Cyber-Physical Systems

Laura Nenzi

Università degli Studi di Trieste
I Semestre 2023

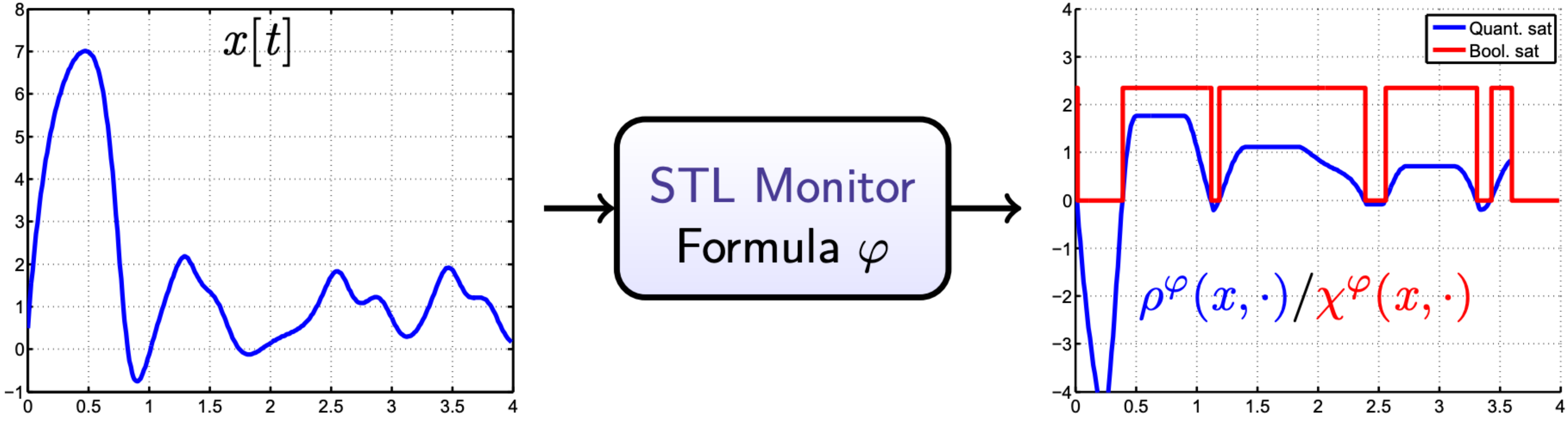
Lecture 19: STL applications

[Many Slides due to J. Deshmukh, S. Silvetti]

Terminology

- **Syntax**: A set of syntactic rules that allow us to construct formulas from specific ground terms
- **Semantics**: A set of rules that assign meanings to well-formed formulas obtained by using above syntactic rules
- **Model-checking/Verification**: $M \vDash \phi \iff \forall \mathbf{x} \in trace(M) \;\; s(\varphi, \mathbf{x}, 0) = 1$
- **Monitoring**: computing s for a single trace $\mathbf{x} \in trace(M)$
- **Statistical Model Checking**: "doing statistics" on s$(\varphi, \mathbf{x}, 0)$ for a finite-subset of $trace(M)$

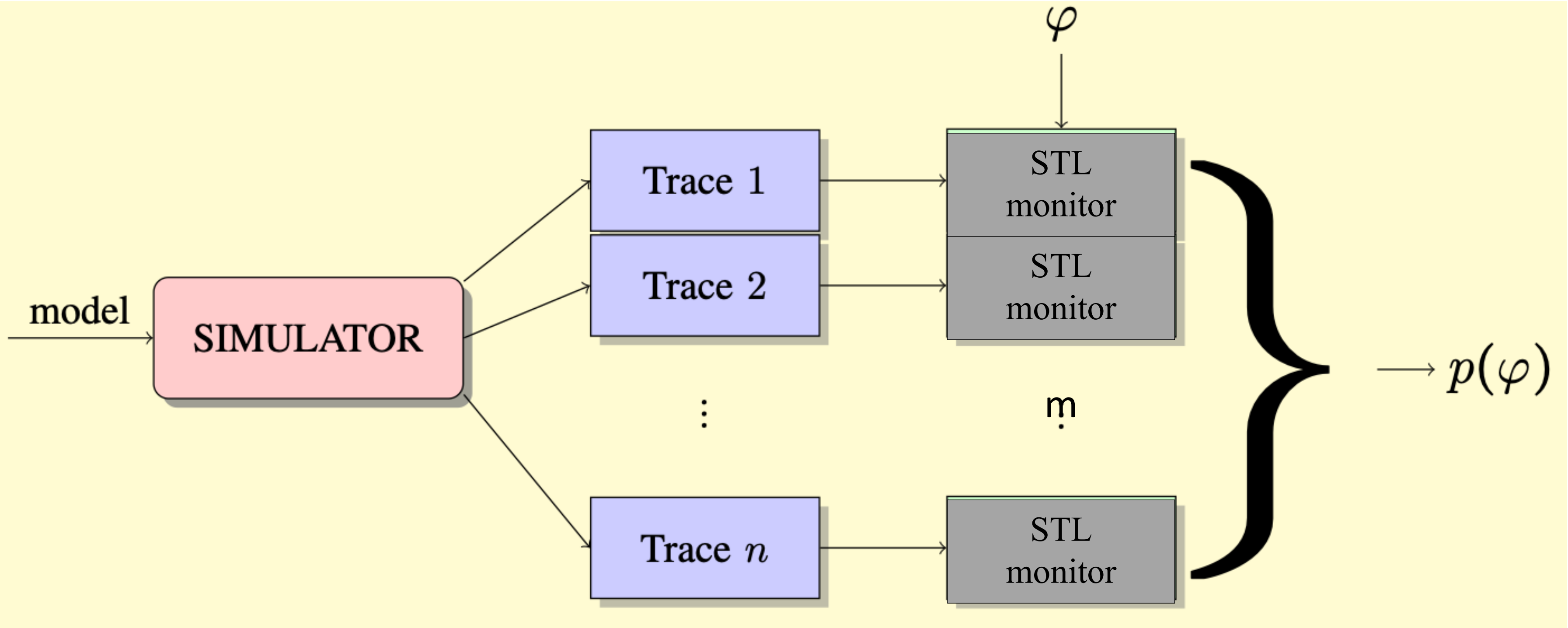
STL Monitor



An STL monitor is a transducer that transforms x into Boolean or a quantitative signal

Statistical Model Checking (SMC)

The probability satisfaction can be estimated as an average of the truth values T_i of the formula φ over many sample trajectories.

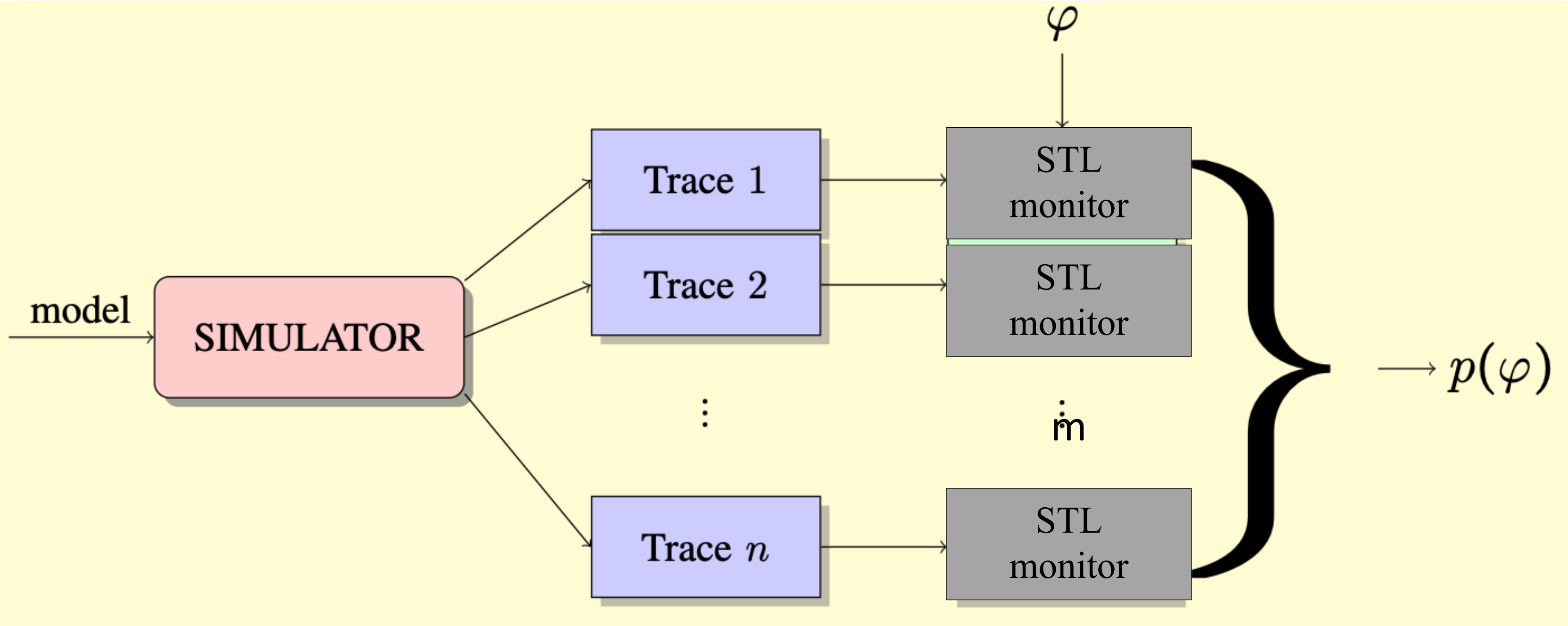


m

Bayesian SMC uses the fact the satisfaction probability of a formula given a model is a number in [0,1], and prior distributions on numbers between [0,1] exist (Beta distribution)}

Statistical Model Checking

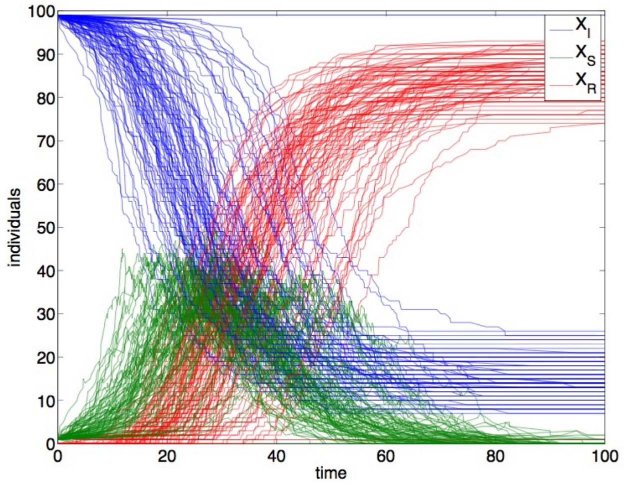
- **Statistical Model Checking:** p_ϕ can be estimated as an average of the truth values T_i of the formula ϕ over many sample trajectories.
- **Bayesian SMC** specifying (Beta) priors $prob\{p_\phi\}$ and estimating a posteriori $prob\{p_\phi | T_i\}$ using Bayes' theorem and the fact that $prob\{T_i | p_\phi\}$ is Bernoulli.

