MARGInS
Model-based Analysis of Realizable Goals in Systems

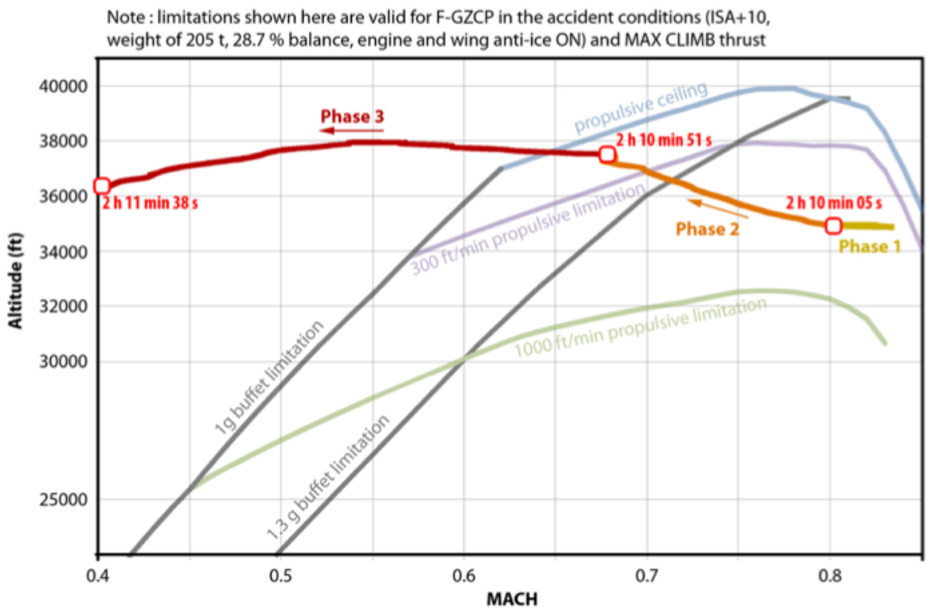
Yuning He
Robust Software Engineering Group

Overview

- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

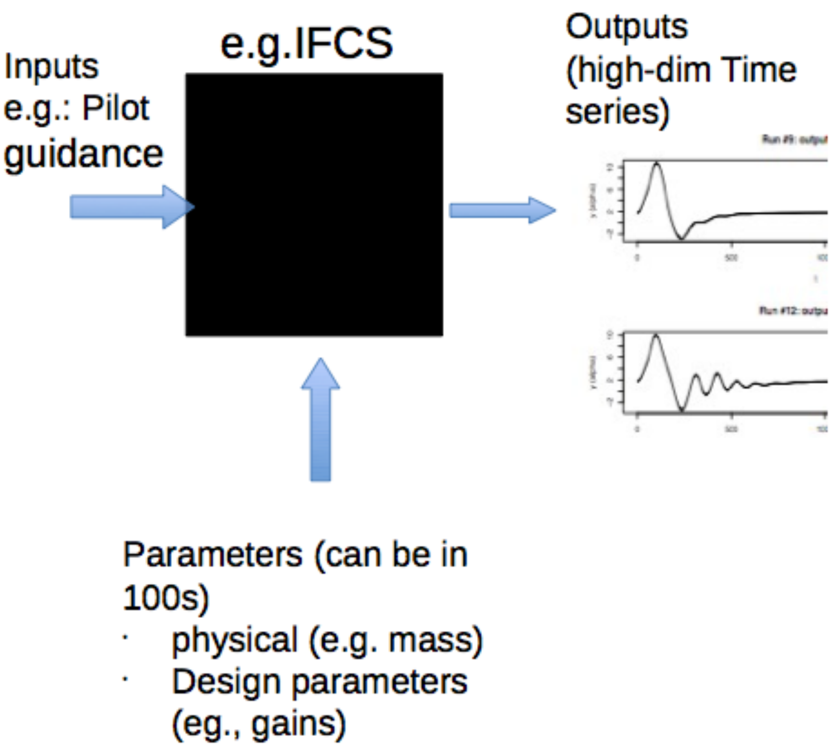
Introduction

- All spacecraft, aircraft and other complex systems can only operate safely within a given operational envelope
- Developers must answer
  - Is the system behaving “well”?
  - Does it stay away from “bad” areas?
  - What are good parameter settings?



Verification and Validation (V&V) is trying to answer these questions

Analysis of a Complex System

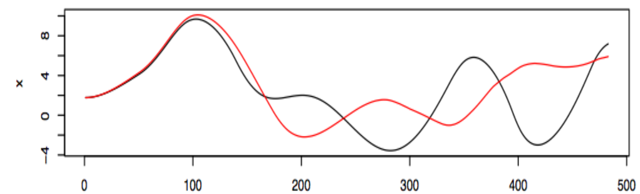


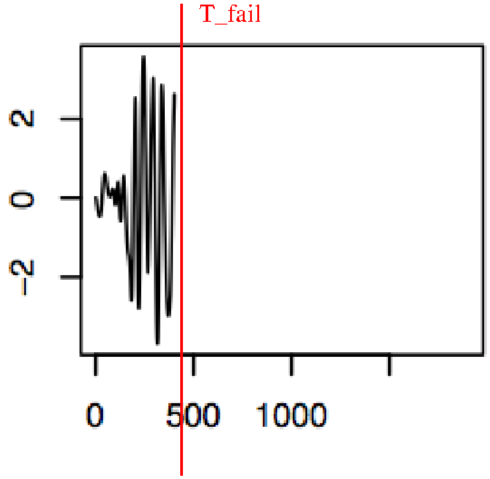
- Safety-critical complex system
- Non-linear, non-trivial software system
- Hybrid: continuous + discrete
- Hardware + Software simulation

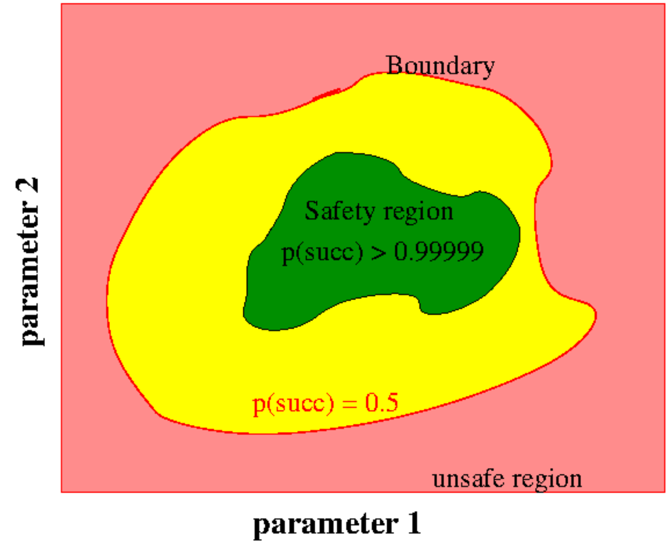
MARGInS uses statistical emulation to quantify uncertainties in models of complex systems

Analysis Tasks for V&V

- In general: unknown mapping from parameters to outputs
- Tasks: learn and build models for
  - Prediction of the whole function of time series
  - Prediction of events, e.g., time to failure
  - Detection and characterization of safety regions and boundaries
- Important for design, analysis, and V&V





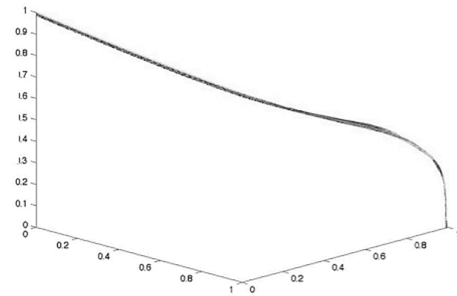


MARGInS helps to perform these analysis and V&V tasks

Overview

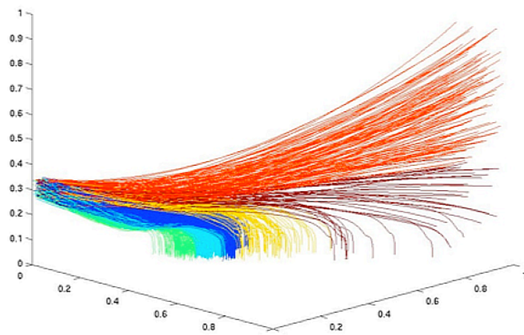
- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

PA-1: Test of Orion Launch Abort System



Traditional Testing

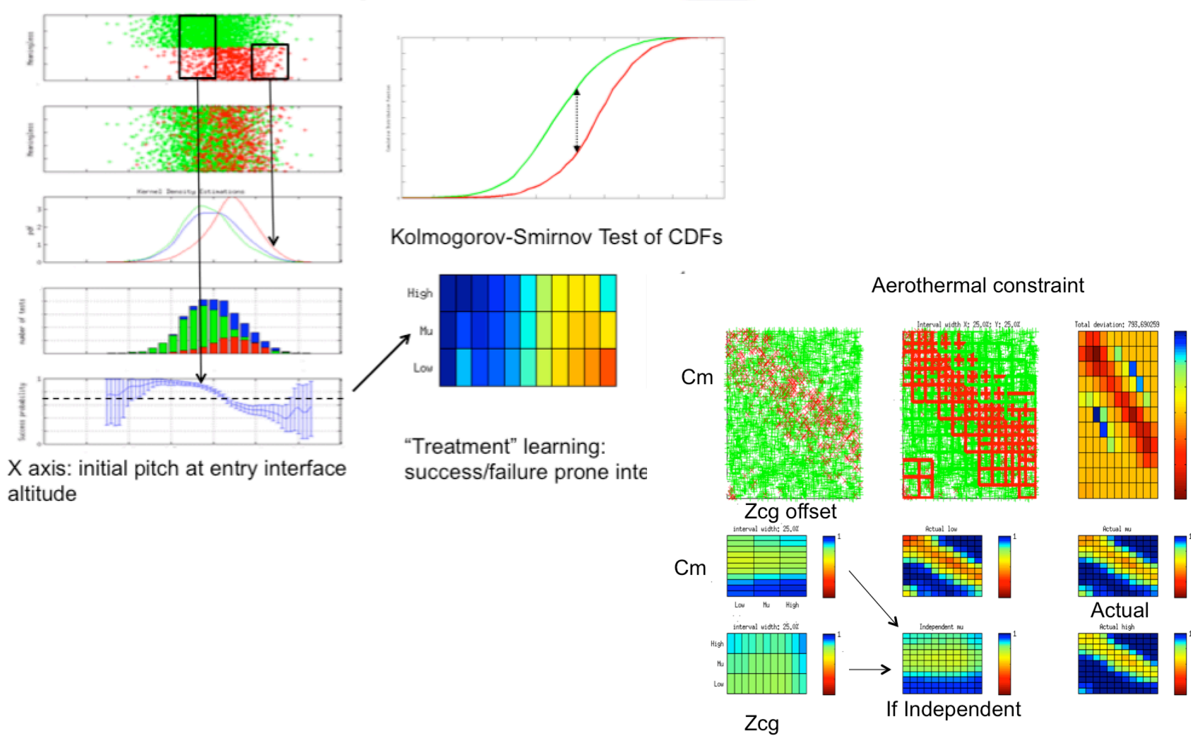
- Low number of tests
- No automatic analysis



MARGInS

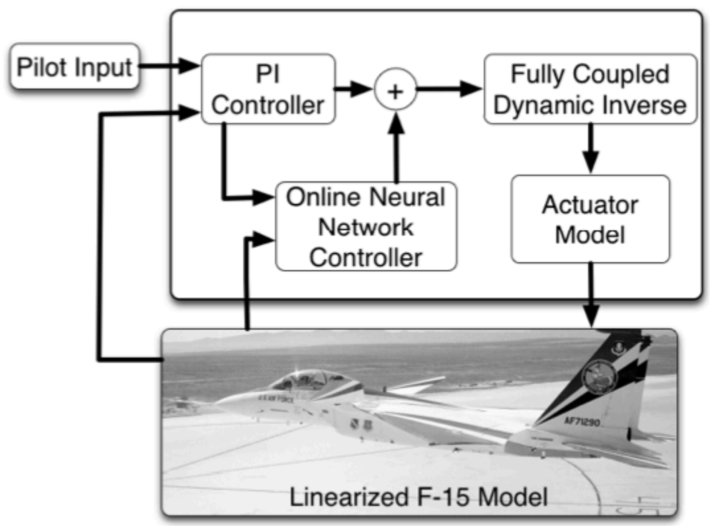
- Exploration of parameter space – many test cases
- Automatic analysis
- Identification of risk classes

