Pareto Front Exploration for Parametric Temporal Logic Specifications of Cyber-Physical Systems

Bardh Hoxha and Georgios Fainekos

1st Workshop on Monitoring and Testing of Cyber-Physical Systems

Vienna, Austria – April 2016 – CPS Week

School of Computing, Informatics and
Decision System Engineering

Arizona State University

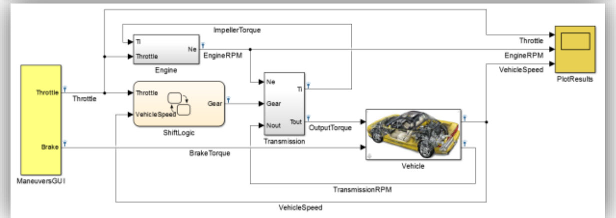
bhoxha at asu edu

http://www.public.asu.edu/~bhoxha

Parameter Mining

System Σ

$$y=\Delta(x_0,u) \qquad x_0 \in X_0 \qquad u \in \mathbb{U}$$

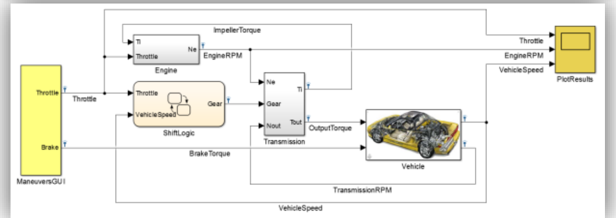


Parameter Mining

What is the shortest time that the engine speed can exceed 3200RPM?

System Σ

$y=\Delta(x_0,u)$ $x_0 \in X_0$, $u \in \mathbb{U}$



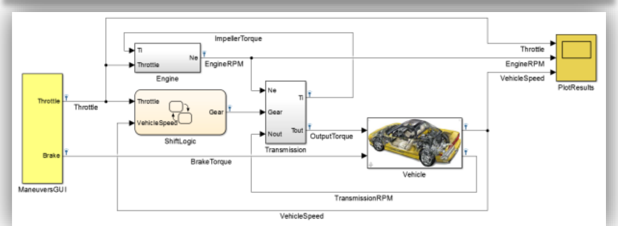
Parameter Mining

What is the shortest time that the engine speed can exceed 3200RPM?

The vehicle speed is always less than parameter θ_1 and the engine speed is always less than θ_2.

System Σ

$y=\Delta(x_0,u)$ $x_0 \in X_0$ $u \in \mathbb{U}$



Parameter Mining

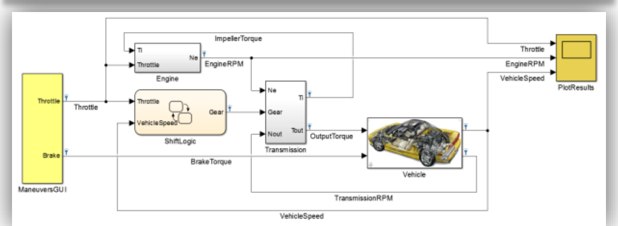
What is the shortest time that the engine speed can exceed 3200RPM?

The vehicle speed is always less than parameter θ_1 and the engine speed is always less than θ_2.

If I increase/decrease θ_1 by a specific amount, how much do I have to in crease/decrease θ_2 so that the system satisfies the specification?”

System Σ

$$y = \Delta(x_0, u) \qquad x_0 \in X_0 \qquad u \in \mathbb{U}$$



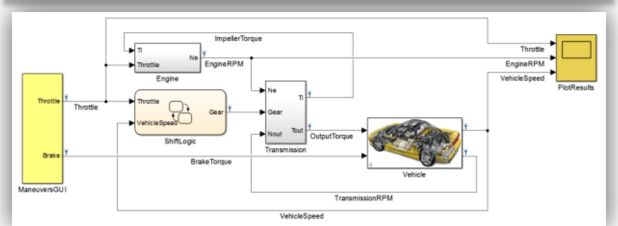
Parameter Mining

What is the shortest time that the engine speed can exceed 3200RPM?