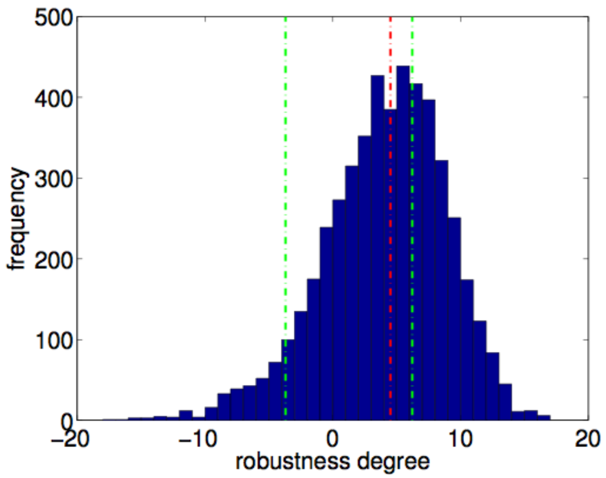


Average robustness degree

Robustness Distribution

$$\mathbb{P}\left(R_{\varphi}(\mathbf{X}) \in [a,b]\right) = \mathbb{P}\left(\mathbf{X} \in \{\mathbf{x} \in \mathcal{D} \mid \rho(\varphi, \mathbf{x}, 0) \in [a,b]\}\right)$$





Indicators

$\mathbb{E}(R_{\varphi})$ (the average robustness degree)

$\mathbb{E}(R_{\varphi} \mid R_{\varphi} > 0)$ and $\mathbb{E}(R_{\varphi} \mid R_{\varphi} < 0)$ (the conditional averages)

The many uses of STL

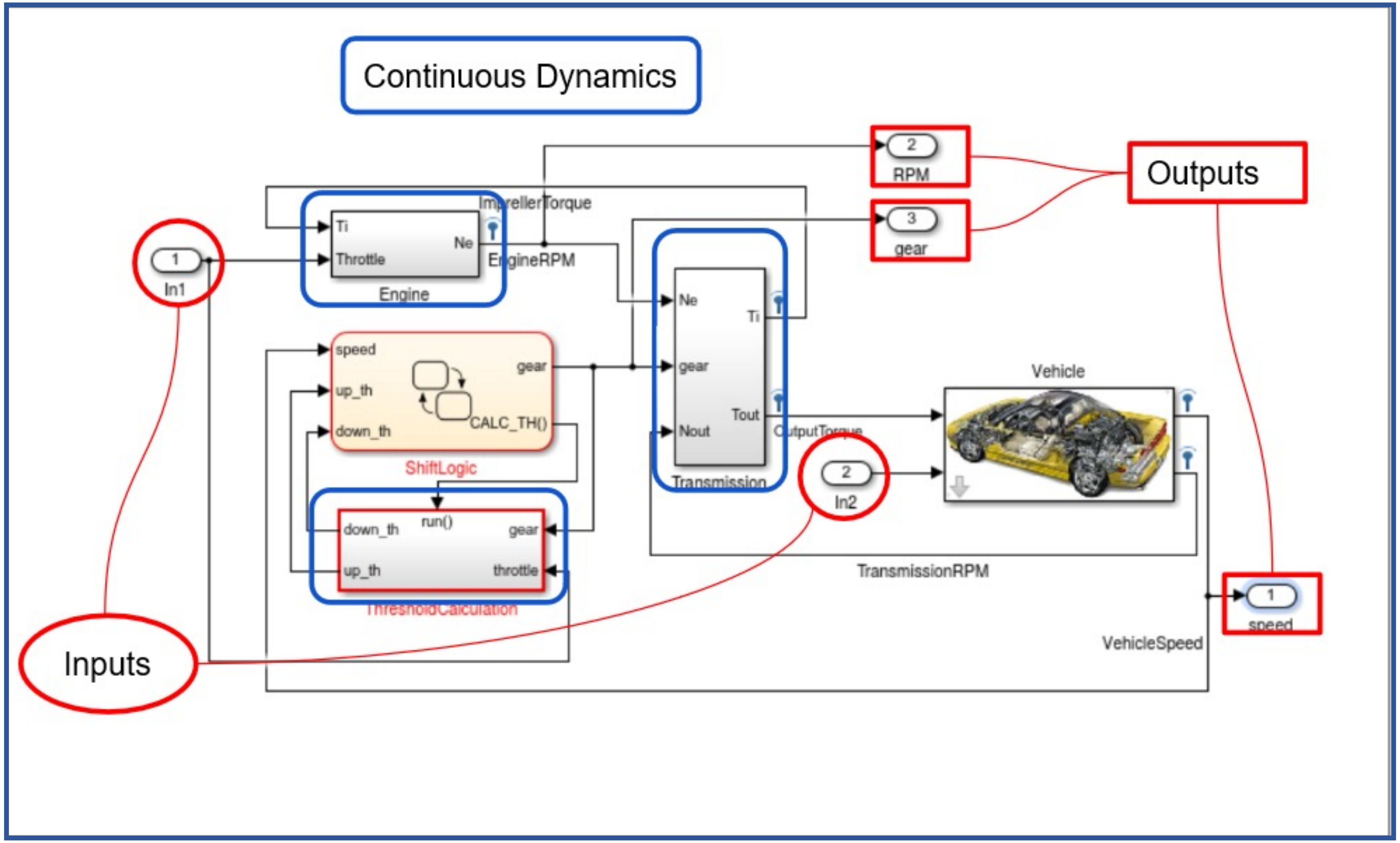
- Requirement-based testing for closed-loop control models
- Falsification Analysis
- Parameter Synthesis
- Mining Specifications/Requirements from Models
- Online Monitoring
- …

Example

Inputs:

Throttle

Brake



Outputs:

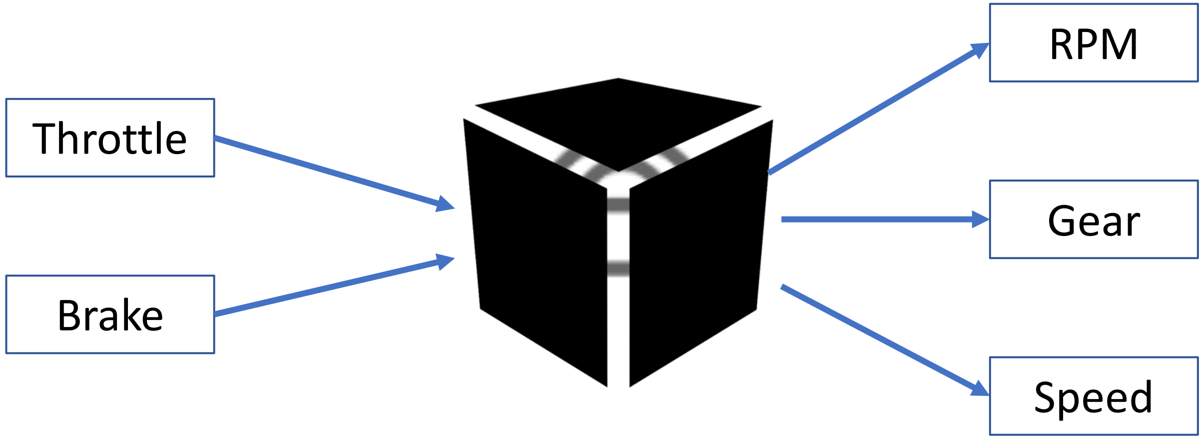
RPM

Gear

Speed

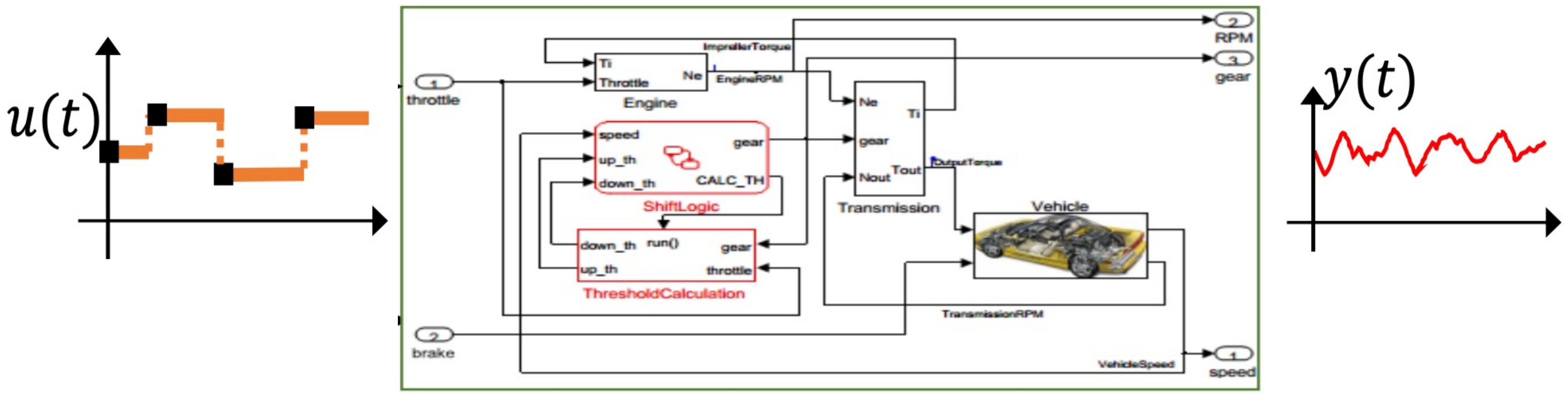
Simulink model of a Car Automatic Gear Transmission Systems