Orion EFT-1 – Critical Factor Tool

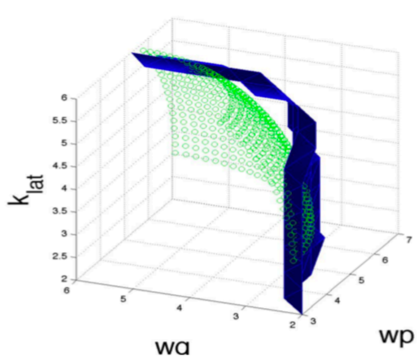


- Identifies critical factors for different objectives and goals
- Generate visualization for domain expert
- Generate documentation and tables

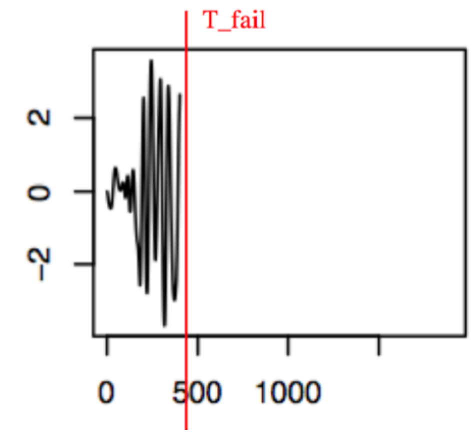
IFCS – Time Series Prediction and Safety Boundary Analysis

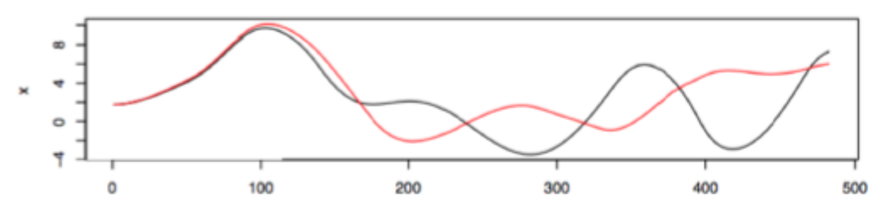


NASA Intelligent Flight Control System



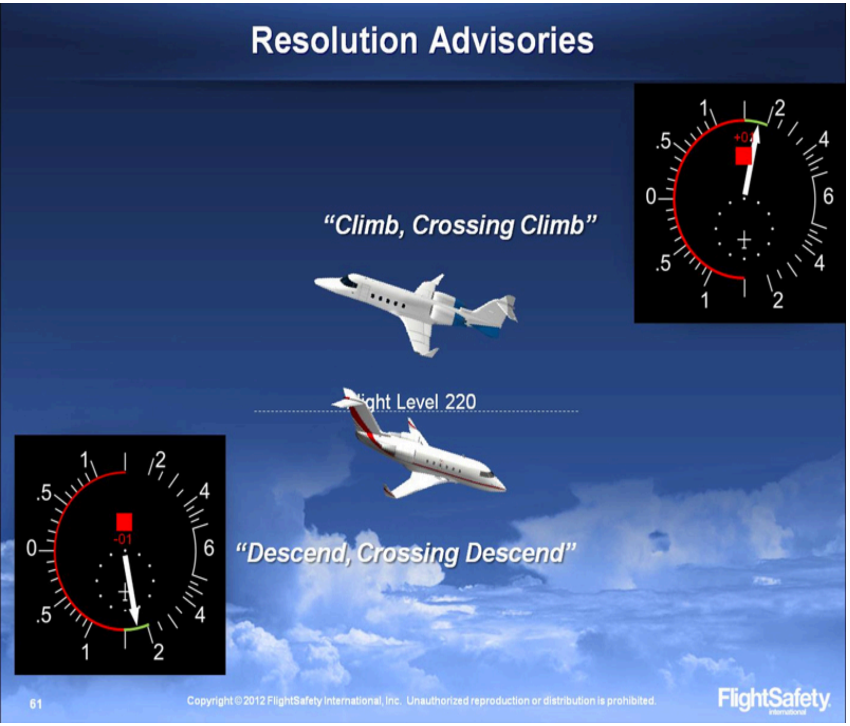
- Damaged AC with adaptive control
  - Will it become unstable? *When?*
  - Predict the trajectory

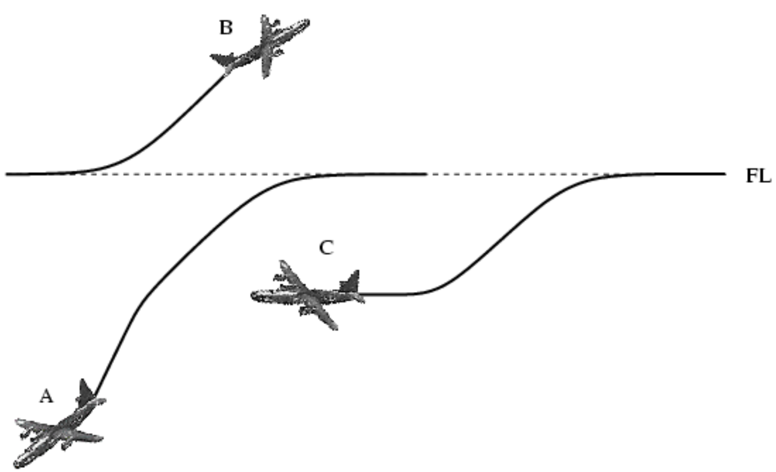




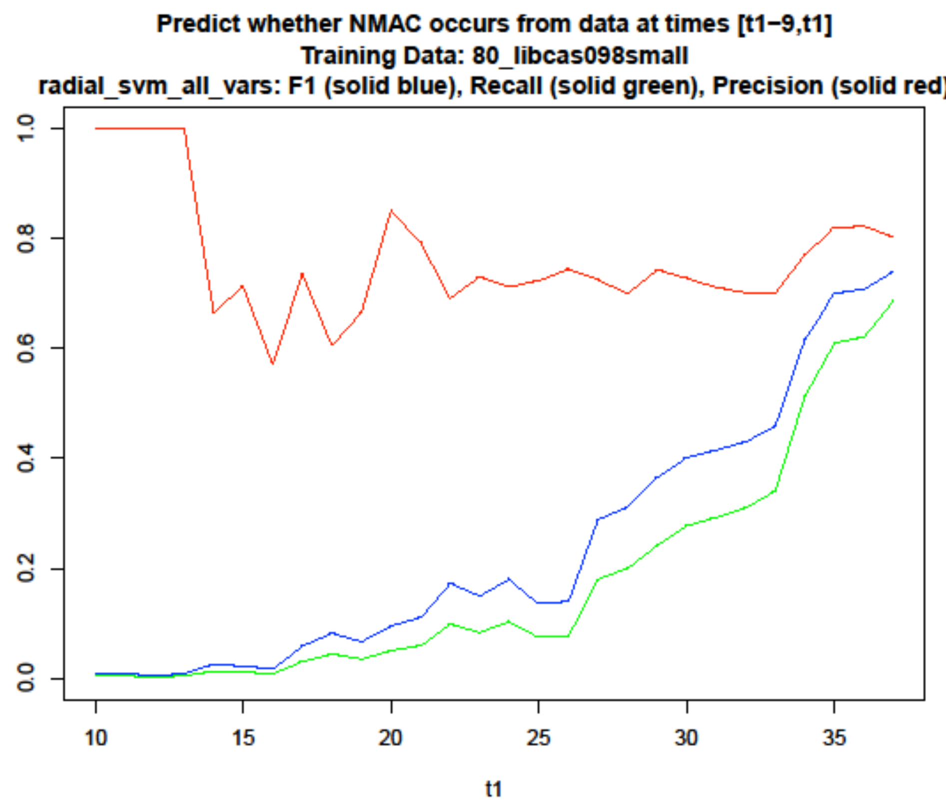
- High dimensional, variable length time series
- Failure events

ACAS X – Prediction of time to NMAC



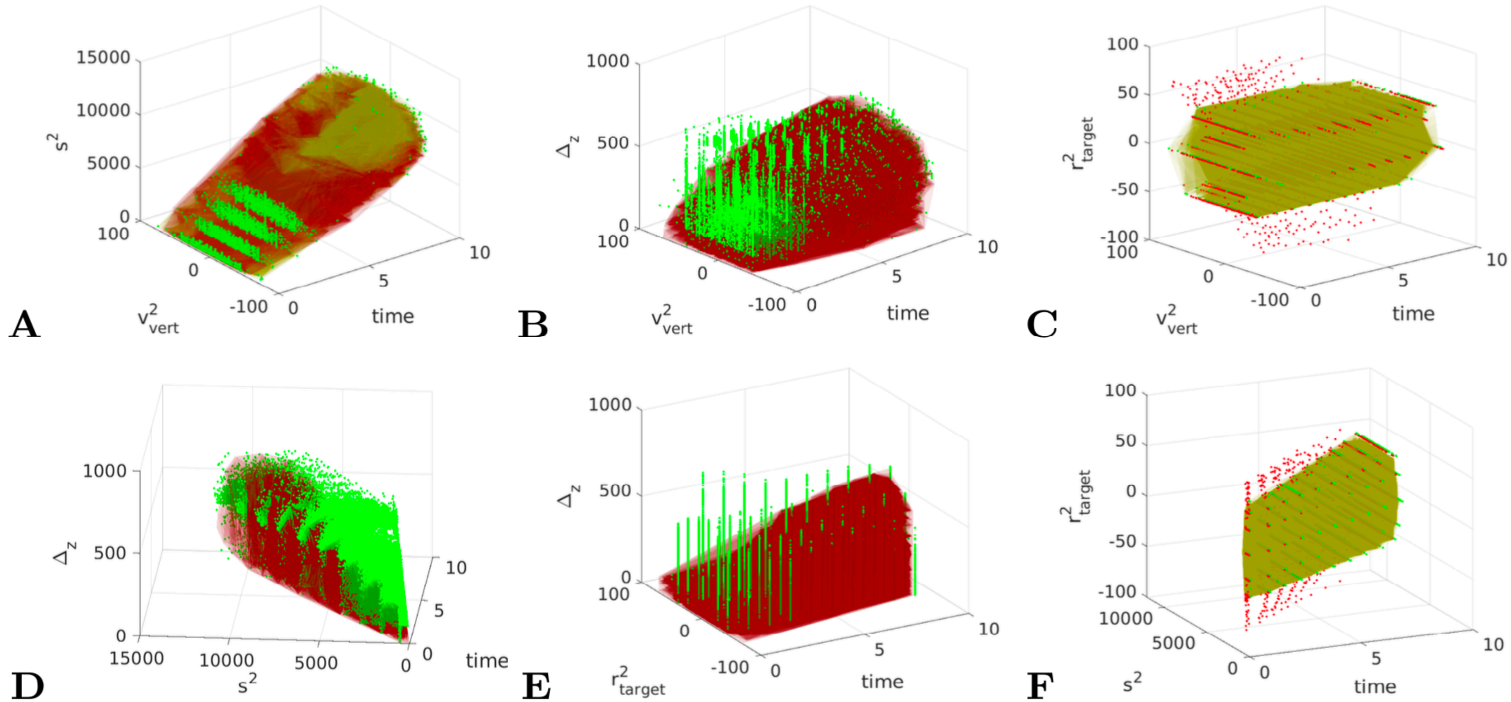


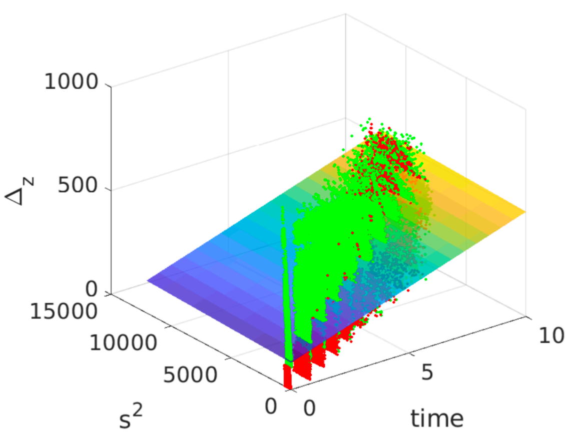
NMAC: Near Mid-Air Collision



- High dimensional time series

ACAS X – Safety Boundaries





Projections of safety boundaries and estimated geometric shapes over different variables