The vehicle speed is always less than parameter θ_1 and the engine speed is always less than θ_2.

System Σ

$y = \Delta(x_0, u)$ $x_0 \in X_0$ $u \in \mathbb{U}$

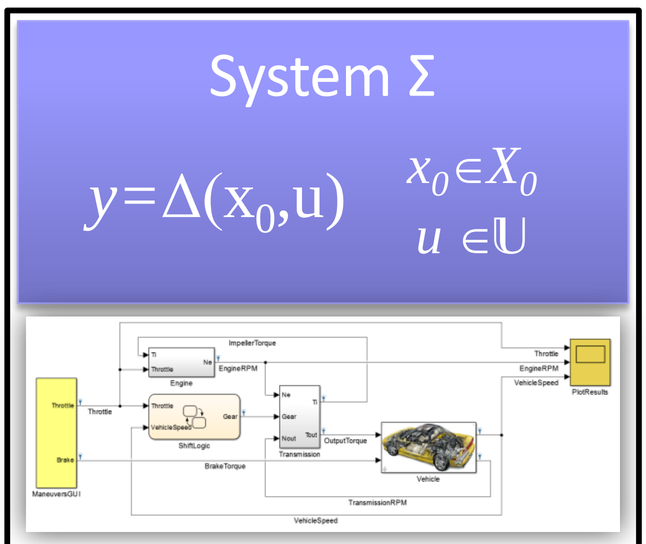


If I increase/decrease θ_1 by a specific amount, how much do I have to in crease/decrease θ_2 so that the system satisfies the specification?”

Parameter Mining

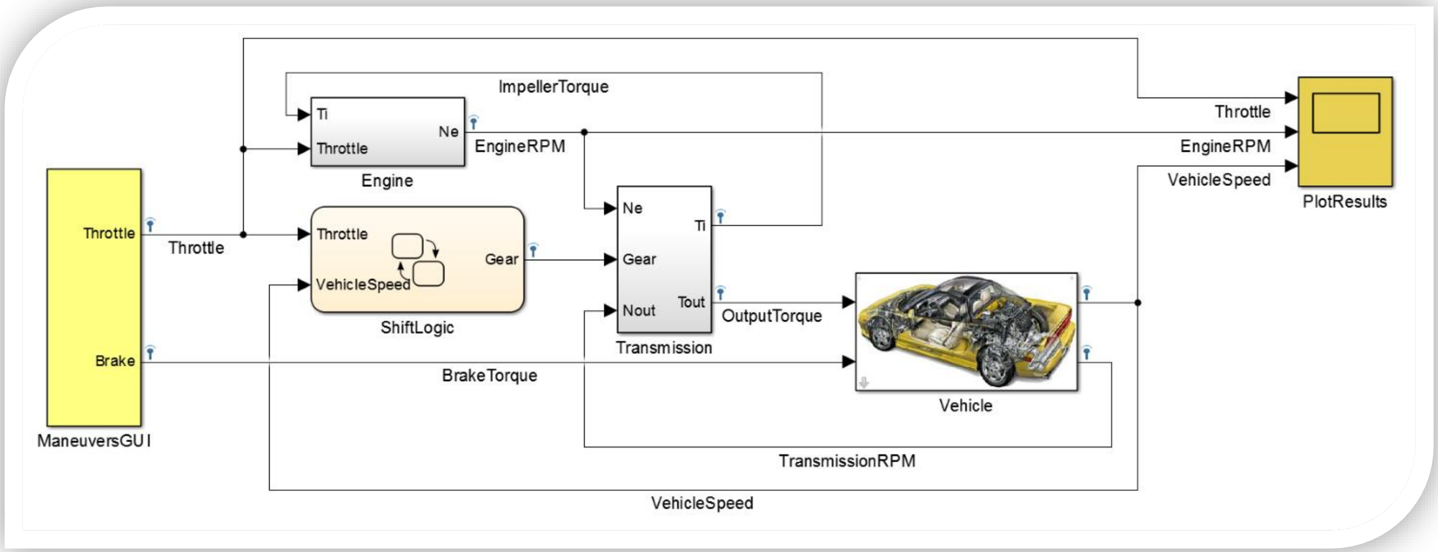
Benefits:

- Facilitate the development of system specifications
  - In many cases, system requirements are not well formalized by the initial system design stages
- Explore and determine system properties
  - If a specification can be falsified, then it is natural to inquire for the range of parameter values that cause falsification.