Black Box Assumption

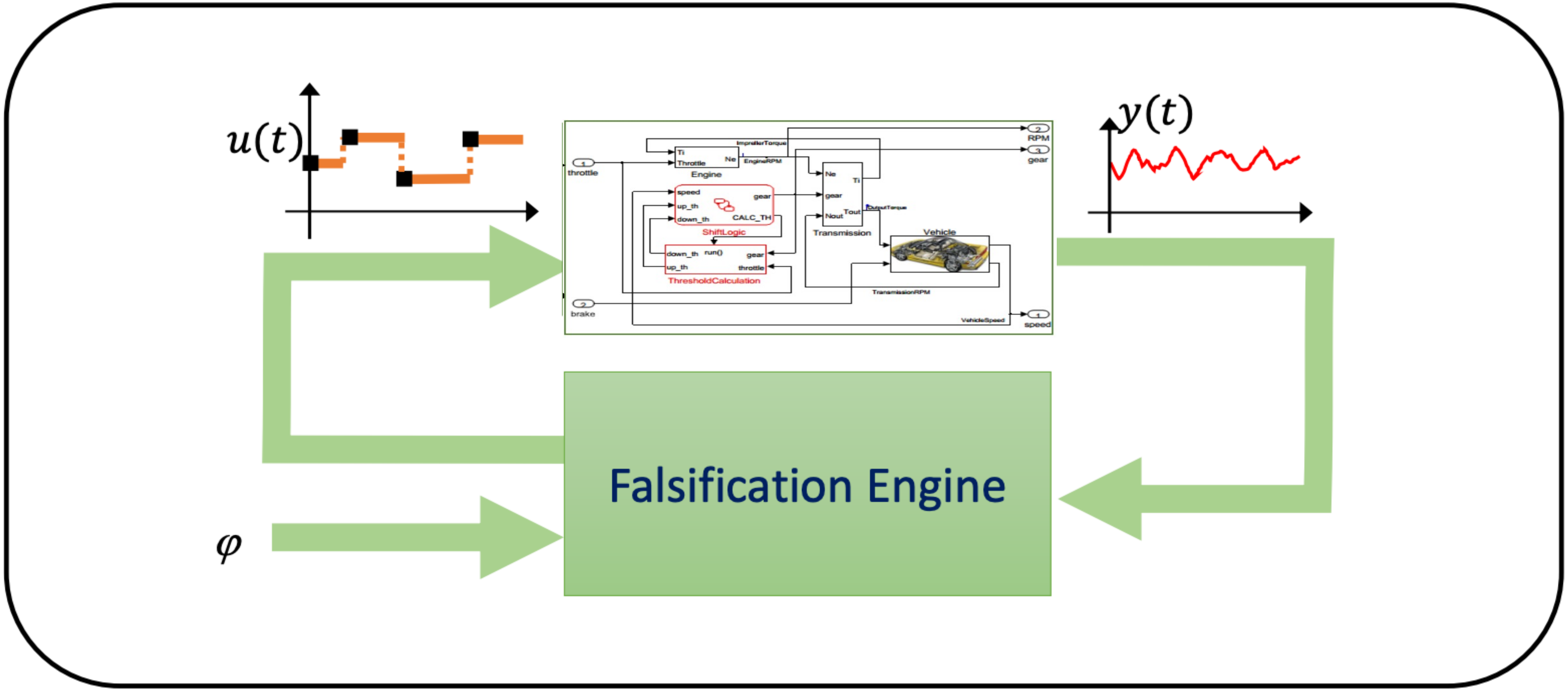


Black Box Assumption

- For simplicity, consider the composed plant model, controller and communication to be a model M that is excited by an input signal $\mathbf{u}(t)$ and produces some output signal $\mathbf{y}(t)$



Falsification/Testing



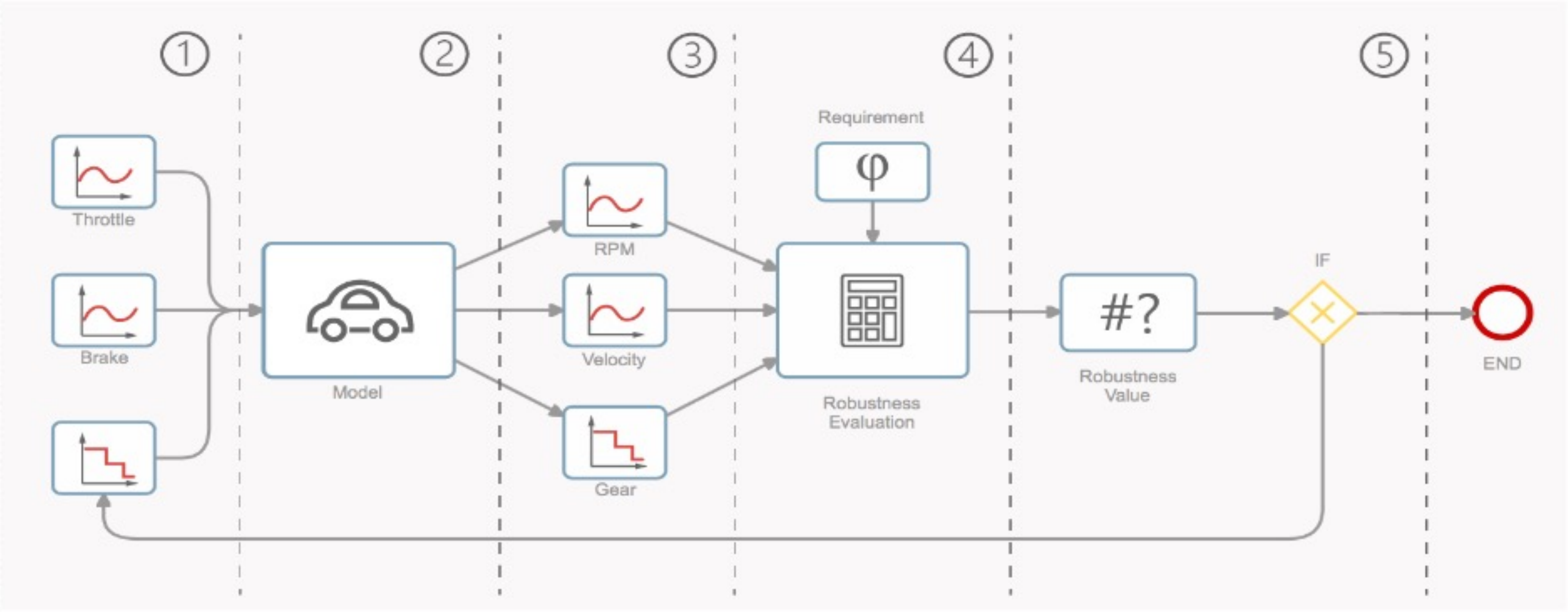
Verification vs. Testing

- For simplicity, $\mathbf{u}$ is a function from $\mathbb{T}$ to $\mathbb{R}^m$; let the set of all possible functions representing input signals be U
- Verification Problem:

  Prove the following: $\forall \mathbf{u} \in U \colon \left(\mathbf{y} = M(\mathbf{u})\right) \vDash \varphi(\mathbf{u}, \mathbf{y})$
- Falsification/Testing Problem:

  Find a witness to the query: $\exists \mathbf{u} \in U : \left(\mathbf{y} = M(\mathbf{u})\right) \nvDash \varphi(\mathbf{u}, \mathbf{y})$
- These formulations are quite general, as we can include the following "*model uncertainties*" as input signals: Initial states, tunable parameters in both plant and controller, time-varying parameter values, noise, etc.,

Falsification CPS



Goal:

Find the inputs (1) which falsify the requirements (4)

Problems:

- Falsify with a low number of simulations
- Functional Input Space

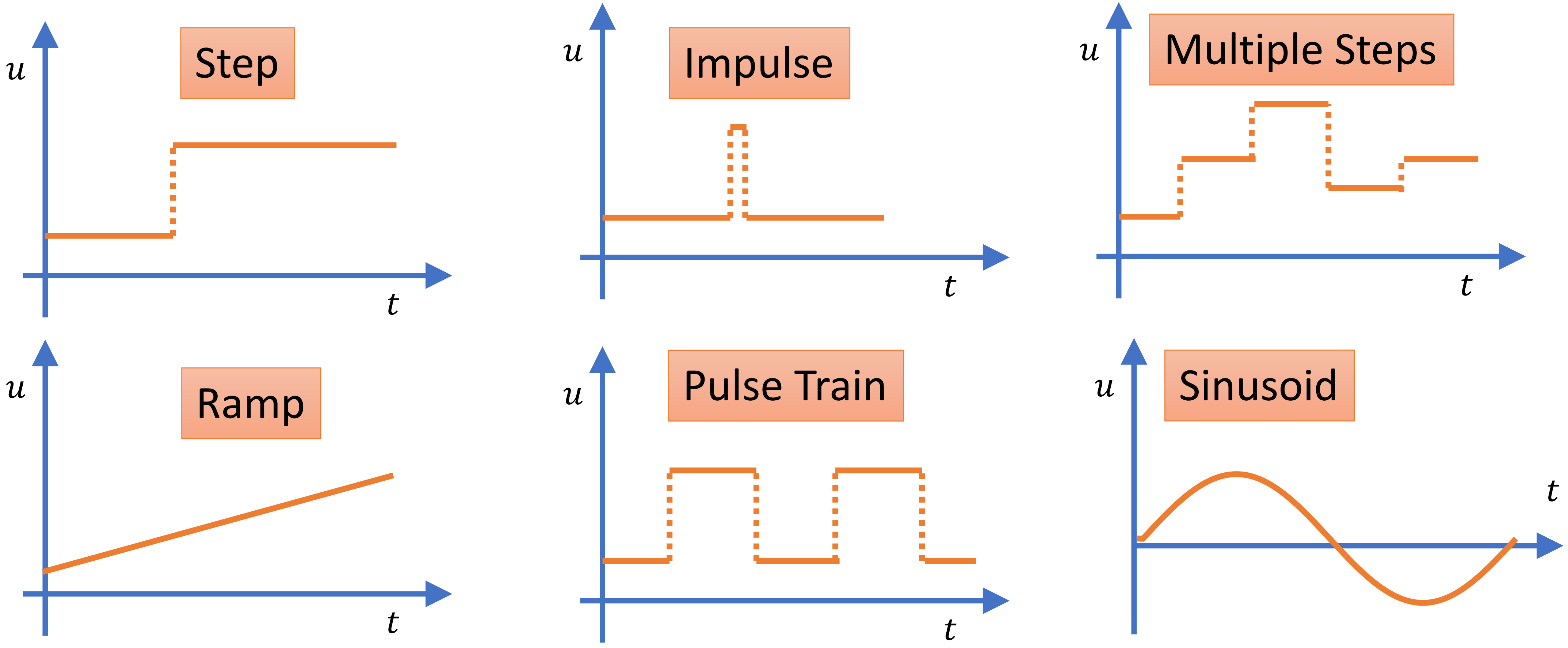
Falsification re-framed

Given:

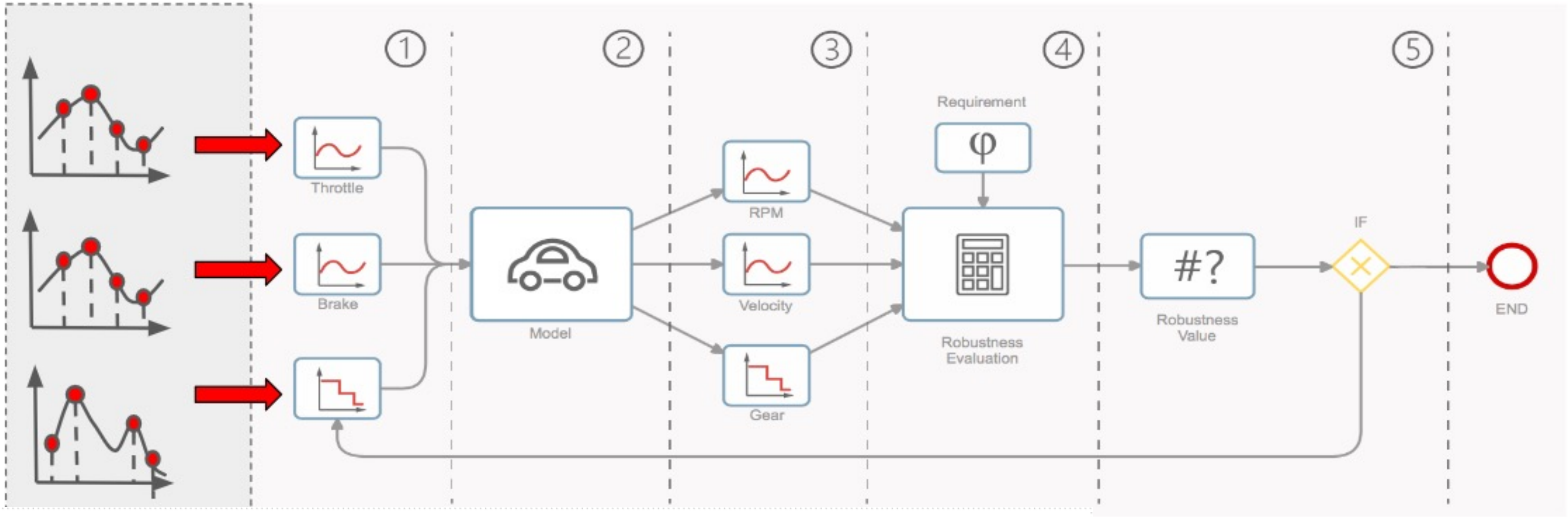
- Set of all such input signals : U
- Input signal $\mathbf{u} : \mathbb{T} \rightarrow D_1 \times \cdots \times D_m$, where $\mathbb{T} \subseteq [0, T], D_i \subset \mathbb{R}$ compact set
- Model M s.t. $M(\mathbf{u}) = \mathbf{y}, \quad \mathbf{y} : \mathbb{T} \rightarrow \mathbb{R}^n$
  M maps $\mathbf{u}$ to some signal $\mathbf{y}$ with the same domain as $\mathbf{u}$, and co-domain some subset of $\mathbb{R}^n$
- Property φ that can be evaluated to true/false over given $\mathbf{u}$ and $\mathbf{y}$

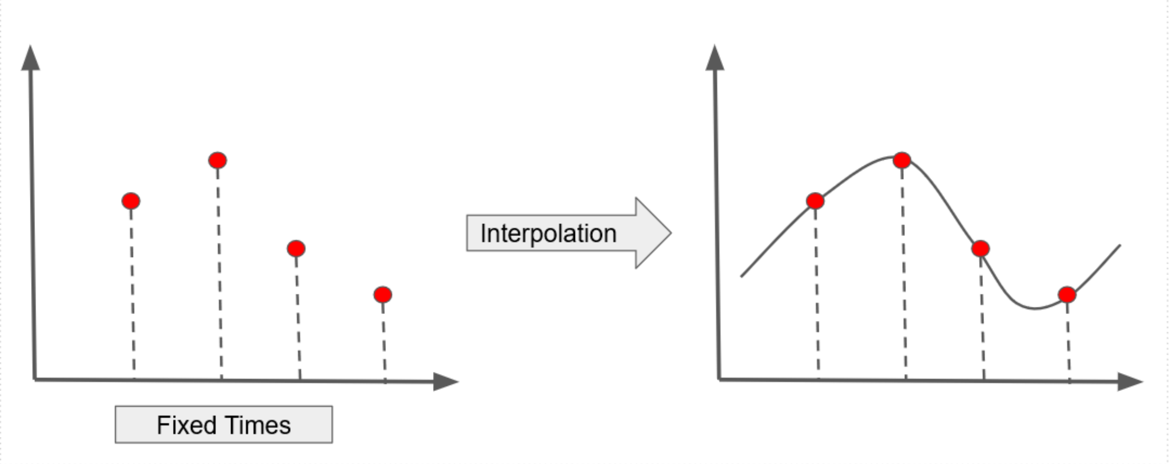
Check: $\exists \mathbf{u} \in U : \big(\mathbf{y} = M(\mathbf{u})\big) \vDash \neg\varphi(\mathbf{u}, \mathbf{y})$

Common input patterns used for testing



Finite Parameterization





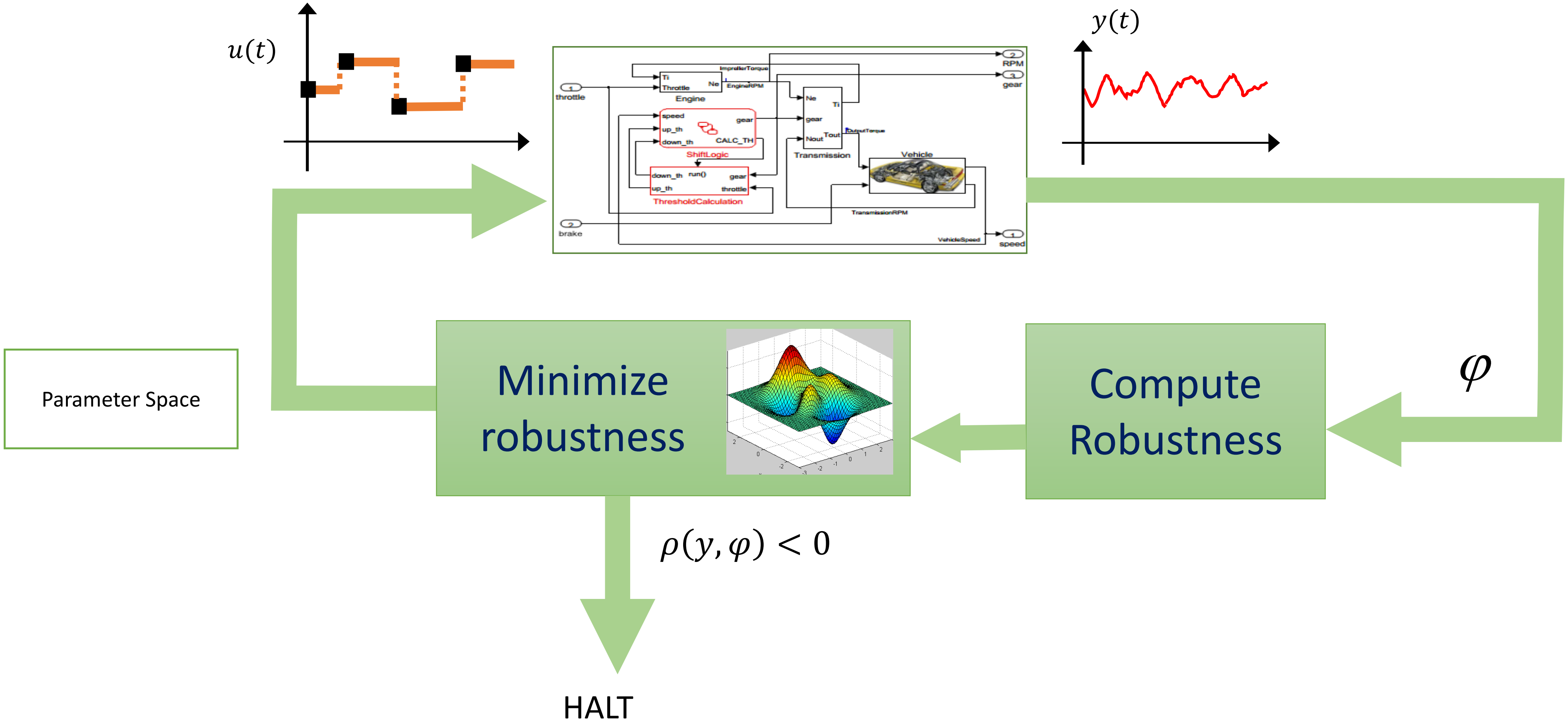
N Control points

N variable

Step-by-step of how falsification works

- Given: a finite parameterization for input signals, a model that can be simulated and an STL property
- While the number of allowed iterations is not exhausted do:
  - pick values for the signal parameters
  - generate an input signal
  - run simulation with generated input signal to get output signal
  - compute robustness value of given property w.r.t. the input/output signals
  - if robustness value is negative, **HALT**
  - pick a new set of values for the signal parameters based on certain heuristics