Closed form representation of shapes

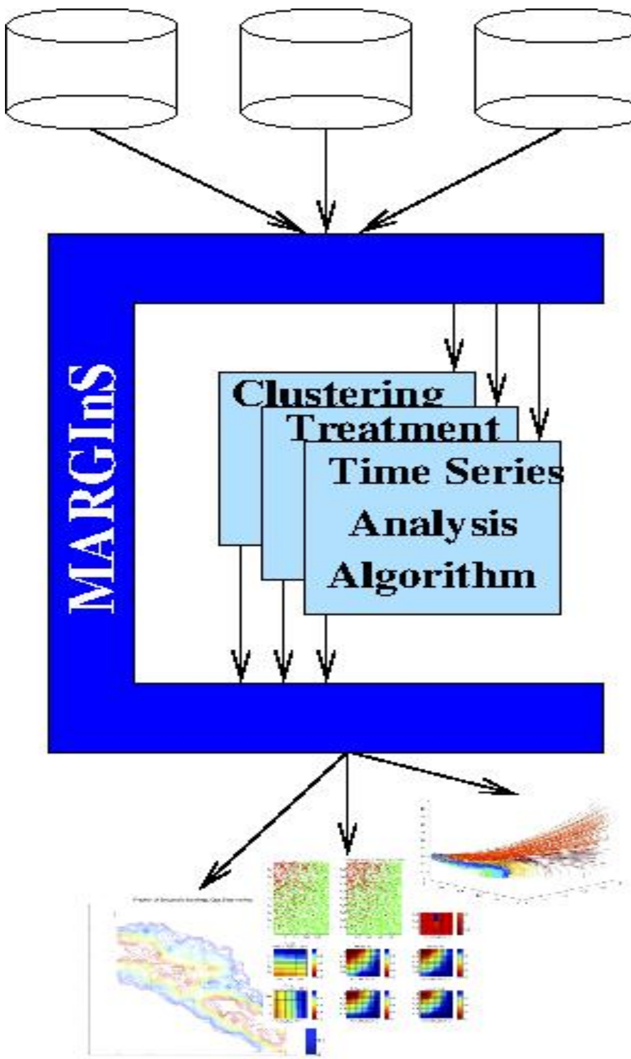
$$0 = 37.68 \text{ time} - \Delta_z + 0.0024 s^2 + 128.7$$

Overview

- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

MARGInS

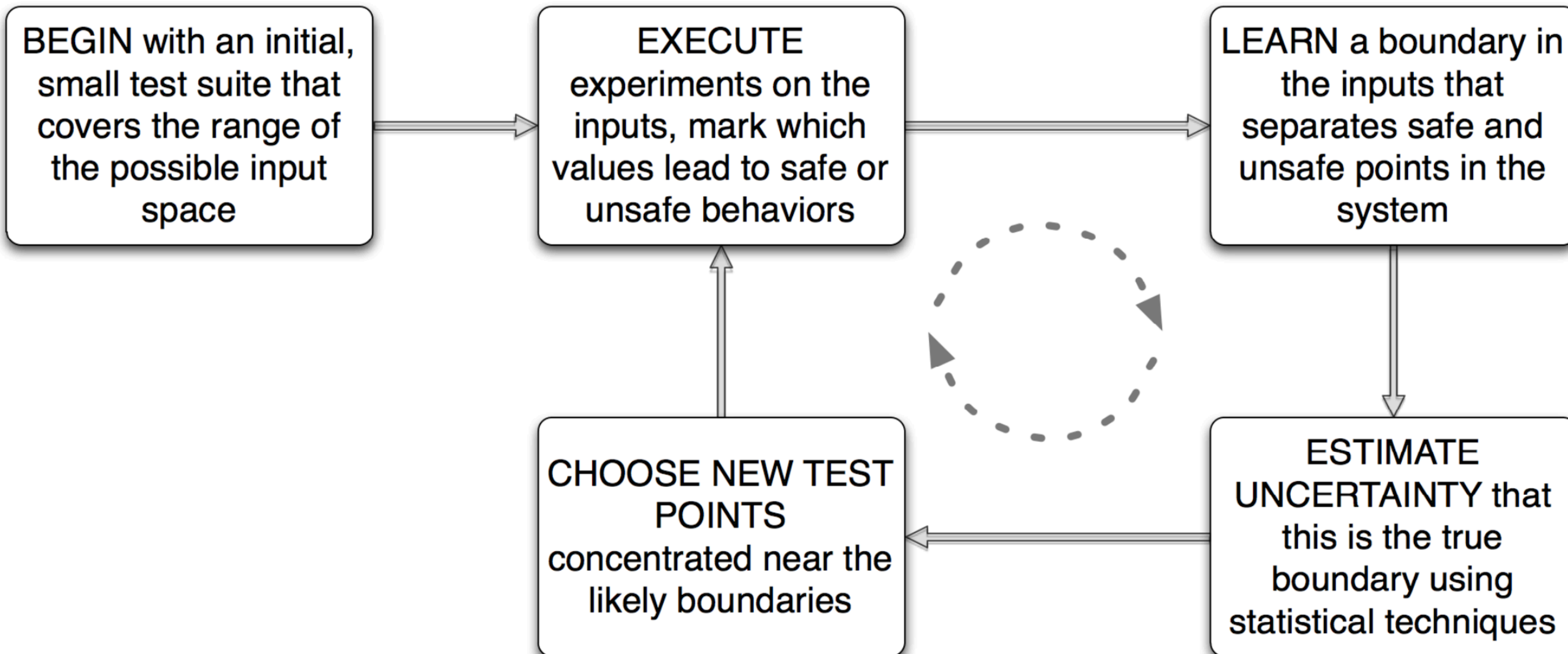
Model-based Analysis of Realizable Goals in System



- Framework and tool set for generating test cases for V&V of complex systems
- Algorithms for
  - Testcase generation
  - Clustering
  - Treatment Learning
  - Critical Factors
  - **Property Checking**
  - **Safety Boundary detection/characterization**

MARGInS is mplemented in Matlab, C/C++, and R

Algorithm Overview



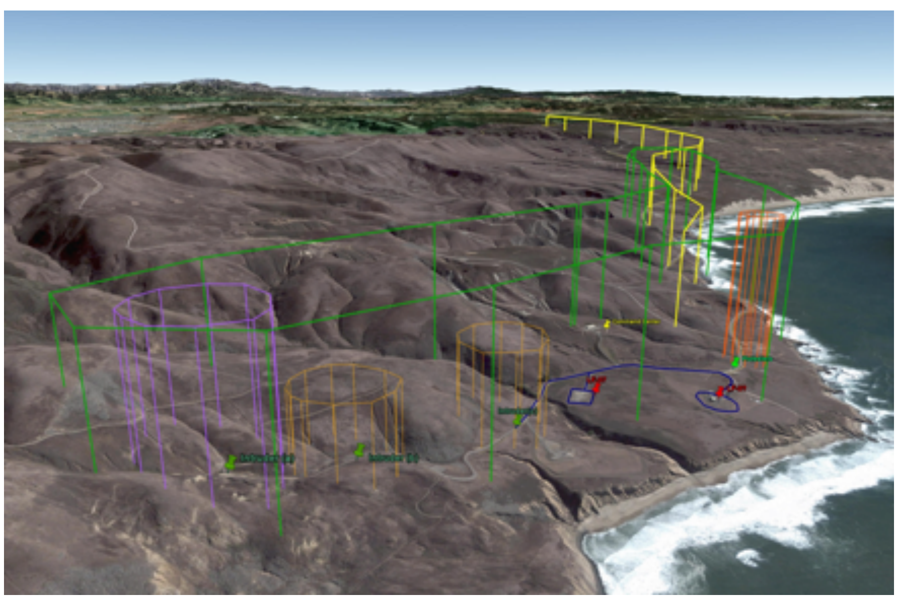
How to use MARGInS

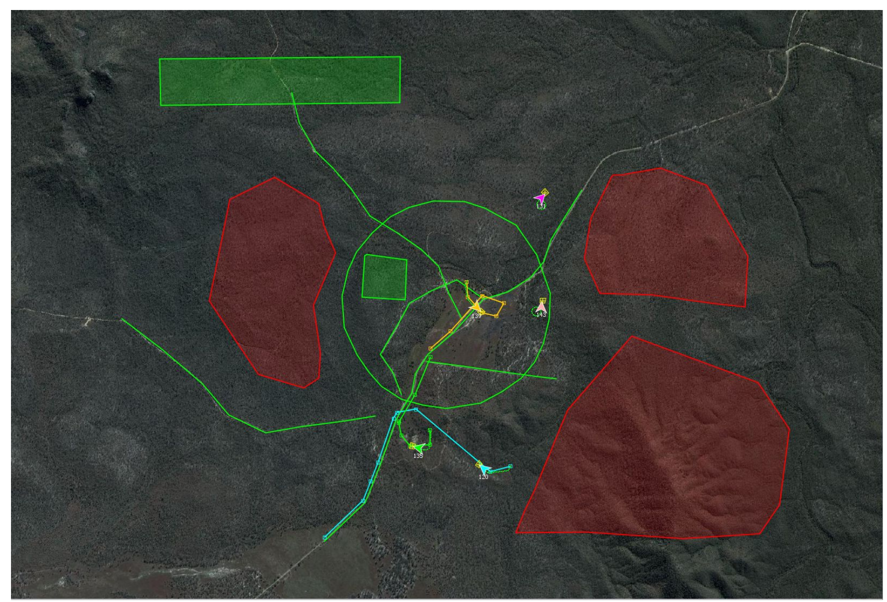
MARGInS is given:

- System under Test. System is implemented in Matlab, Simulink, Java, C,… or a combination
- Analysis tasks and safety properties
- Scenarios of interest
- System information and variables

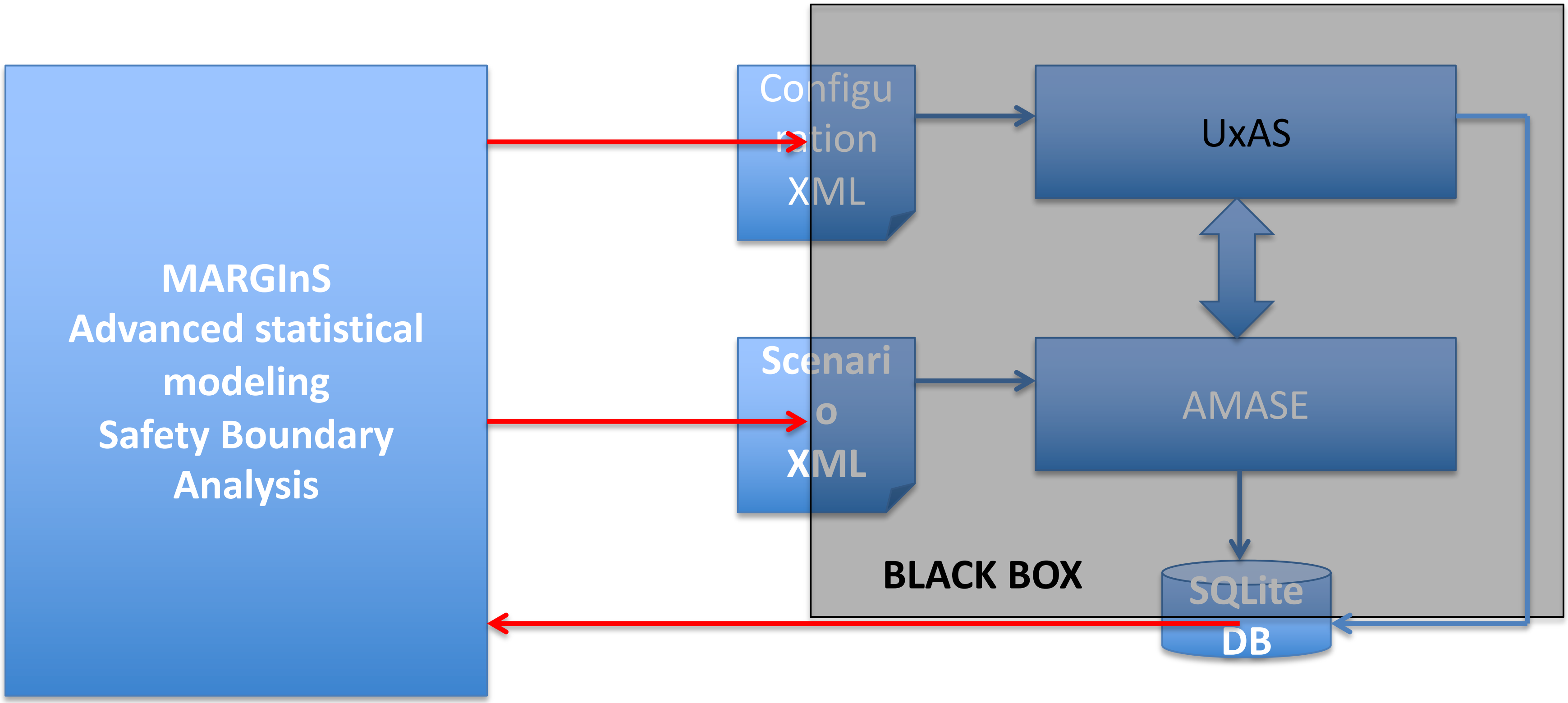
How to connect MARGInS
Application Example: UxAS