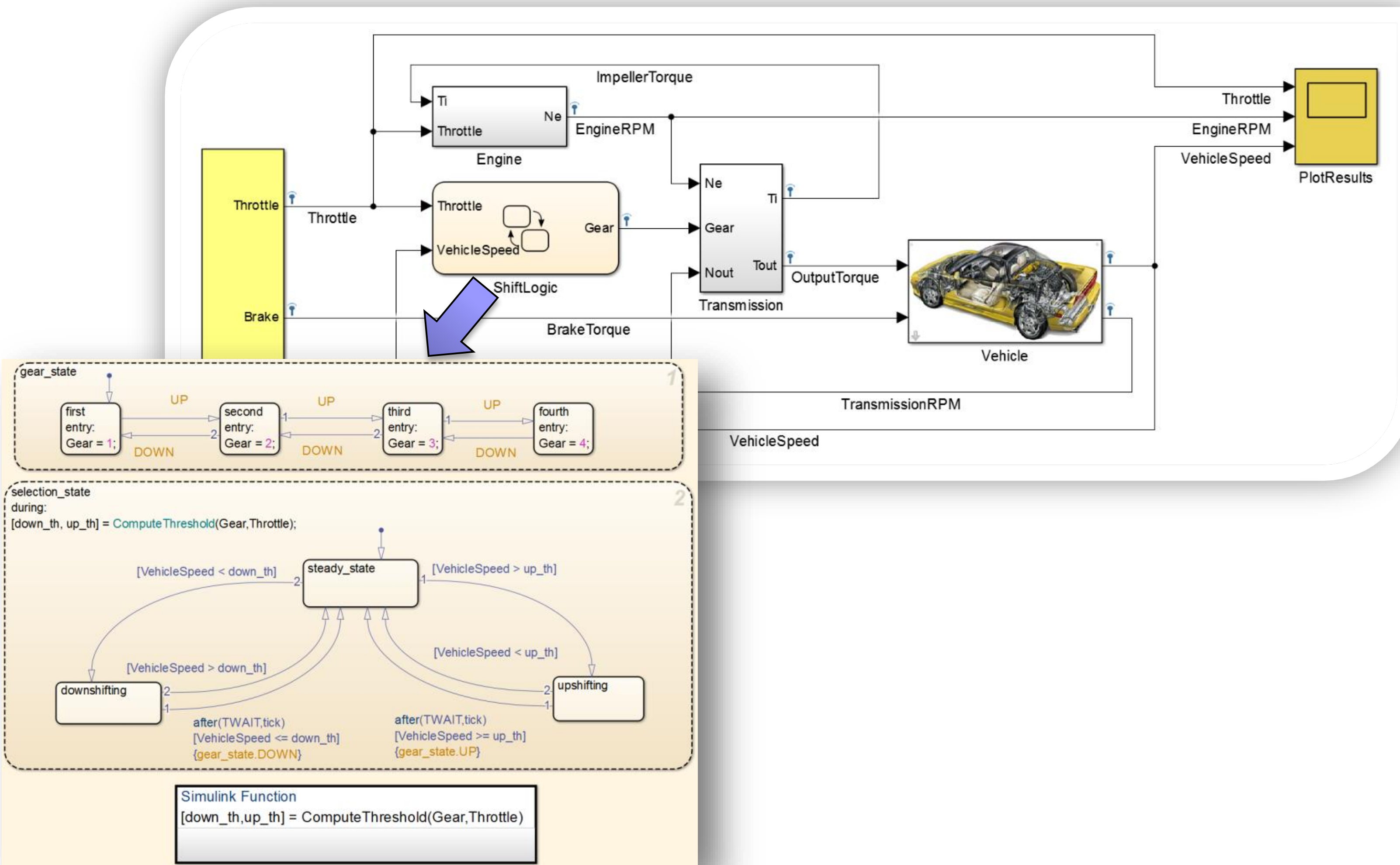
Preliminaries – Running Example

Automotive Transmission Simulink Demo



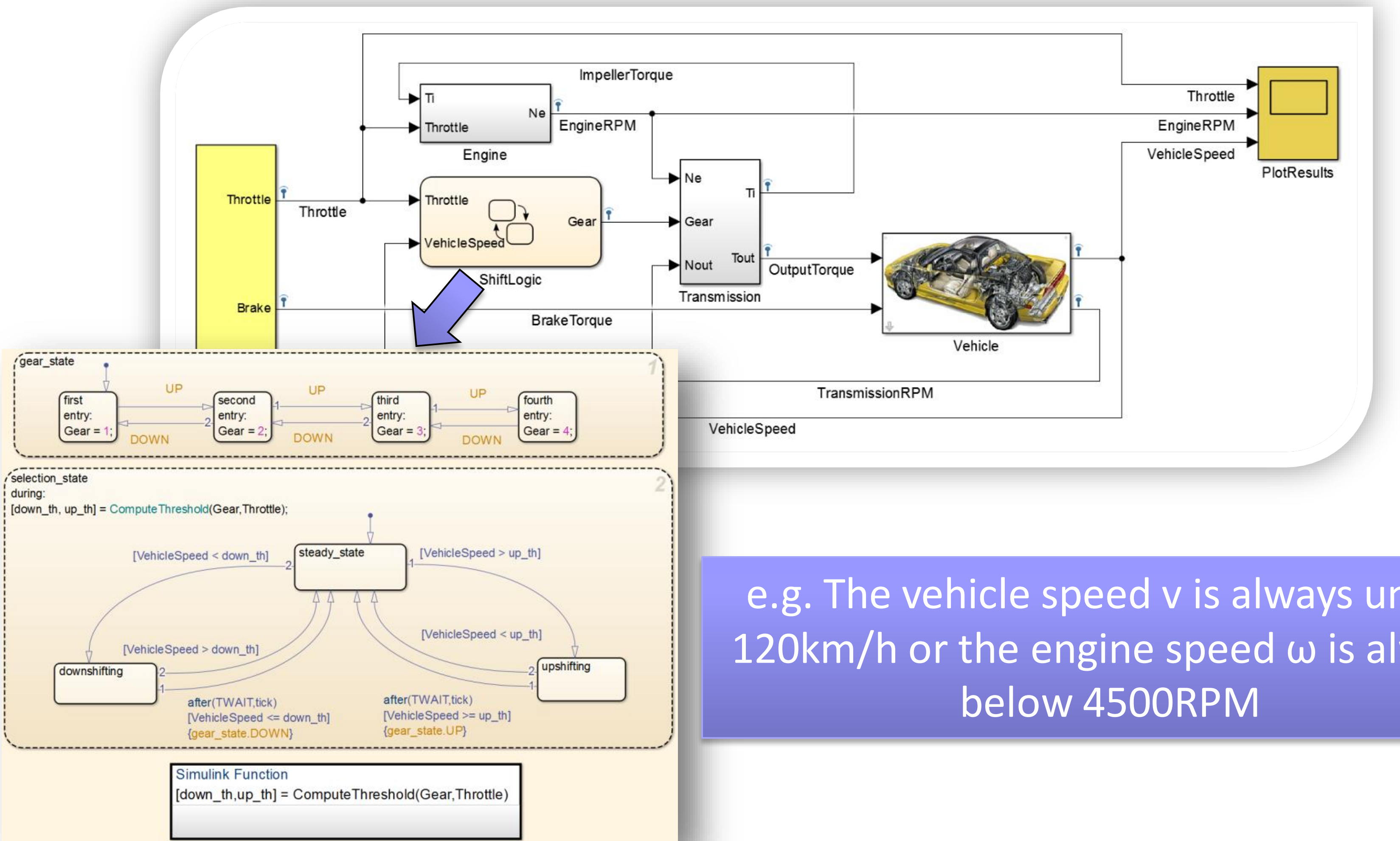
Preliminaries – Running Example

Automotive Transmission Simulink Demo



Preliminaries – Running Example

Automotive Transmission Simulink Demo



e.g. The vehicle speed v is always under 120km/h or the engine speed ω is always below 4500RPM