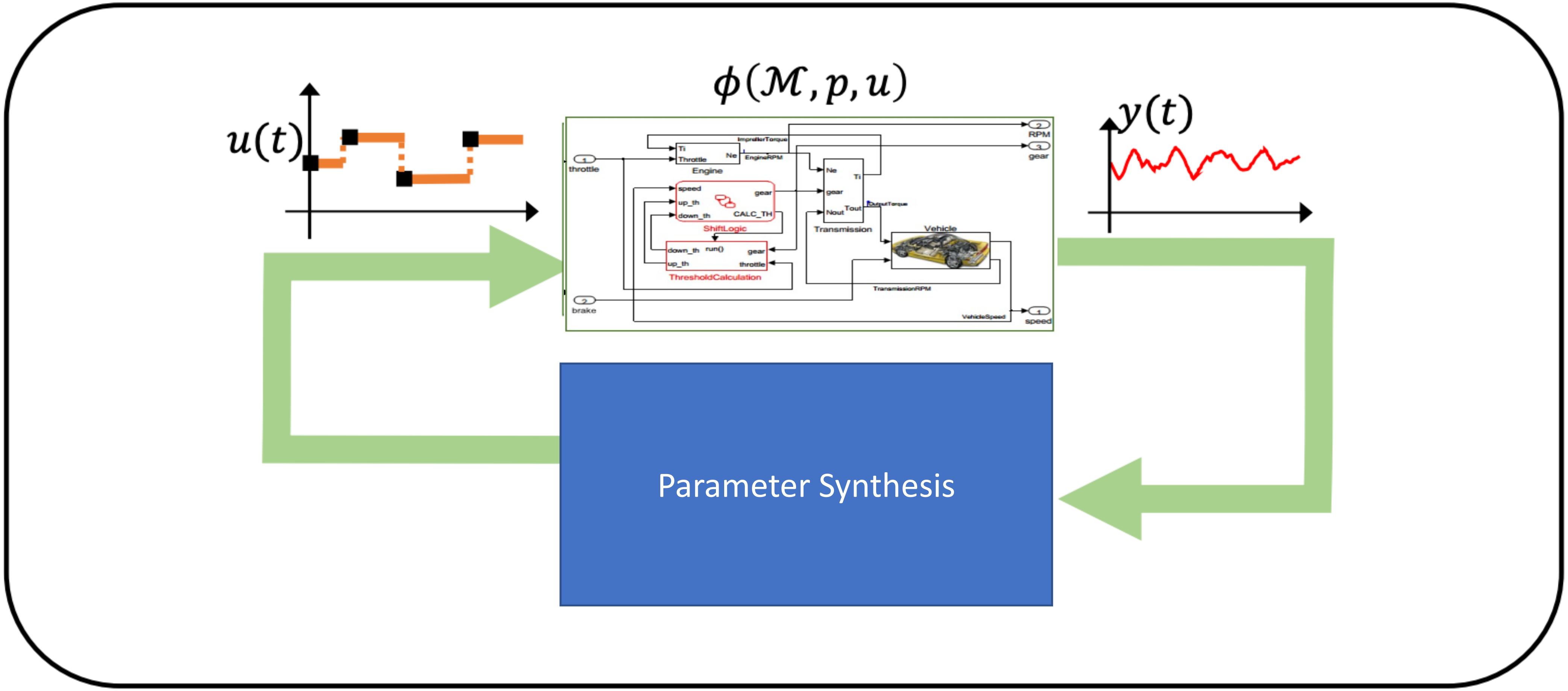
Falsification using Optimization



Picking new parameter values to explore

- Pick random sampling as a (not very good) strategy!
- Basic method: locally approximate the gradient of the function ρ locally, and chose the direction of steepest descent (greedy heuristic to take you quickly close to a local optimum)
- Challenge 1: cost surface may not be convex, thus you could have many local optima
- Challenge 2: cost surface may be highly nonlinear and even discontinuous, using just gradient-based methods may not work well
- Heuristics rely on:
  - combining gradient-based methods with perturbing the search strategy (e.g. simulated annealing, stochastic local search with random restarts)
  - evolutionary strategies: Covariance Matrix Adaptation Evolution Strategy (CMA-ES), genetic algorithms etc.
  - probabilistic techniques: Ant Colony Optimization, Cross-Entropy optimization, Bayesian optimization

Parameter Synthesis



Parameter Synthesis

Problem

Given a model, depending on a set of parameters $\theta \in \Theta$, and a specification ϕ (STL formula), find the parameter combination θ s.t. the system satisfies ϕ as more as possible

Solution Strategy

- **rephrase** it as a optimisation problem (maximizing ρ)
- **evaluate** the function to optimise
- **solve** the optimisation problem

Parameter Synthesis

Problem

Find the parameter configuration that maximizes $E[R_\phi](\theta)$, of which we have few costly and noisy evaluations.

Methodology

1. Sample $\{(\theta_{(i)}, y_{(i)}), i = 1,\ldots,n\}$
2. Emulate (**GP Regression**): $E[R_\phi] \sim GP(\mu, k)$
3. Optimize the emulation via **GP-UCB algorithm**, new $\theta_{(n+1)}$

Gaussian Process Regression

Gaussian Processes can be used for Bayesian prediction and classification tasks.

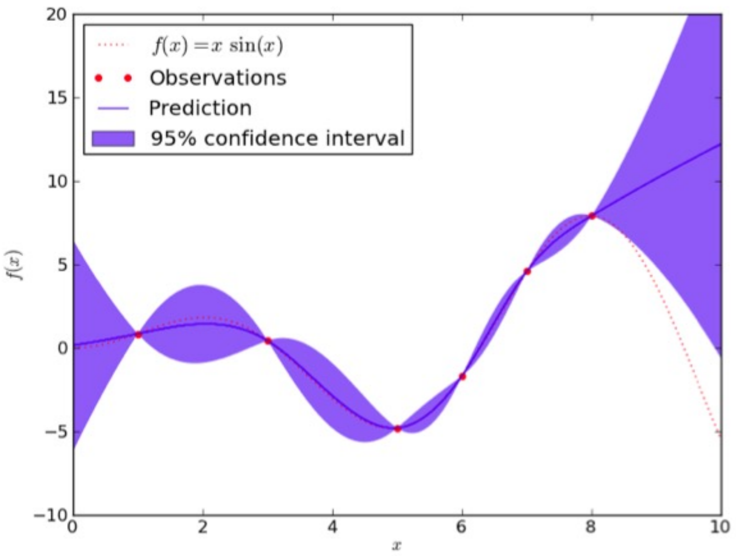
Idea: put a **GP prior** on functions; condition on **observed data (training set)** (x_i, y_i); we compute a **posterior** distribution on functions; make **predictions**.

Latent function: f , GP ; **Noise model**: $p(y_i | f(x_i))$

Prediction (latent function f^* at x^*)

$$p(f^*|\mathbf{y}) \propto \int df(\mathbf{x}) p(f^*, f(\mathbf{x})) p(\mathbf{y}|f(\mathbf{x}))$$

Under Gaussian noise $y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$
predictions have an analytic expression.