- UxAS: Unmanned Systems Autonomy Services
- Net-centric system to automate mission-level decision making for multiple UASs
  - Task assignment
  - Cooperative control
- UxAS system with simulator (AMASE)





How to connect MARGInS



- Automatic variation of selected variables and parameters
- Automatic checking of properties

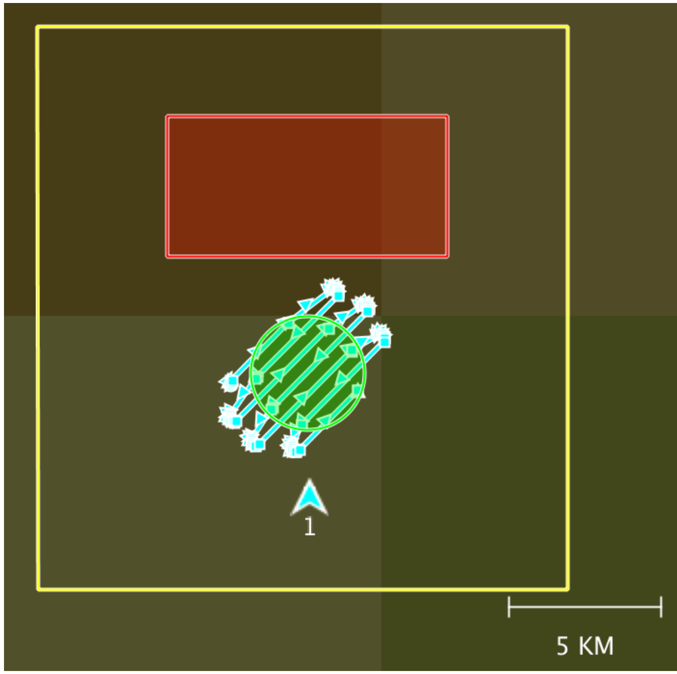
Property Checking with MARGInS

1. Formalize properties to return SAFE/UNSAFE

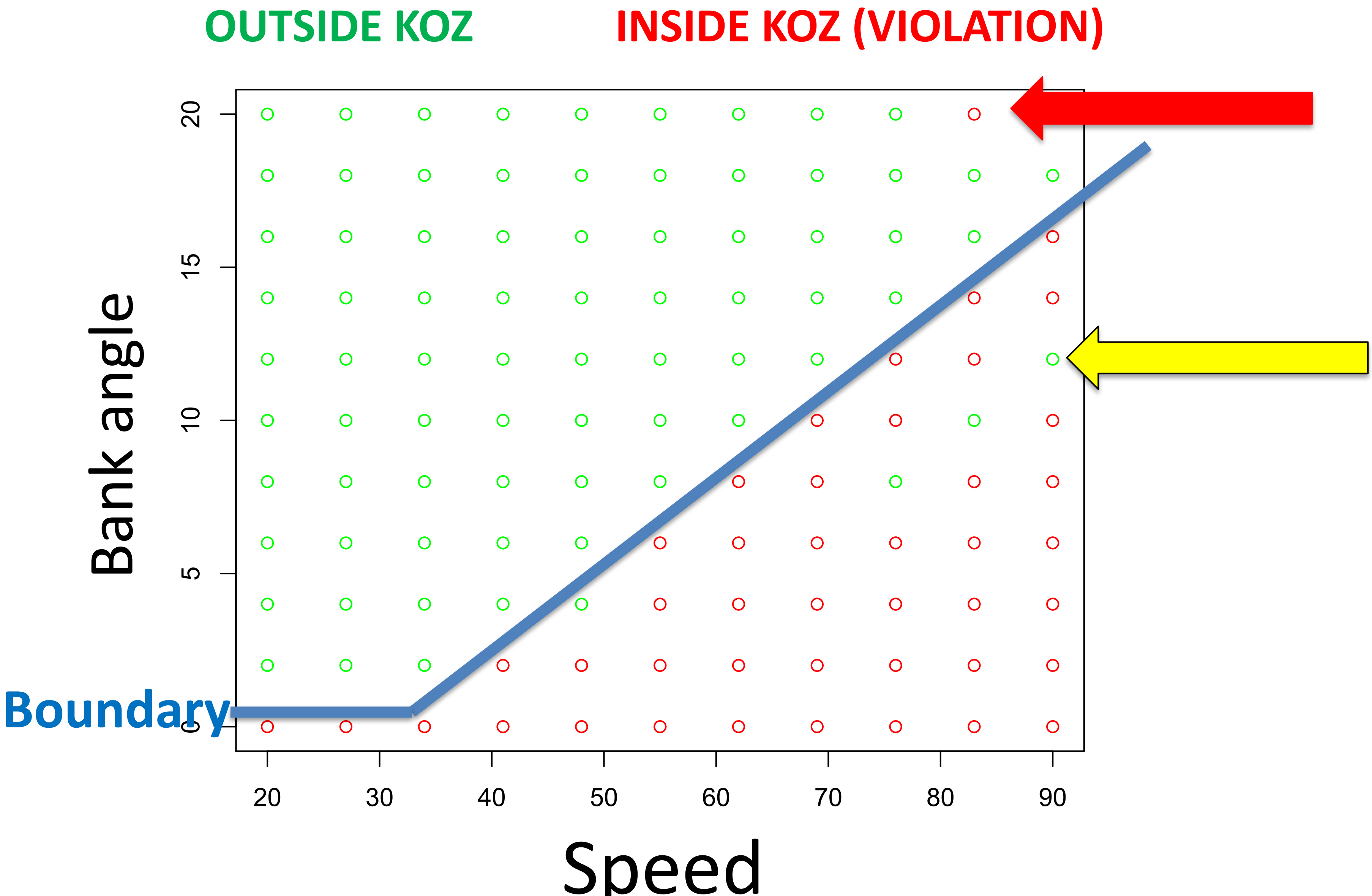
   Example:

   “never enter the KeepOutZone”

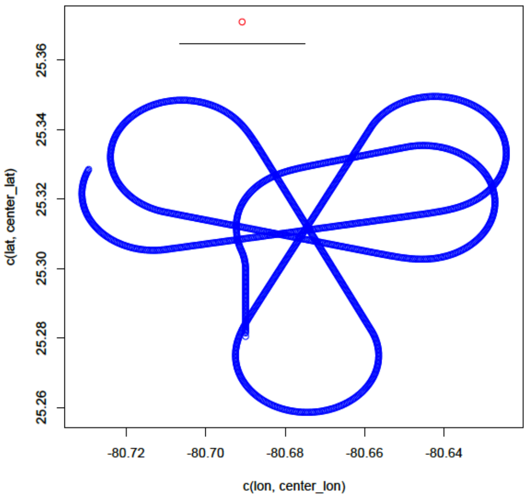
2. Select relevant variables
3. Run MarginS
4. Visualization of results



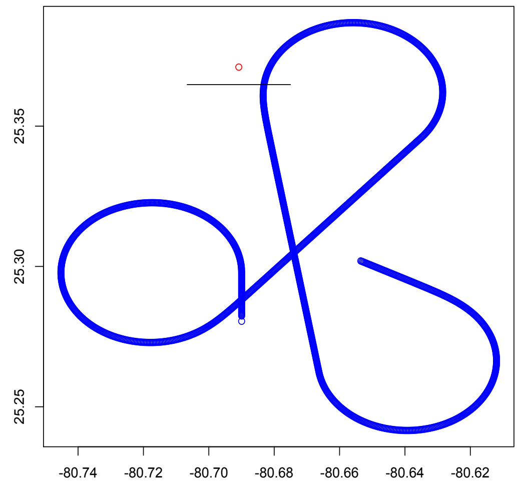
Results


OUTSIDE KOZ
INSIDE KOZ (VIOLATION)
Bank angle
Speed
Boundary
0
5
10
15
20
20
30
40
50
60
70
80
90

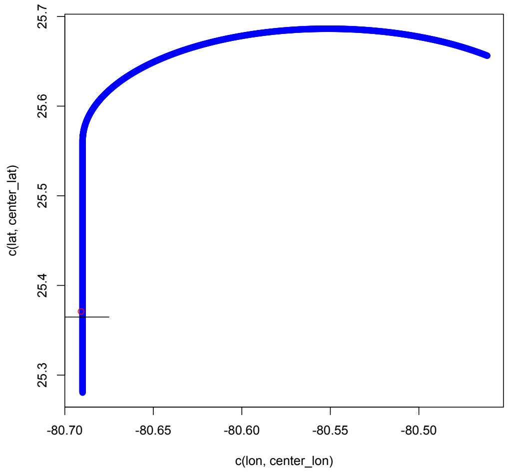
Results: High-Speed situations



SAFE



VIOLATION



VIOLATION

Application Example: Deep Neural Networks in Aerospace

- Deep Neural Networks (DNNs) have become very popular in many areas
- DNN are increasingly used in the Aerospace domain for *mission- and safety-critical* applications
- Verification and Validation (V&V) is extremely important
- Traditional software testing is not suitable for DNN
- MarginS supports effective testcase generation for DNNs in Aerospace systems

Our Application: physics-based DR-RNN

- Given: physics-based Deep Residual Recurrent Neural Network
- Modeling the aircraft dynamics
- For 747-100 aircraft

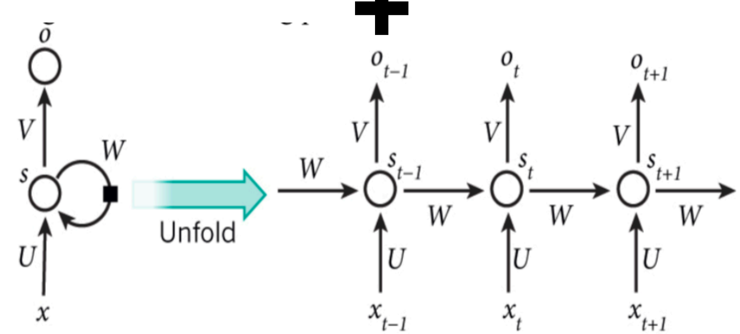
- Is the DR-RNN a suitable approximation for the real aircraft dynamics?
- Is the deviation between the DR-RNN and the real system acceptable?

