of systems with probabilistic behaviors.
- Primary target is autonomous systems.
- Integrated Input Attribution

## Demeter Components

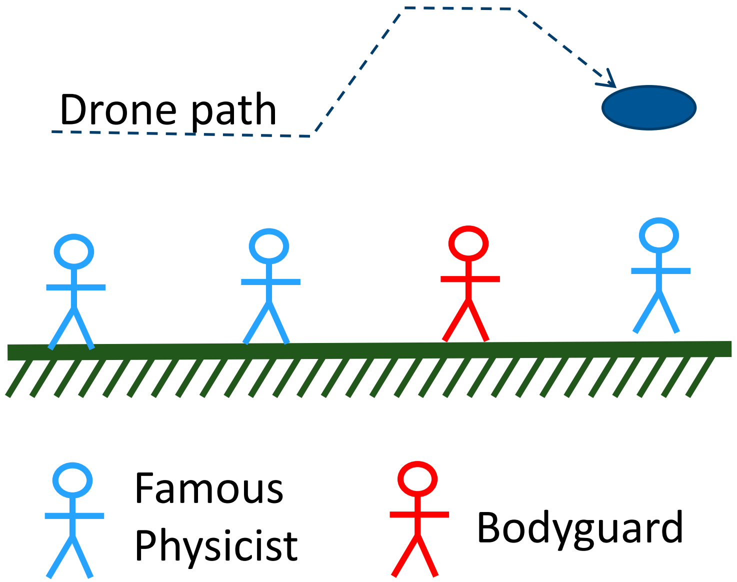
- Client runs in web browser (e.g., Firefox)
- Master runs in Apache server with PHP
- Data stored in MySQL database
- Input Attribution uses R statistical system
- Individual simulations conducted in Docker containers. Managed by “Runner”.



# Target/Threat Experiment

## Scenario

- Drone flies pre-programmed path over area.
- Along path are “targets” to be photographed.
  - Close to ground → Better chance of good photo.
- Path also includes “threats” to be avoided.
  - Close to ground → More likely to be destroyed.
- Adaptive algorithm with imperfect sensors, sense threats ahead and controls altitude.




## Inputs

- Number of targets/threats
- Target detector range (down)
- Target/Threat detector range/accuracy (forward)
- Threat range

## Predicate

- Drone photographs at least 50% of targets while avoiding being destroyed by threats.

# Target/Threat Experiment

Key Observations

- False positives on threats reduce the probability of detecting targets.
  - Reacting to threats that are not there results in drone flying at higher altitude when not necessary and missing some targets.
- Increasing number of targets reduces probability of survival.
  - Drone takes more risks by flying lower to photograph targets.
- False negatives on threat and target detection do not have statistically significant effect on mission, detection or survival probabilities.
  - Verified with additional simulations varying false negative rate. Could indicate problem with adaptation algorithm controlling drone.

**Simulation Results**

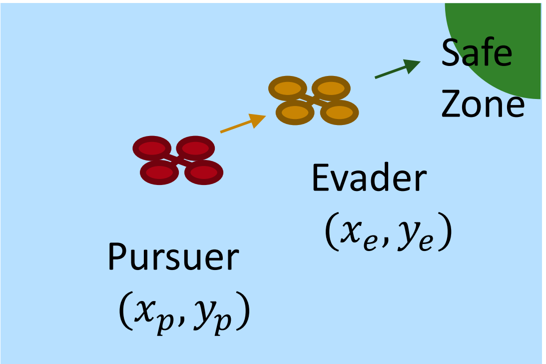
| | |
|---|---|
| #Trials: | 22,560 |
| P[SAT-mission]: | 0.308 |
| P[SAT-survive]: | 0.618 |
| P[SAT-detect]: | 0.361 |
| Relative Error: | 0.05 |
| Batch Size: | 120 |
| Run Time: | 10 hours, 6 min |

**Input Attribution (AUC=0.926)**

| Name | $\hat{\beta}$ mission | $\hat{\beta}$ detect | $\hat{\beta}$ survive |
|---|---|---|---|
| Target Detector Range | 1.33 | 1.46 | |
| Threat Range | -1.57 | -1.189 | -2.37 |
| Threat Lookahead | 0.233 | 0.194 | 0.377 |
| Number of Threats | -0.0892 | -0.0943 | -0.0792 |
| Number of Targets | | | -0.0296 |
| Target False Positives | | | -17.81 |
| Threat False Positives | -3.26 | -10.04 | 32.74 |

# Motivating Example - Revisited

Pursuer/Evader Example




***Initial hypothesis*** – Distance between pursuer and evader was deciding factor for survival of evader.

**Simulation Results**

| | |
|---|---|
| #Trials: | 36,960 |
| # SAT: | 7,900 |
| P[SAT]: | 0.214 |
| Relative Error: | 0.01 |
| Batch Size: | 120 |
| Run Time: | 5 hours, 20 min |

***Results*** – Factoring the IA predictors gives us:

$$0.0602\left(x_e - 1.03x_p\right)^2 + 0.0561\left(y_e - 1.09y_p\right)^2$$

With error less than $4se(\beta)$ on each coefficient.

**Input Attribution (AUC=0.77)**

| Name | $\hat{\beta}$ | $se(\hat{\beta})$ | p-value |
|---|---|---|---|
| $x_e x_p$ | -0.124 | 0.0027 | $< 10^{-4}$ |
| $y_e y_p$ | -0.122 | 0.0027 | $< 10^{-4}$ |
| $x_e^2$ | 0.06 | 0.0031 | $< 10^{-4}$ |
| $y_e^2$ | 0.056 | 0.0031 | $< 10^{-4}$ |
| $x_p^2$ | 0.056 | 0.0031 | $< 10^{-4}$ |
| $y_p^2$ | 0.056 | 0.0031 | $< 10^{-4}$ |

Resulting IA expression very close to square of Euclidean distance. Constant factor represents relation between distance and log odds of survival.

# Summary

## Input Attribution Addresses the “Why” of SMC

- Shows which variables are most important
- Concise human understandable expressions
- Implementation in DEMETER
  - Based on Logistic Regression
  - Extended to Non-Linear Attribution

## Future Work

- Explore other machine learning techniques
- Partitioned/conditional Input Attributions
- Higher order polynomial and non-polynomial predictors