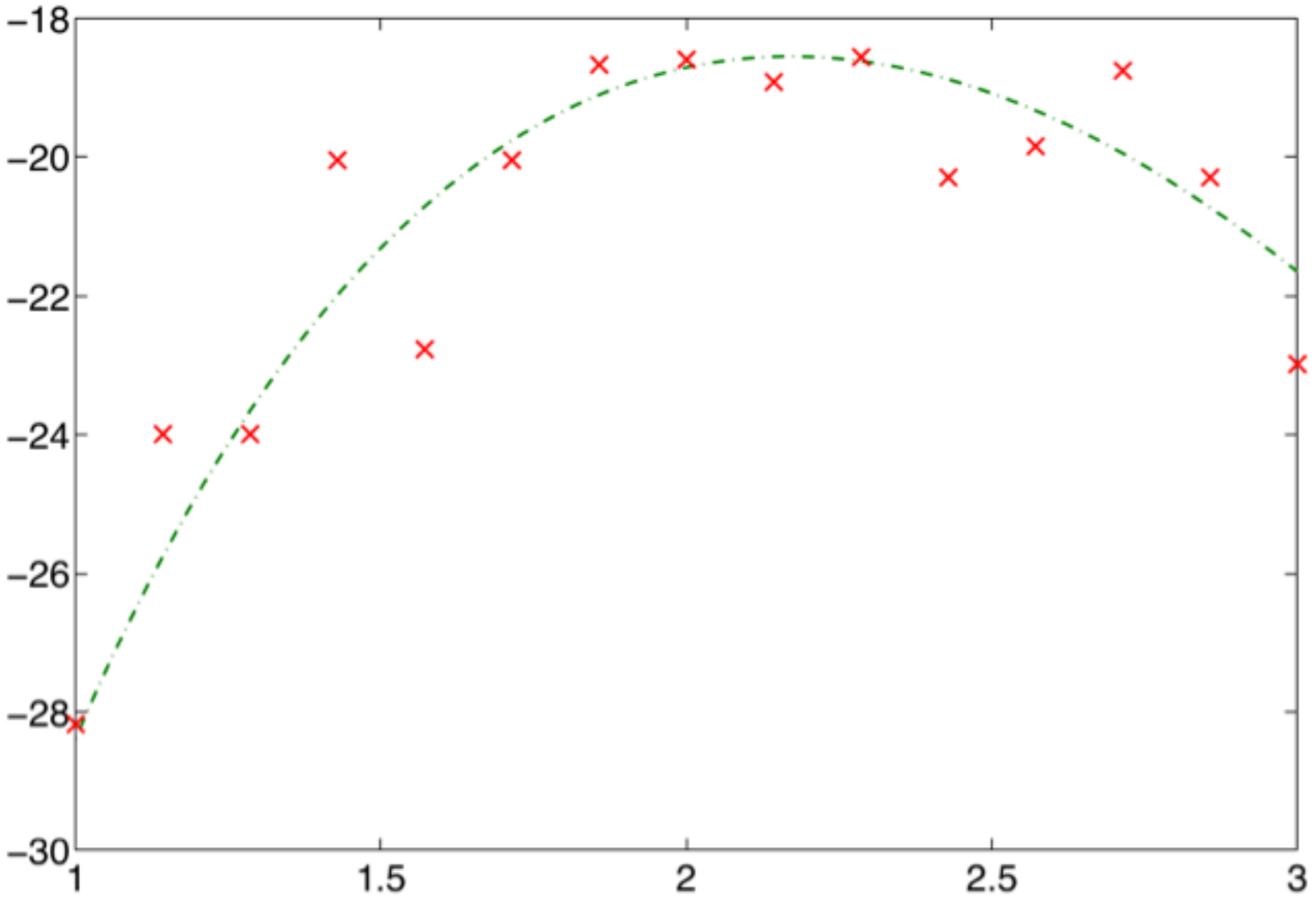


$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

$$\begin{aligned} \mathbf{f}_* | X, \mathbf{y}, X_* &\sim \mathcal{N}(\bar{\mathbf{f}}_*, \operatorname{cov}(\mathbf{f}_*)), \quad \text{where} \\ \bar{\mathbf{f}}_* &\triangleq \mathbb{E}[\mathbf{f}_* | X, \mathbf{y}, X_*] = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y}, \\ \operatorname{cov}(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*) \end{aligned}$$

(1) Sample

Collection of the training set $\{(\theta^{(i)}, y^{(i)}), i = 1,\ldots,m\}$ for parameters values θ.

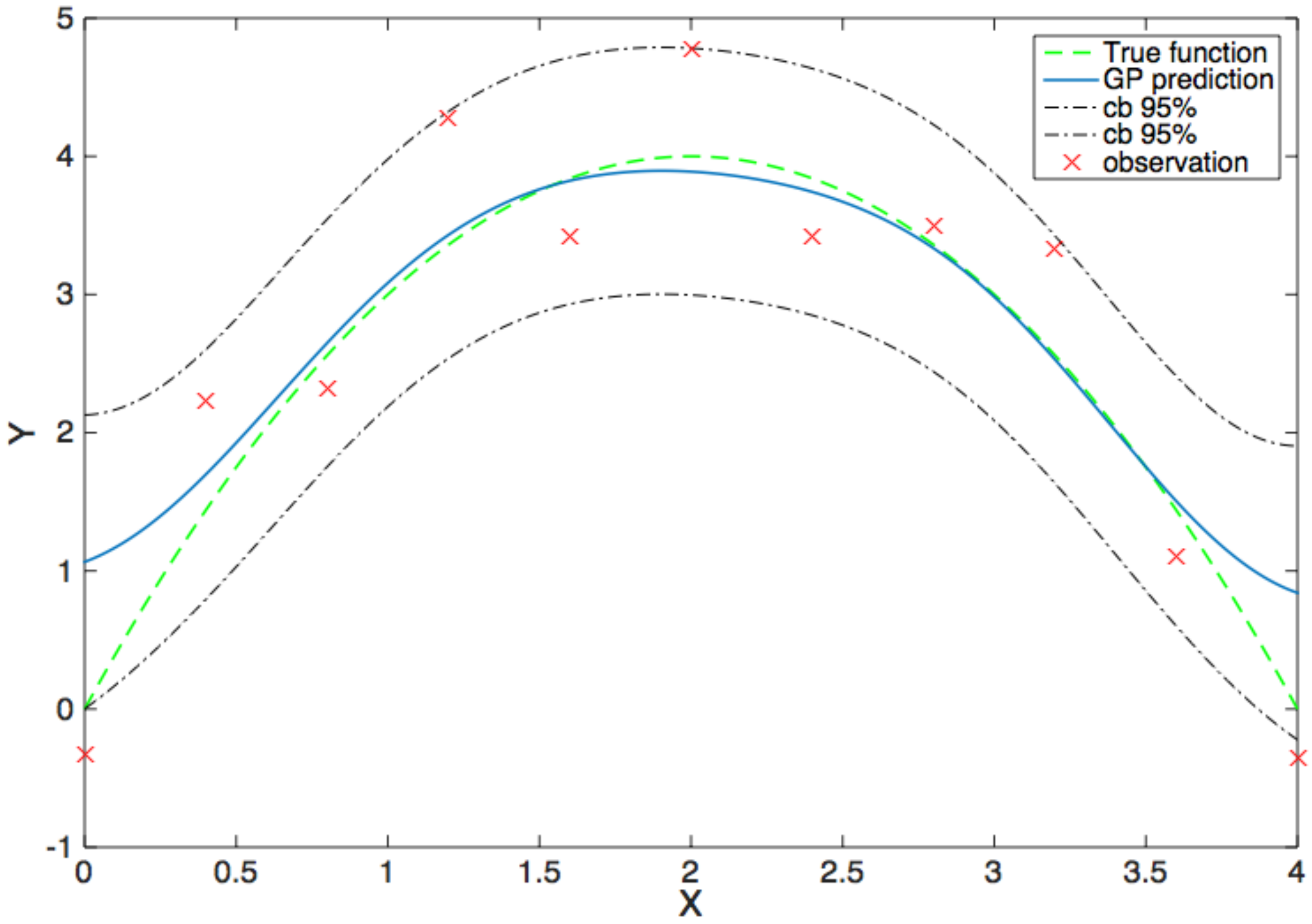


(2) The GP Regression

We have noisy observations y of the function value distributed around an unknown true value f (θ) with spherical Gaussian noise

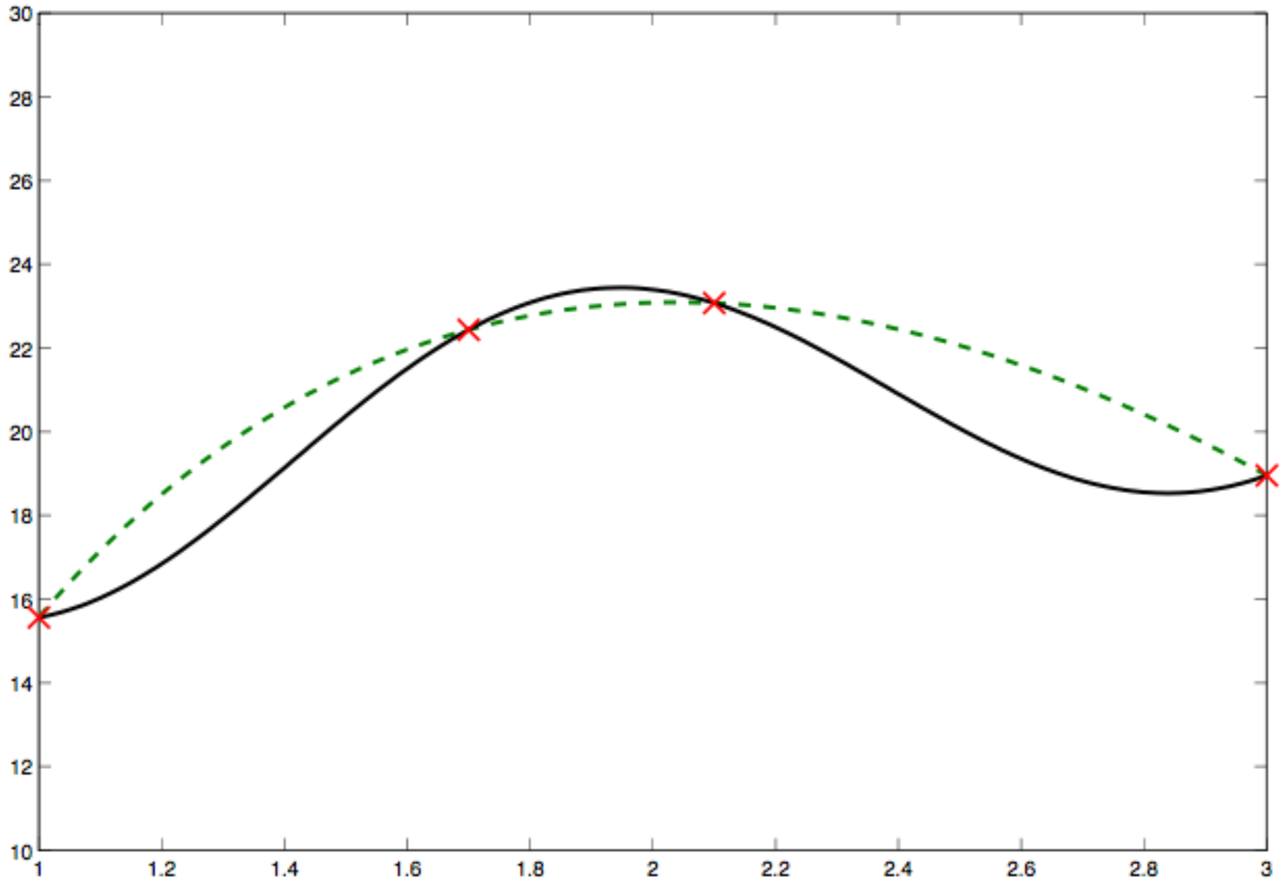
(2) The GP Regression

We have noisy observations y of the function value distributed around an unknown true value f (θ) with spherical Gaussian noise