Our Application: physics-based DR-RNN for 747-100 Aircraft

Deep Residual Recurrent Neural Networks (DR-RNN) for modeling of aircraft dynamics

$$
\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\varphi} \end{bmatrix}
=
\underbrace{\begin{bmatrix}
\frac{Y_v}{m} & \frac{Y_p}{m} & (\frac{Y_r}{m} - u_0) & g\cos(\theta_0) & 0 \\
\frac{L_v}{I'_x} + I'_{xz}N_v & \frac{L_p}{I'_x} + I'_{xz}N_p & \frac{L_r}{I'_x} + I'_{xz}N_r & 0 & 0 \\
I'_{xz}L_v + \frac{N_v}{I'_z} & I'_{xz}L_p + \frac{N_p}{I'_z} & I'_{xz}L_r + \frac{N_r}{I'_z} & 0 & 0 \\
0 & 1 & \tan(\theta_0) & 0 & 0 \\
0 & 0 & \sec(\theta_0) & 0 & 0
\end{bmatrix}}_{\mathbf{A}}
\begin{bmatrix} v \\ p \\ r \\ \phi \\ \varphi \end{bmatrix}
+
\underbrace{\begin{bmatrix}
\frac{Y_{\delta_a}}{m} & \frac{Y_{\delta_r}}{m} \\
\frac{L_{\delta_a}}{I'_x} + I'_{xz}N_{\delta_a} & \frac{L_{\delta_r}}{I'_x} + I'_{xz}N_{\delta_r} \\
I'_{xz}L_{\delta_a} + \frac{N_{\delta_a}}{I'_z} & I'_{xz}L_{\delta_r} + \frac{N_{\delta_r}}{I'_z} \\
0 & 0 \\
0 & 0
\end{bmatrix}}_{\mathbf{B}}
\begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}
$$

+



- System dynamics given as differential equations

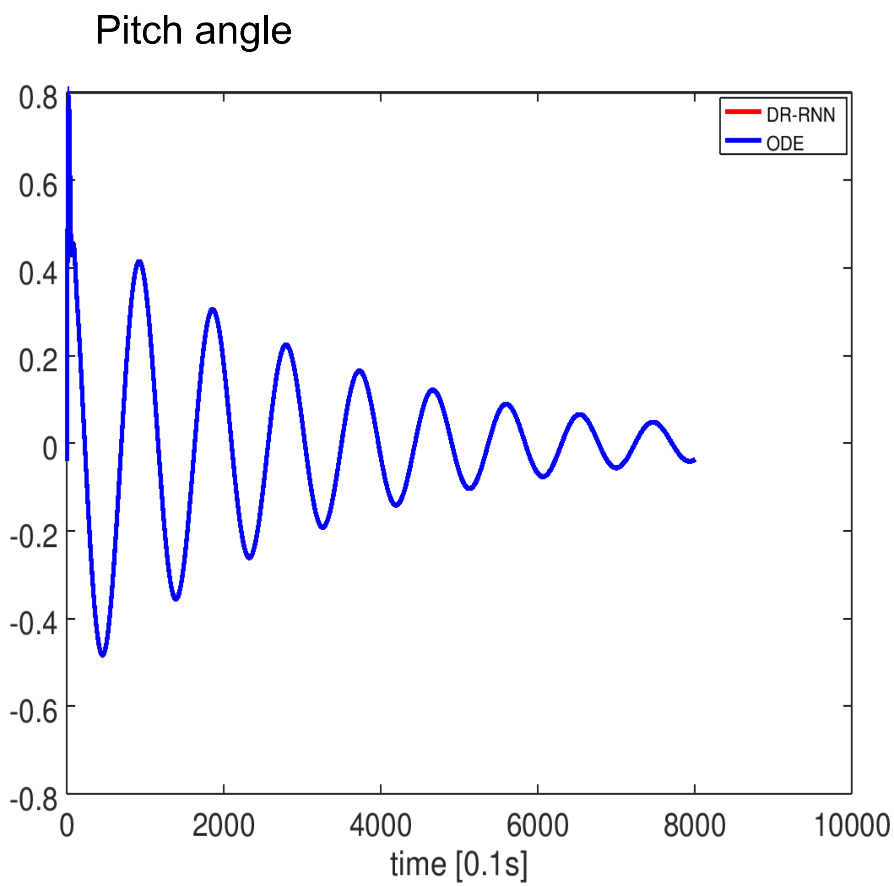
$$\dot{y} = \mathbf{A}y + \mathbf{B}u$$

- Deep recurrent network with *k* layers to learn the residuals

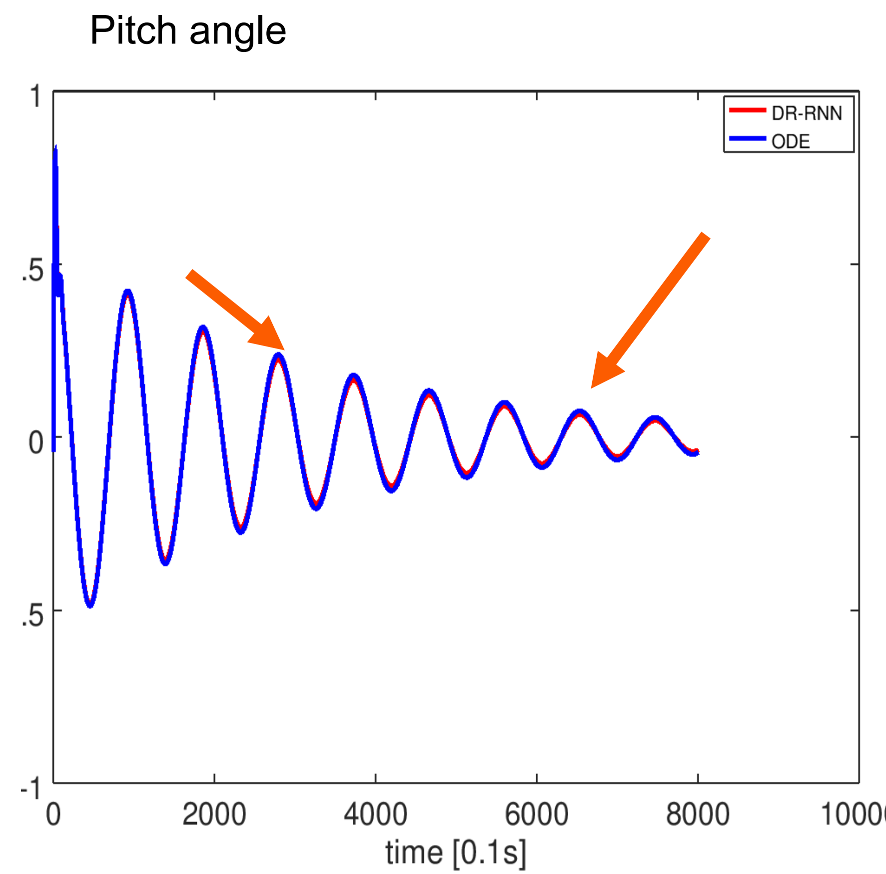
$$r_{t+1} = y_{t+1} - y_t - \Delta_t(\mathbf{A}y_{t+1} + \mathbf{B}u_{t+1})$$

[Yu,Yao,Liu, PHM 2018] DR-RNN for 747-100 aircraft 6DoF dynamics model

Dynamics Deviations



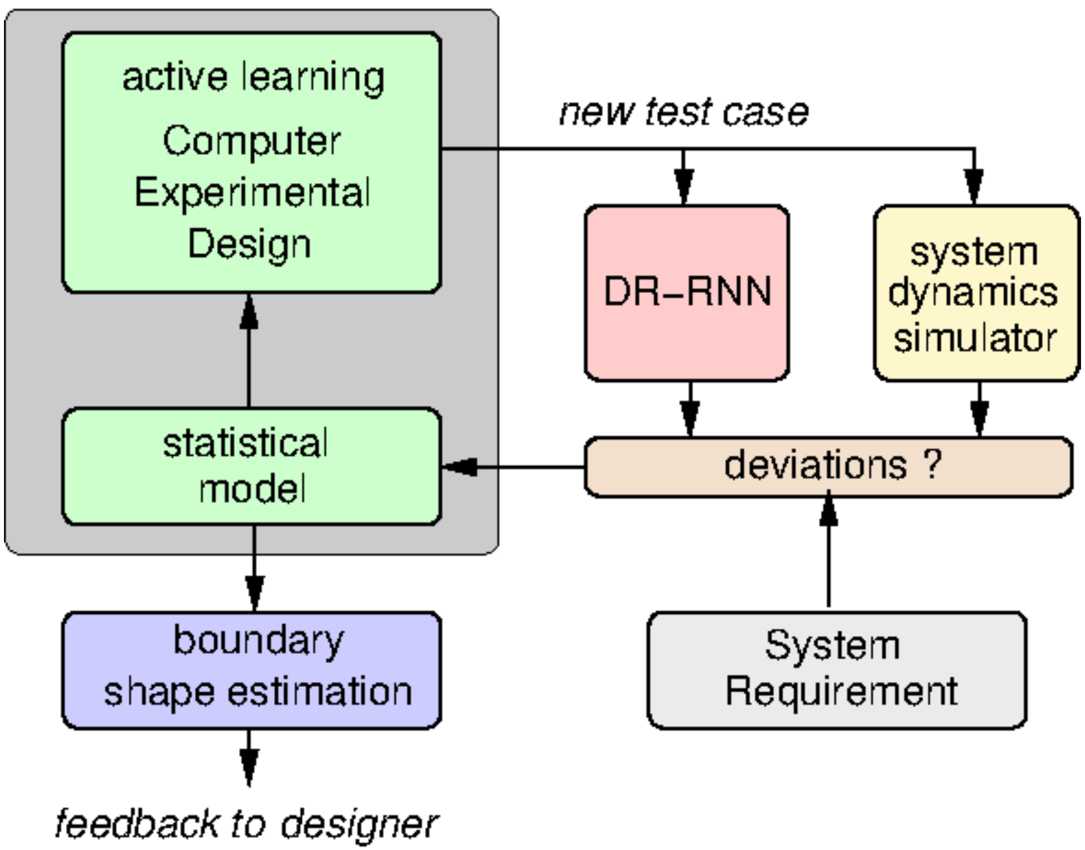
Requirement met



Requirement not met

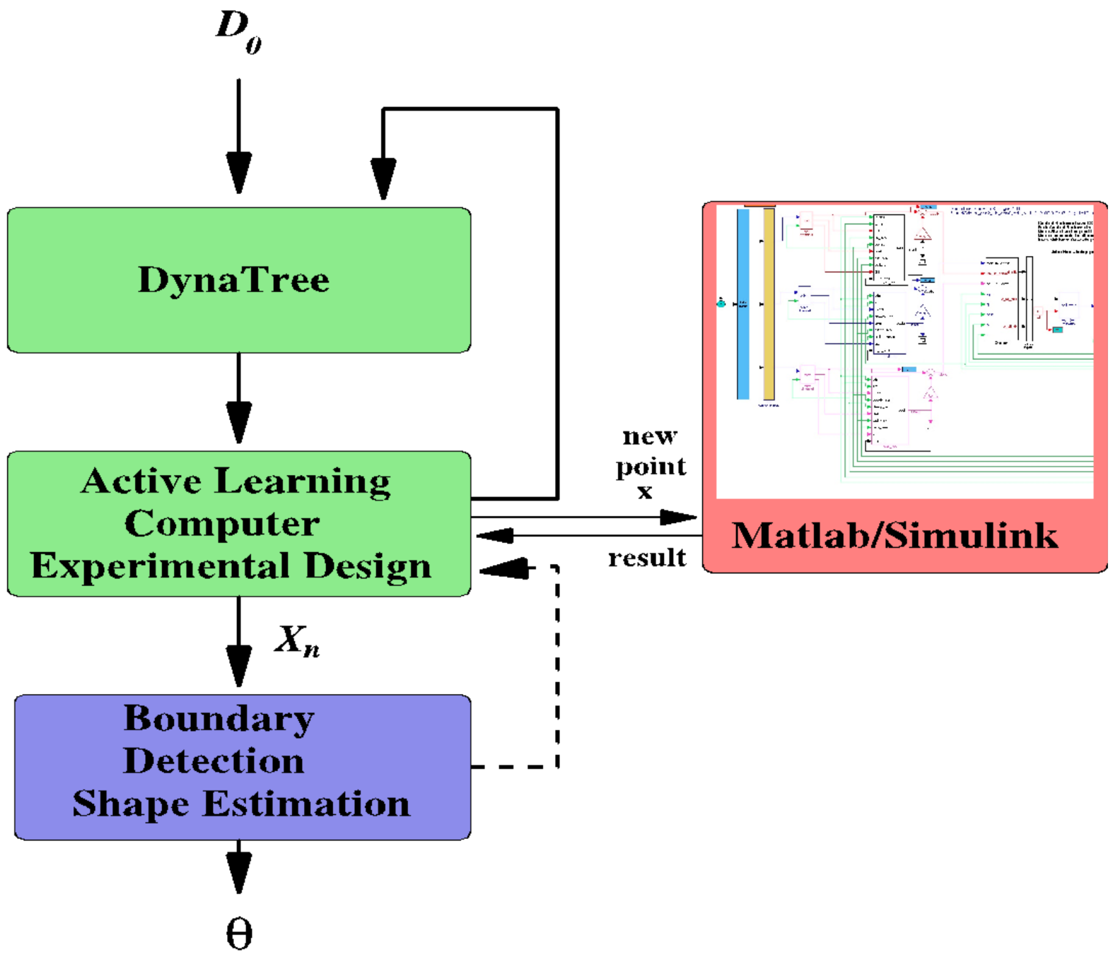
MARGInS Testing Framework

Hierarchical Bayesian statistical modeling with Active Learning in Computer Experiment Design