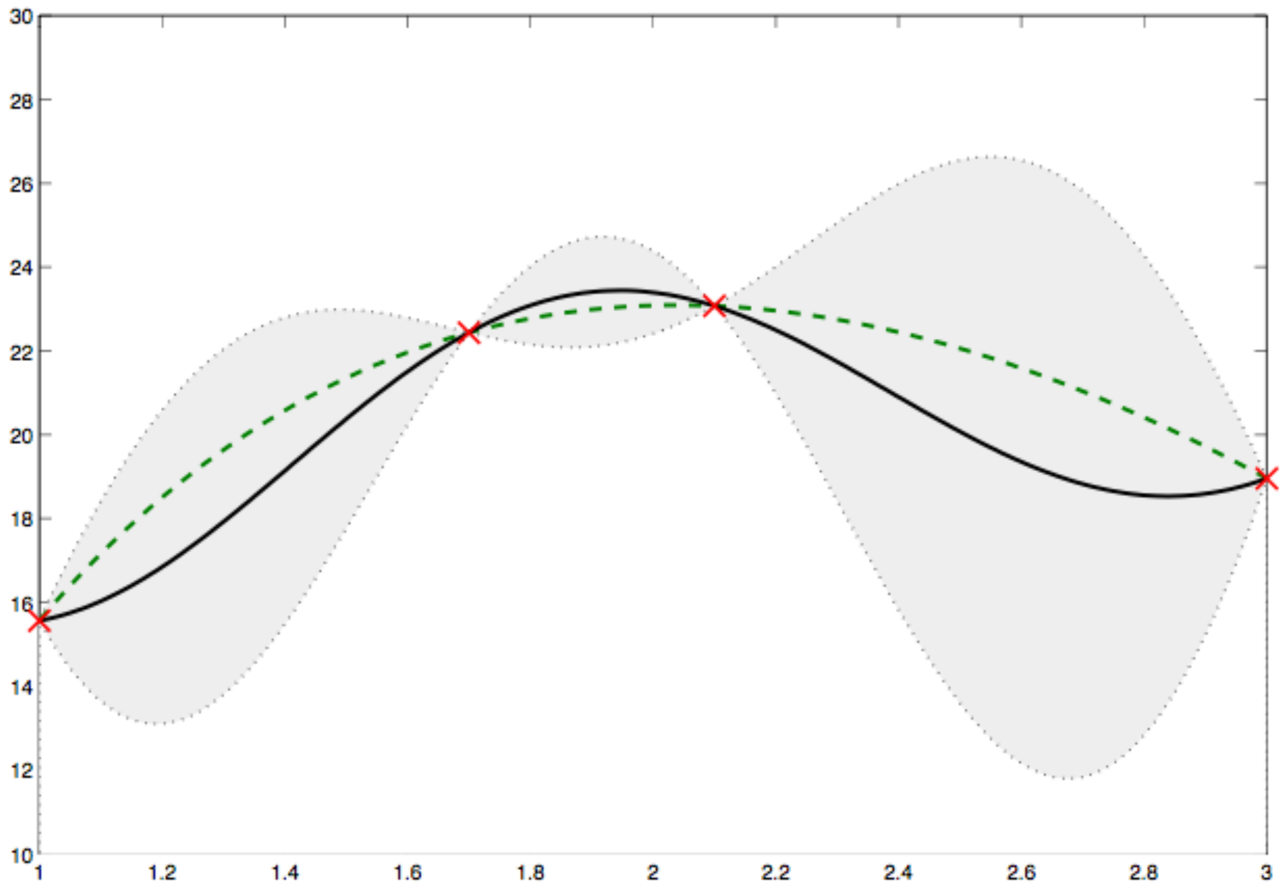
(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



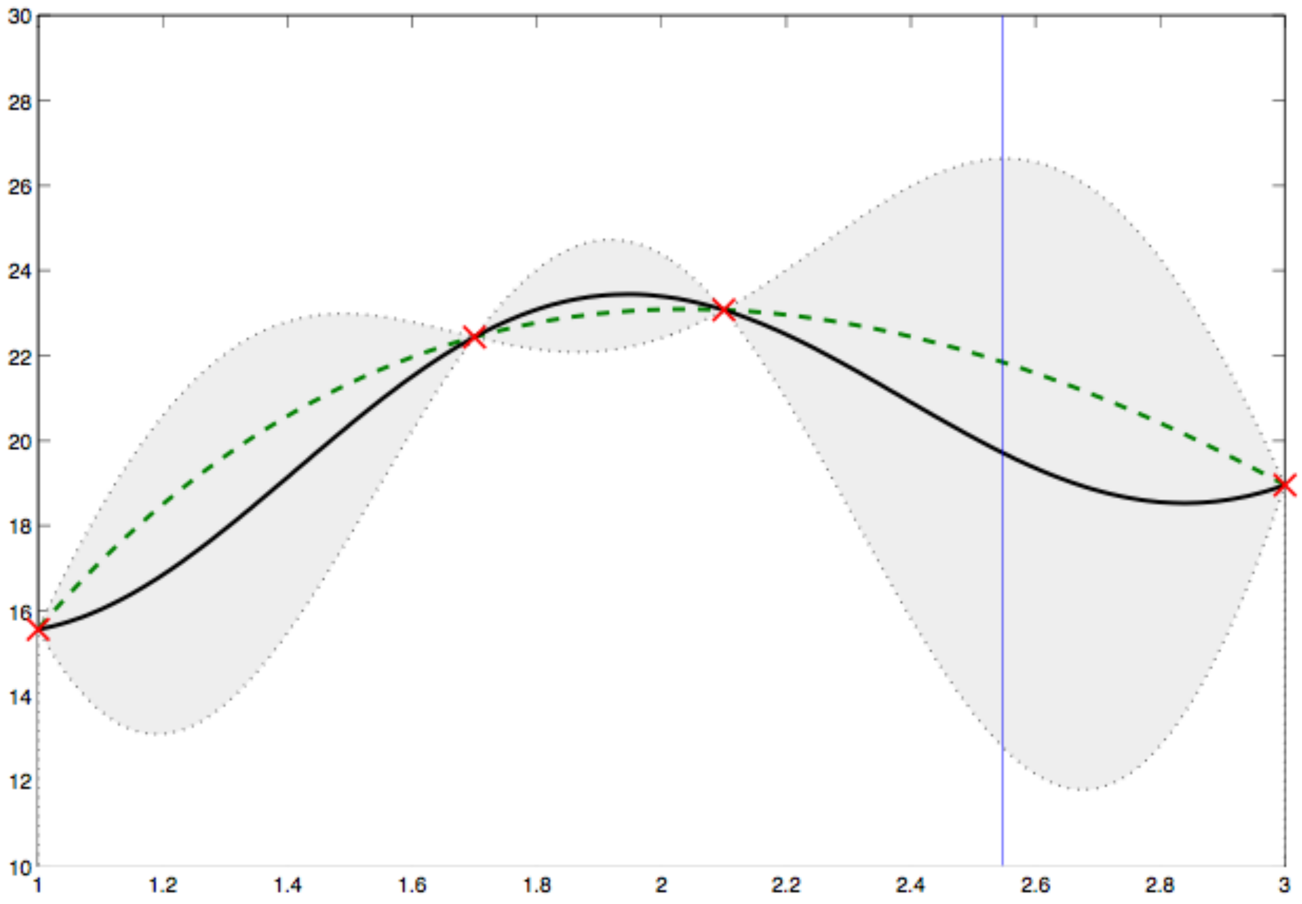
(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



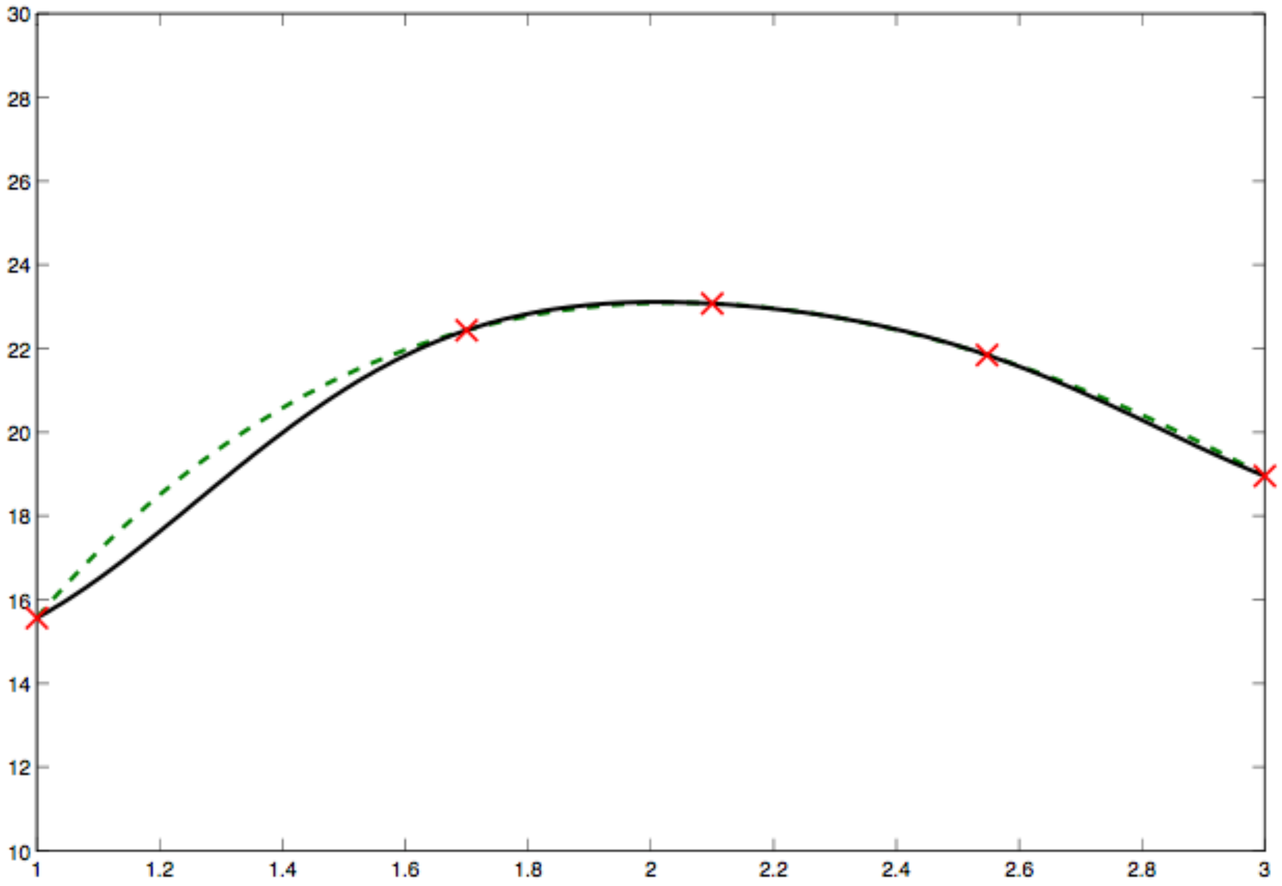
(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