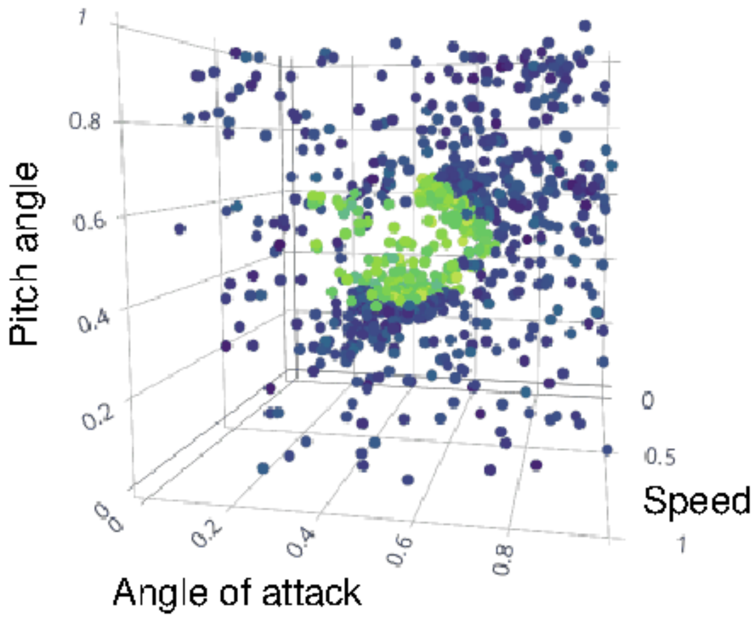


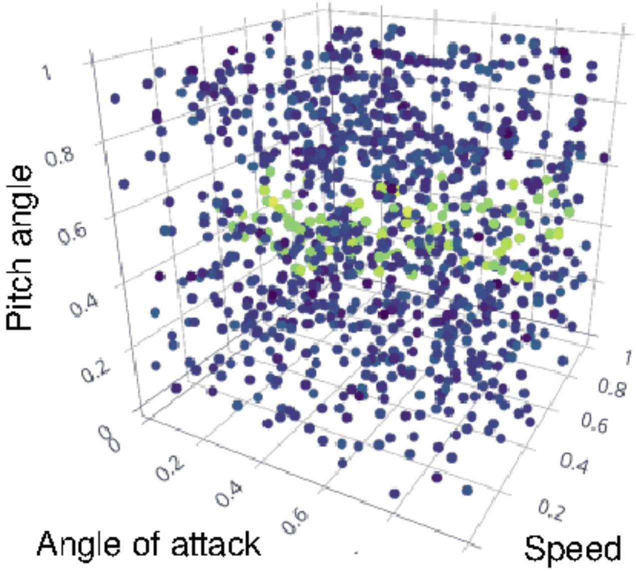
- Generate test cases to find regions of deviation between the DR-RNN and the ground truth (obtained by high-fidelity simulator)
- Threshold given in system requirements
- Active learning selects new test cases close to the estimated boundaries for higher efficiency

Algorithm Overview



Results

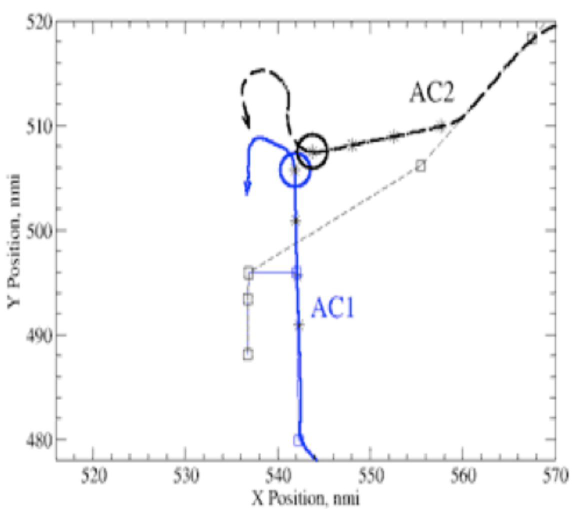


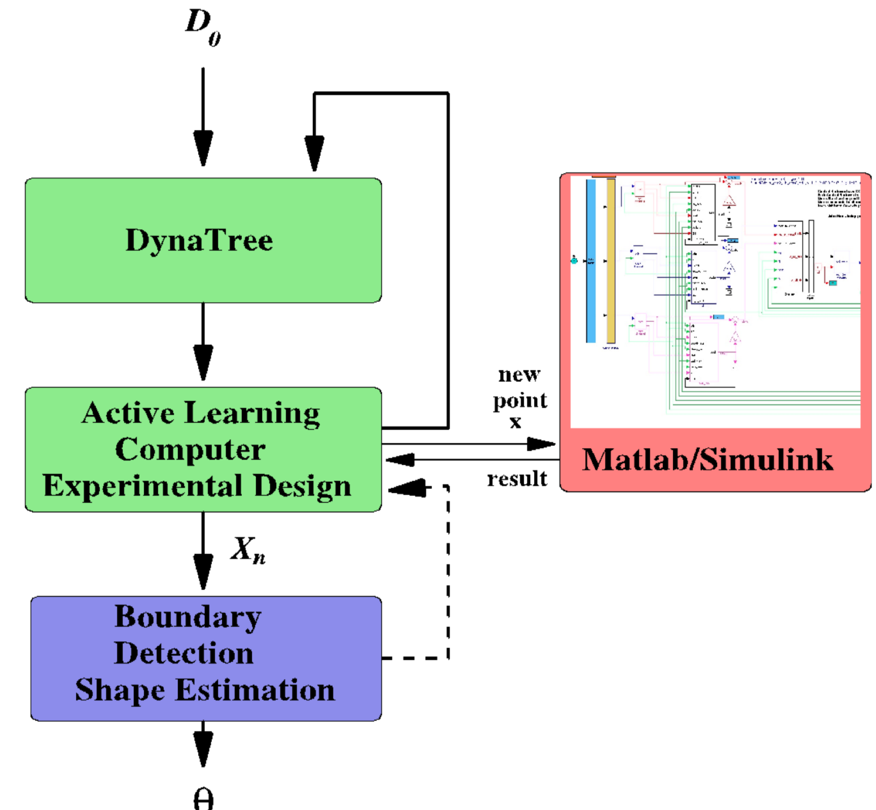


MarginS (left) and Monte Carlo (right)

Shape for conformance region of small deviations much clearer modeled

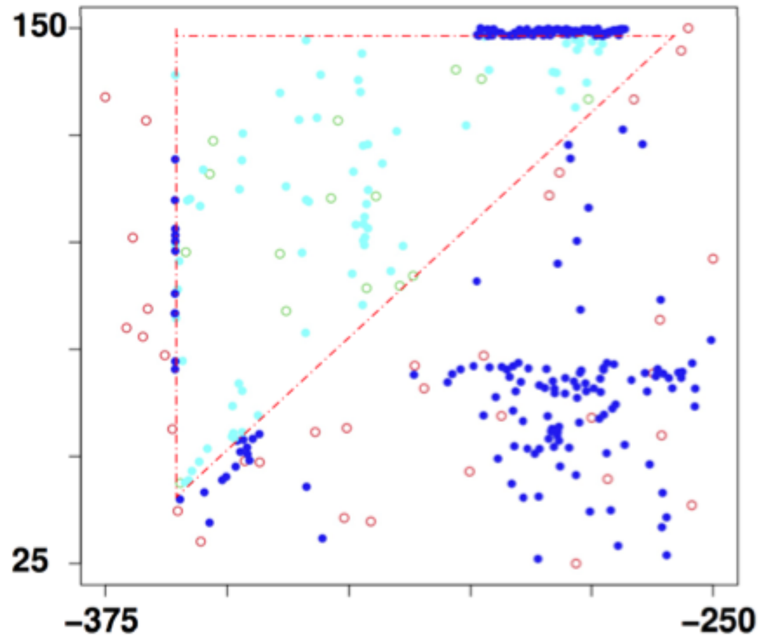
Application Example:
Terminal TSAFE – Safety Boundary Analysis

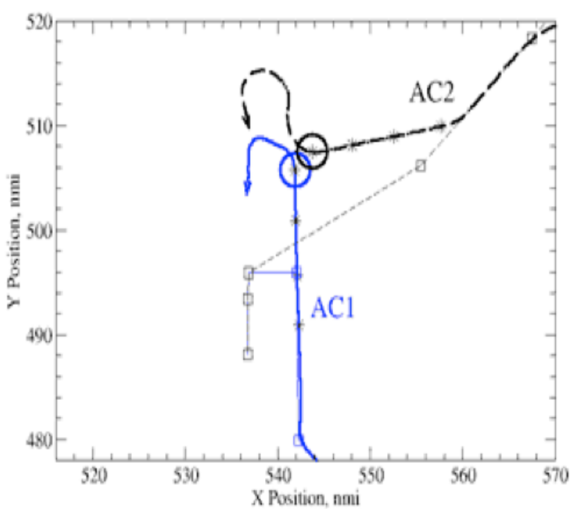




- Active learning for efficient sampling
- Bayesian modeling for boundary shape characterization

Terminal TSAFE – Boundary Detection and Characterization



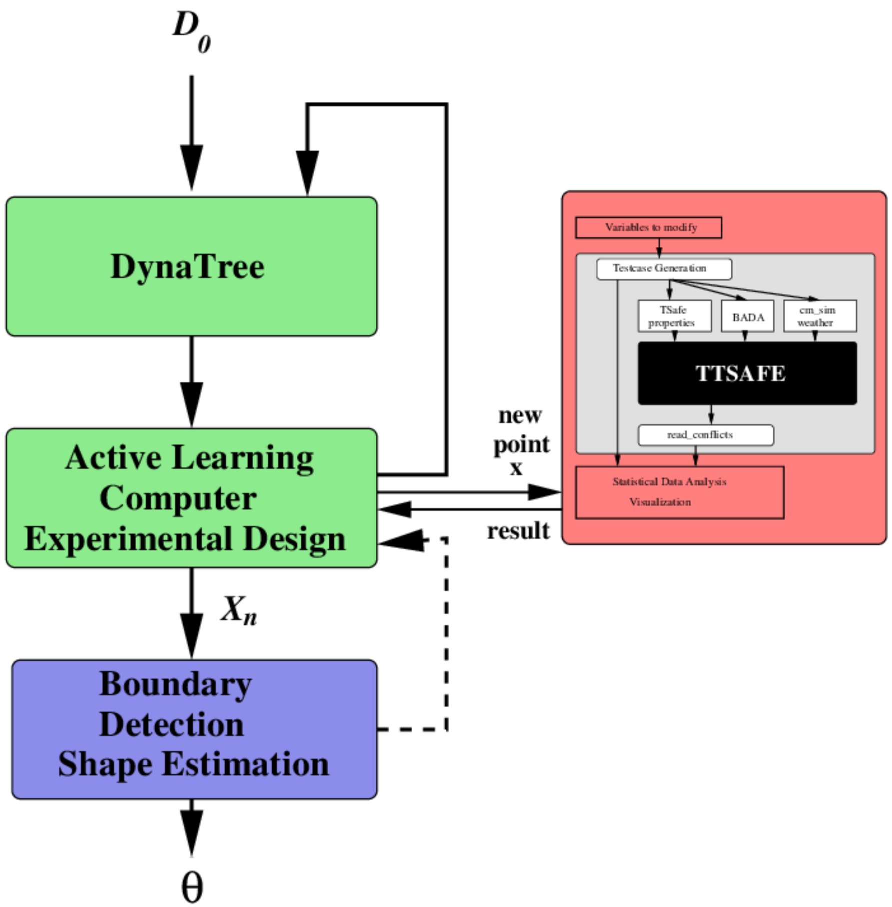


- High-dimensional safety boundary detection
- Boundary characterization as geometric shapes

Overview

- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

Overview of our Method



- **Active Learning to detect points near boundaries X_n**
- Estimation of shapes and shape parameters Θ

DynaTree-Background

- Likelihood: $p(y^t|x^t, T, \theta) = \prod_{\eta \in L_T} p(y^\eta|x^\eta, \theta_\eta)$
- Split Rule: $p_{split}(T, \eta) = \alpha(1 + D_\eta)^{-\beta}$ with $\alpha, \beta > 0$
- Joint Prior:

$$\pi(T) \propto \prod_{\eta \in I_T} p_{split}(T, \eta)\pi(T) \propto \prod_{\eta \in L_T} (1 - p_{split}(T, \eta))$$

- Likelihood after marginalization:

$$p(y^t|T_t, x^t) = \prod_{\eta \in L_{T_t}} p(y^\eta|x^\eta) = \prod_{\eta \in L_{T_t}} \int p(y^\eta|x^\eta, \theta_\eta) d\pi(\theta_\eta)$$

-

$$p([T,S]_t|[x,y]^t) = \int p([T,S]_t|[T,S]_{t-1}) dP([T,S]_{t-1}|[x,y]^t)$$

$$\propto \int p([T,S]_t|[T,S]_{t-1}, [x,y]_t) \int p([x,y]_t|[T,S]_{t-1}) dP([T,S]_{t-1}|$$

solved with resampling and propagation

Output Class Model

- Model class($\mathbf{x}$) using classification TGP model (CTGP)
- CTGP is an extension of TGP that handles categorical outputs
- Suppose M possible output classes $m = 1, \ldots, M$
- Introduce latent continuous variables $\{Z_m(\mathbf{x})\}_{m=1}^{M}$ to model

$$p_m(\mathbf{x}) = P(\mathrm{class}(\mathbf{x}) = m) = \frac{\exp(-Z_m(\mathbf{x}))}{\sum_{m'=1}^{M} \exp(-Z_{m'}(\mathbf{x}))}$$

- CTGP uses M independent TGP models for the mappings $\mathbf{x} \to Z_m$, $m = 1, \ldots, M$
- $\mathrm{class}(\mathbf{x}) \sim \mathrm{multinomial}(1, \mathbf{p}(\mathbf{x}))$ where $\mathbf{p}(\mathbf{x}) = (p_m(\mathbf{x}))_{m=1}^{M}$
- Actually only $M - 1$ latent variables $Z_m(\mathbf{x})$ are needed

Selection of Next Data Points

- General goal: candidate points should be near boundaries
- Maximum entropy $Y = -\sum_{c \in c_1,..,c_n} p_c \log p_c$ is too greedy
- Active Learning McKay (ALM): select maximum variance
- Active Learning Cohn (ALC): maximize reduction in predictive variance
- Expected Improvement (EI): maximize posterior expectation of improvement statistic

Limitation: ALM, ALC, EI do not take boundaries into account.

Boundary-aware metric

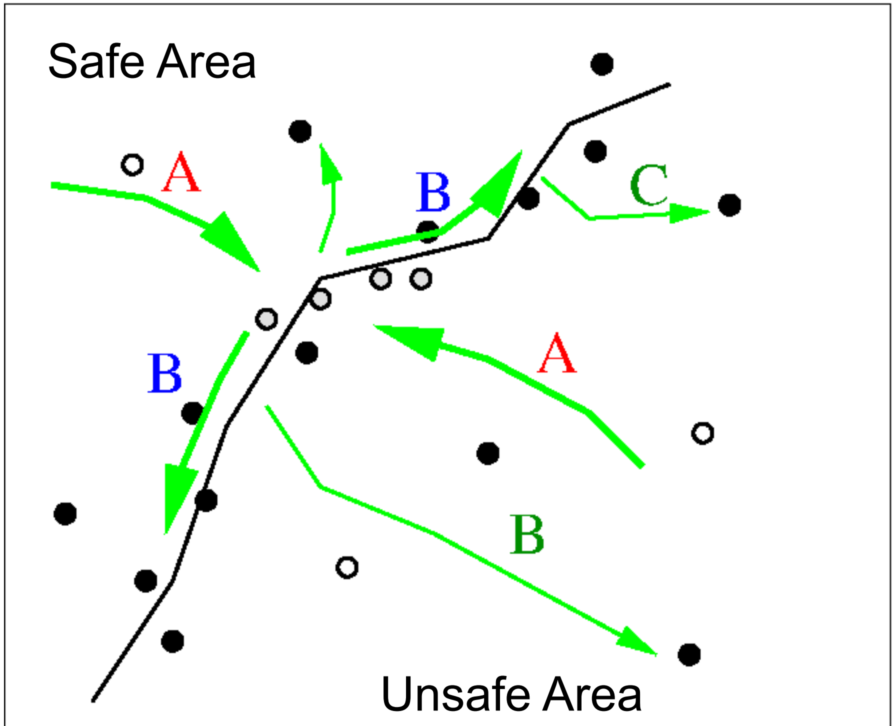
$$E[I(x)] = -\int_{0.5-\alpha s(x)}^{0.5+\alpha s(x)} (y - \hat{y}(x))^2 \phi\left(\frac{y-\hat{y}(x)}{\sigma(x)}\right) dy$$
$$+2(\hat{y} - 0.5)\sigma^2(x)\left[\phi\left(\frac{0.5-\hat{y}(x)}{\sigma(x)} + \alpha\right) - \phi\left(\frac{0.5-\hat{y}(x)}{\sigma(x)} - \alpha\right)\right]$$
$$+(\alpha^2\sigma^2(x) - (\hat{y}(x) - 0.5)^2)\left[\Phi\left(\frac{0.5-\hat{y}(x)}{\sigma(x)} + \alpha\right) - \Phi\left(\frac{0.5-\hat{y}(x)}{\sigma(x)} - \alpha\right)\right.$$

New test cases proposed:

A) Variability of response in neighborhood

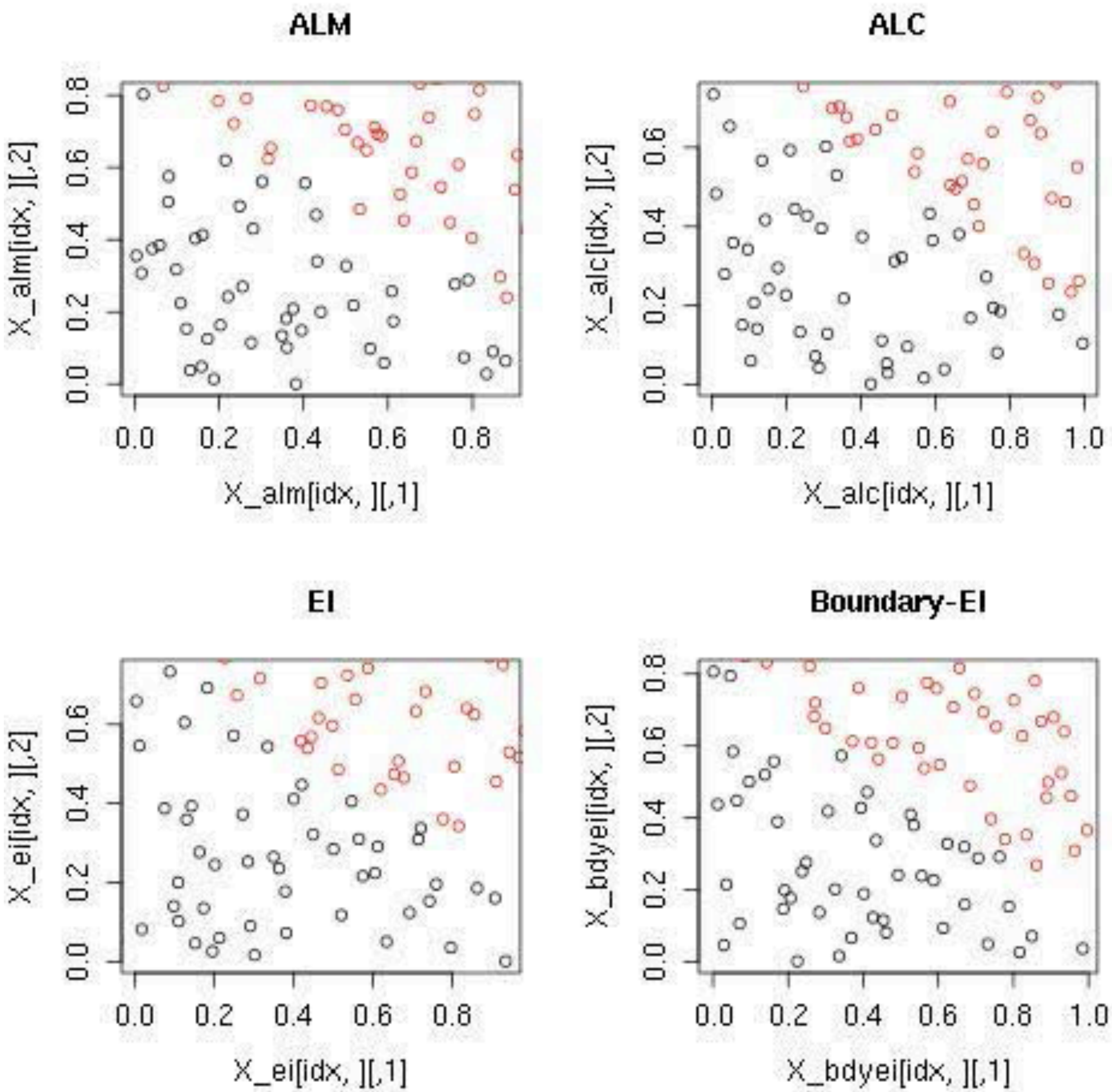
B) Farther away and in areas with high variance

C) Close to estimated boundary

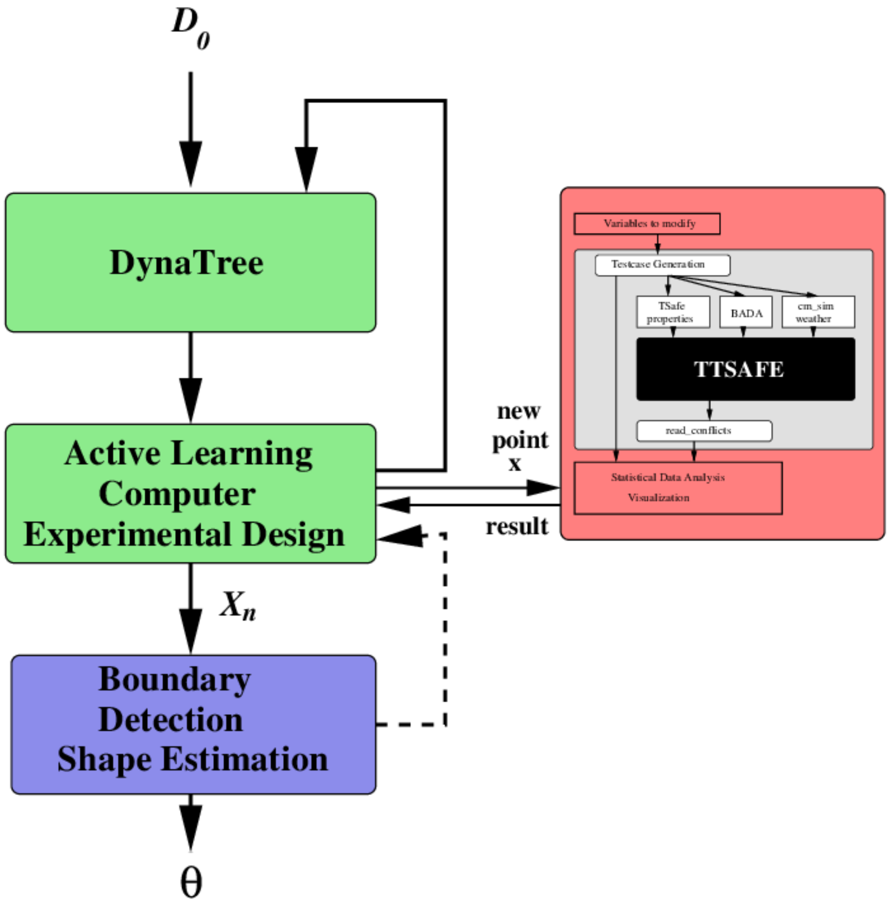


[He2015, He2016]

Selection of New Test Points



Overview of our Method



- Active Learning to detect points near boundaries X_n
- **Estimation of shapes and shape parameters Θ**