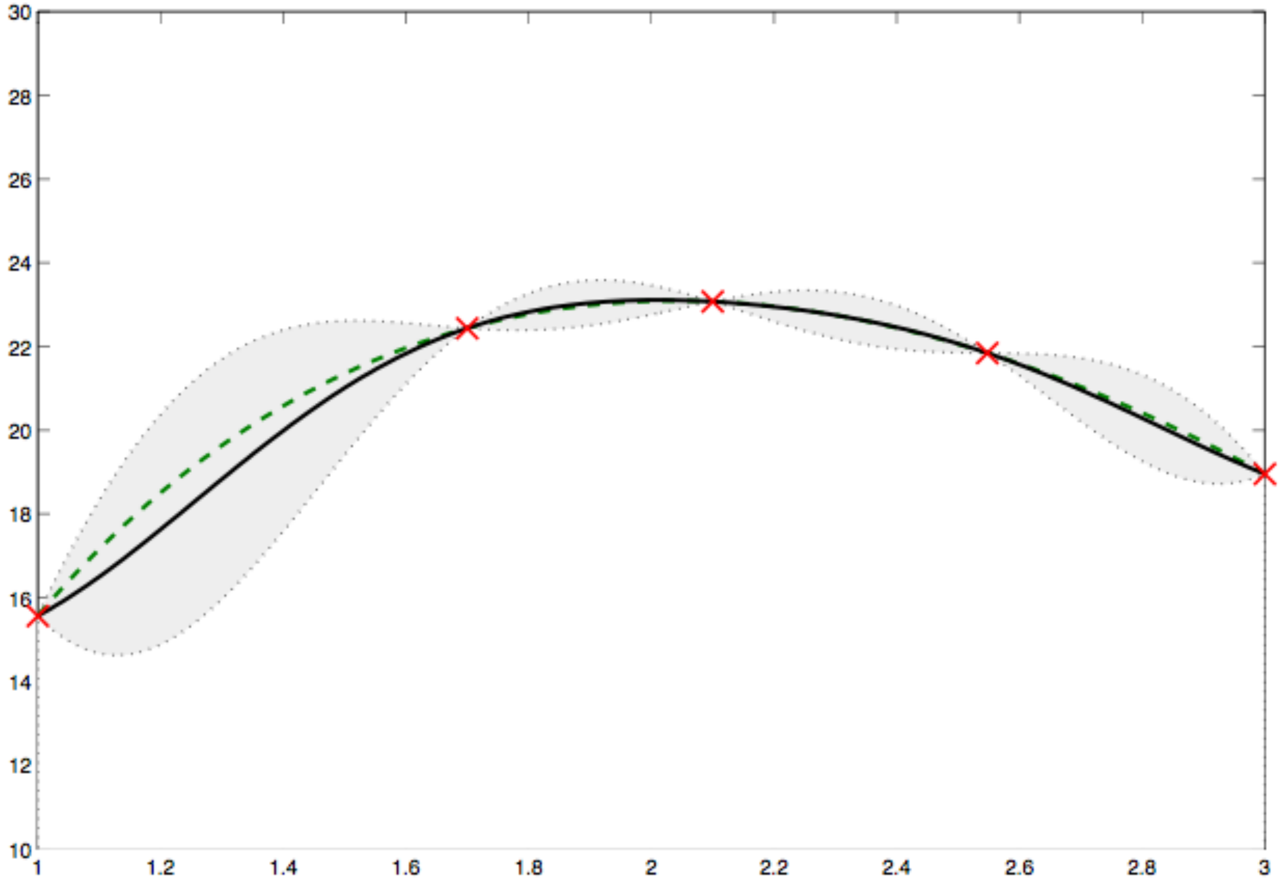
(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t \sqrt{k^*(\theta,\theta)}]$



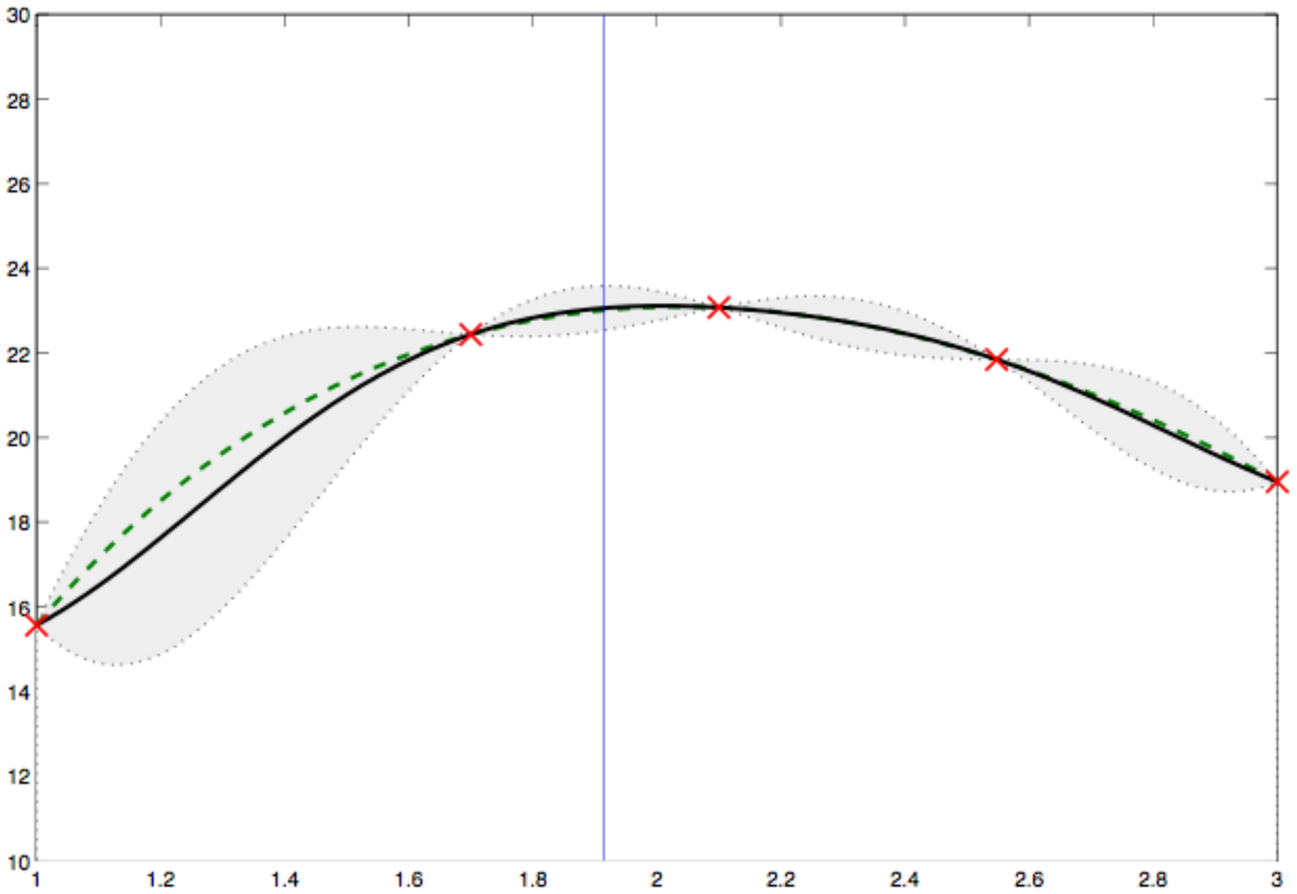
(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t \sqrt{k^*(\theta, \theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the **95% upper quantile of the distribution**: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



Bibliography

Falsification:

- Silvetti S., Policriti A., Bortolussi L. (2017) *An Active Learning Approach to the Falsification of Black Box Cyber-Physical Systems.* IFM 2017. LNCS, vol 10510. Springer, Cham.
- Several excellent papers on the first development of falsification technology can be found on the web-site of S-TaLiRo : https://sites.google.com/a/asu.edu/s-taliro/references
- Jyotirmoy Deshmukh, Marko Horvat, Xiaoqing Jin, Rupak Majumdar, and Vinayak S. Prabhu. 2017. Testing Cyber-Physical Systems through Bayesian Optimization. *ACM Trans. Embed. Comput. Syst.* 16, 5s, Article 170 (September 2017)
- Deshmukh, Jyotirmoy, Xiaoqing Jin, James Kapinski, and Oded Maler. Stochastic Local Search for Falsification of Hybrid Systems. In *International Symposium on Automated Technology for Verification and Analysis*, pp. 500-517.

Parameter Synthesis:

- Ezio Bartocci, Luca Bortolussi, Laura Nenzi, Guido Sanguinetti, System design of stochastic models using robustness of temporal properties. Theor. Comput. Sci. 587: 3-25 (2015)
- Bortolussi L., Silvetti S. (2018*) Bayesian Statistical Parameter Synthesis for Linear Temporal Properties of Stochastic Models.* TACAS 2018. LNCS, vol 10806. Springer, Cham