Characterizing Boundaries

- A common metric to describe a boundary uses the entropy $Y(x) = -\sum_{c \in c_1,..,c_C} p(x = c) \log p(x = c)$. $Y(x)$ becomes maximal for x on a boundary
- The metric advantage $adv(x) = |p(x = success) - p(x = failure)|$ becomes minimal on the boundary.
- A classification method that can produce posterior probabilities can directly be used to select points which are close to a boundary
- In general, a k-nearest neighbor approach can be used to determine points close to the boundary. This approach is slow $O(n^2)$

Statistical Modeling

- Posterior $P(\mathcal{S}|X_n) \propto P(X_n|\mathcal{S})P(\mathcal{S})$
- Likelihood $P(X_n|\mathcal{S})$ models completeness (next)
- Prior $P(\mathcal{S})$ models minimality of complete shape sets
  - encourage intershape distance $\overline{D}^2_{\mathcal{S}}$ to be large
  - $\mathcal{S} \sim N(\overline{D}_{\mathcal{S}}^{-1}; 0, \sigma^2_{\text{shapesim}})$
- Bayesian Loss models summary: $\text{loss}(\mathcal{S}, X_n) = \lambda_{\text{summary}} \overline{D}^2_{\mathcal{S},X_n}$

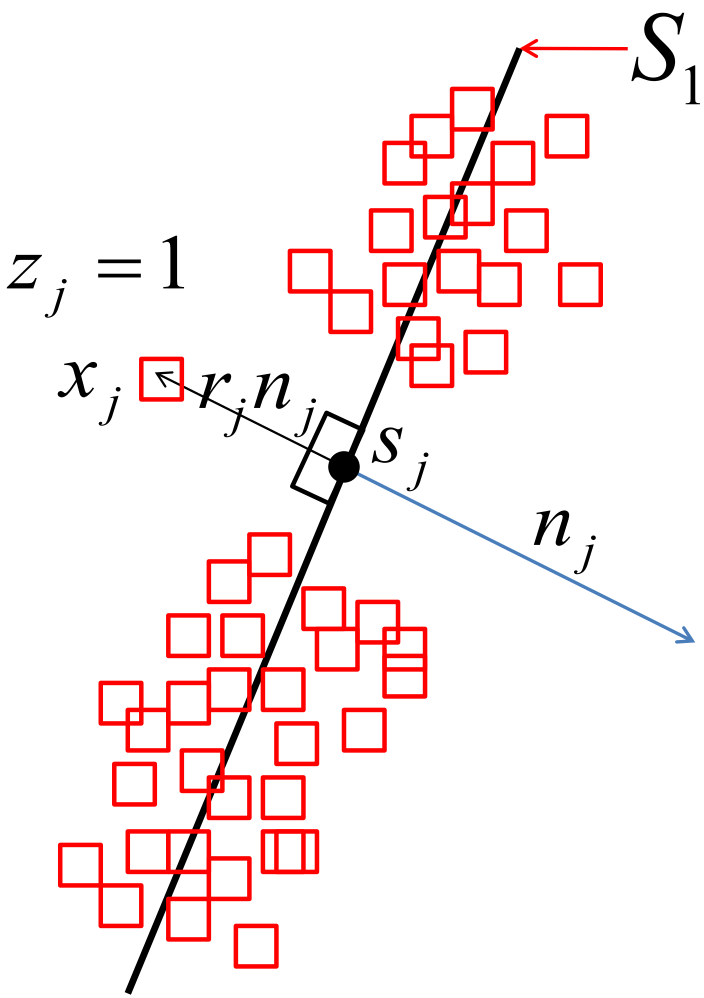
Step 1 Minimize the expected loss

$$g(l) = E[\text{loss}(\mathcal{S}, X_n)], \quad |\mathcal{S}| = l$$

over the shape set size l to obtain the number of shapes l^*

Step 2 Compute the MAP shape set $\mathcal{S}^{*,l^*}$ for shape sets of size l^*

Likelihood



$$x_j = s_j + r_j n_j$$

$$r_j \sim N(0, \sigma^2)$$

$$s_j \sim U(S_{z_j})$$

$$x_j \mid z_j, S_{z_j} \sim N(r_j; 0, \sigma^2)$$

$$r_j^2 = \min_{s_j \in S_{z_j}} \left\| x_j - s_j \right\|_2^2$$

Likelihood

$$
\begin{aligned}
& P(x_1, \dots, x_n | z_1, \dots, z_n, S_1, \dots, S_l) \\
&= \prod_{j=1}^{n} P(x_j | z_1, \dots, z_n, S_1, \dots, S_l) \\
&= \prod_{j=1}^{n} P(x_j | z_j, S_{z_j}) = \prod_{j=1}^{n} N(r_j; 0, \sigma^2) \\
&= C\sigma^{-n} \prod_{j=1}^{n} \exp(-0.5\sigma^{-2} r_j^2) = C\sigma^{-n} \exp(-0.5\sigma^{-2} \sum_{j=1}^{n} r_j^2)
\end{aligned}
$$

$$
P(X|Z, \mathcal{S}) = C\sigma^{-n} \exp(-0.5\sigma^{-2} \sum_{j=1}^{n} \min_{s_j \in S_{z_j}} ||x_j - s_j||_2^2)
$$

Likelihood is maximized by choosing shape set $\mathcal{S}$ such that all points in X are close to some shape in $\mathcal{S}$ (completeness)

Likelihood

- Model $(z_1, \ldots, z_n)|\mathcal{S}$ to encourage each of $I = |\mathcal{S}|$ shapes to generate n/I points

$$
\begin{aligned}
c_i &= \sum_{j=1}^{n} 1_{z_j = i} \\
C = (c_1, \ldots, c_I) &\sim \mathrm{multinomial}(n, (1/I, 1/I, \ldots, 1/I)) \\
P(X|\mathcal{S}) &= \int_Z P(X|Z, \mathcal{S}) P(Z|\mathcal{S}) dZ
\end{aligned}
$$

- Implies we expect to see points around each shape

Overview

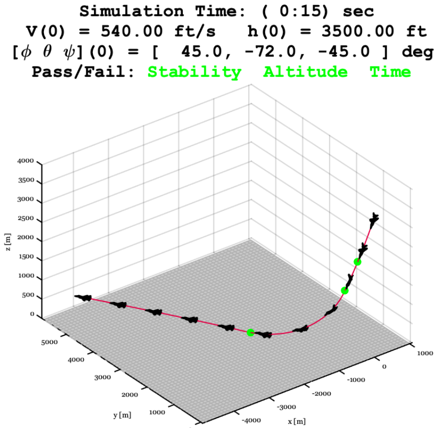
- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

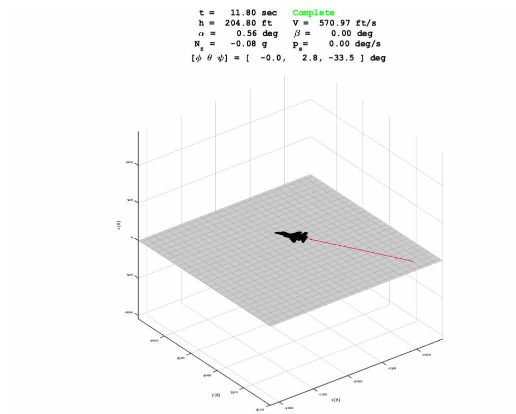
Demo I

- Find safety boundary for an example Simulink Model
- The Simulink model – a “complex” system

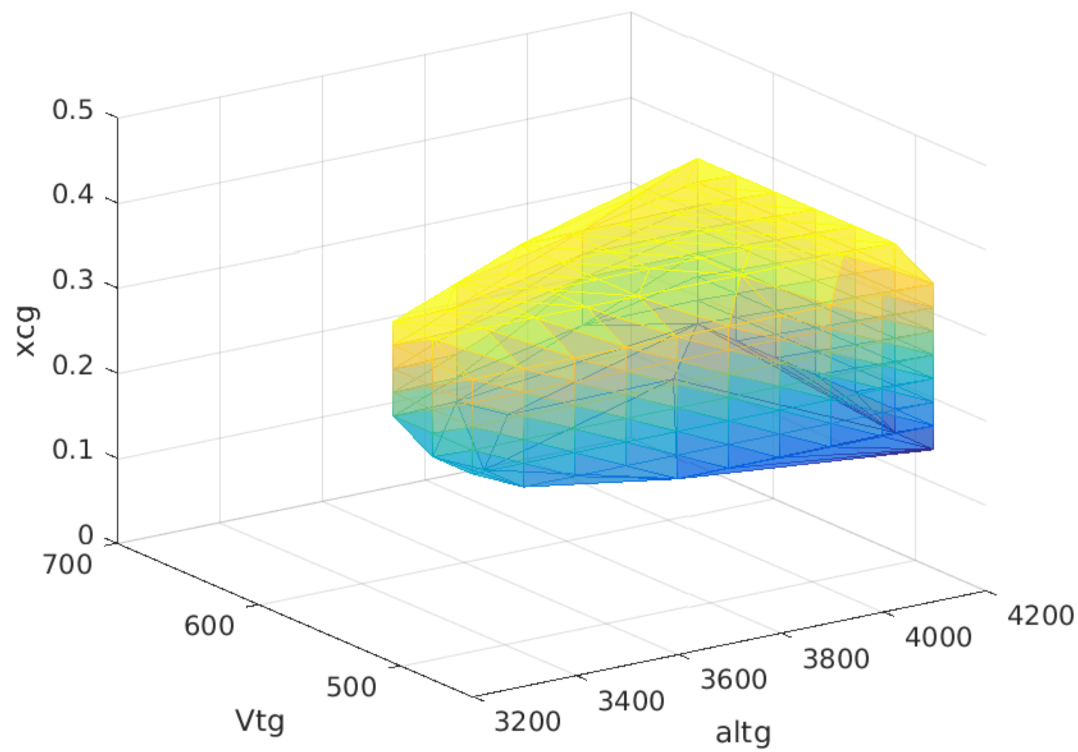
Demo II

- Connection of a realistic system, the *Ground Collision Avoidance System (GCAS)*
  - Control system to stabilize F16 without ground collision
  - Challenging problem for V&V
  - Matlab system based on AeroBenchVV





Demo II Results



- Estimated shape for safety envelope in 3D projection

Overview

- Introduction
- Applications of MARGInS
- MARGInS Architecture
  - Tool Interfaces
  - Boundary detection and characterization
- Theory behind the tool
- Demos
- Summary

Summary

- MARGInS is a flexible framework and can be applied for complex system's:
  - Safety and Performance Analysis
  - V&V of Deep Neural Networks
  - V&V of Autonomous Systems
  - Prognostics
  - Runtime verification/Monitoring

Scalability

- Tool can handle *large and complex systems*
  - The system is seen as a black box and is simulated. So resources, runtime depends on that.
  - Easy and flexible interface to the system
- Tool can handle systems with *large number of parameters* (high dimensionality)
  - Algorithms are for analysis of high-dimensional spaces
  - Tool contains functionality for an explainable reduction of dimensionality

Use of Tool during SW process

- Model analysis during *early* design stages
  - Provide feedback to designer
- Supports unit testing of complex components, e.g., DNN
- Analysis of complex system as a black box during system integration
- Should be useful for Processor-in-the-loop and HW-in-the-loop as it provides *informative and valuable* test cases
- During deployment for diagnosis, prognostics, and runtime verification

List of published Papers

- Y. He and J. Schumann “A Framework for Online Testing of Deep Neural Networks using Bayesian Statistics and Active Learning”, DeepTest (ICSE Workshop on Deep Learning and Testing), 2019.
- Y. He, “Online Detection and Modeling of Safety Boundaries for Aerospace Applications using Bayesian Statistics”, 2015 International Joint Conference on Neural Networks (IJCNN)
- Y. He, “Predicting Time Series Outputs and Time-to-Failure for an Aircraft Controller using Bayesian Statistics”, SIAM 2015 SIAM Conference on Control& Its Application
- Y. He, “Detection and modeling of high-dimensional thresholds for Fault Detection and Diagnosis using Bayesian Statistics.” 2015 IEEE International Conference on Prognostic and Health Management"
- Y. He, and M. Davies. “Bayesian Statistics for Complex Systems Safety Analysis.” IEEE Software Technologies Conference, 2014
- Y. He, and M. Davies. “Validating an Air Traffic Management Concept of Operation using Statistical Modeling.” AIAA Modeling and Simulation Technologies Conference, 2013

Team

- Karen Bundy-Gurlet
- Misty Davies
- Yuning He
- Tom Pressburger
- Johann Schumann