Preliminaries - Metric Temporal Logic

Syntax: Boolean connectives with temporal operators

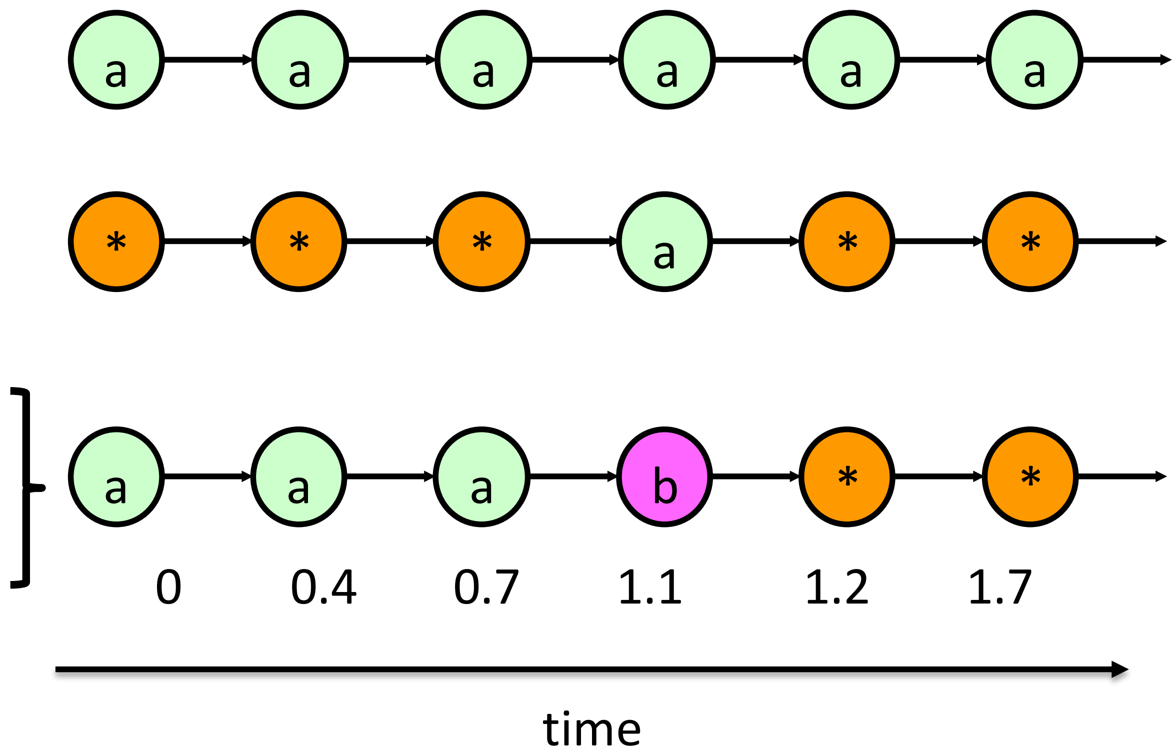
$$\phi \ ::= \top \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid G\ \phi \mid F\ \phi \mid \phi_1 U_I \phi_2$$

$G\ a$ - always a

$F\ a$ - eventually a

$a\ U\ b$ - a until b

$a\ U_{[1,1.5]}\ b$ - a until b



Other notation: $Ga \equiv \Box a$ and $Fa \equiv \Diamond a$

Parameter Mining

The vehicle speed is always less than parameter θ_1 and the engine speed is always less than θ_2.

Parametric MTL: $\phi_1[\vec{\theta}] = \Box((v \leq \theta_1) \wedge (\omega \leq \theta_2))$

PMTL formulas may contain state and/or timing parameters

Ex. $\phi_2[\vec{\theta}] = \neg(\Diamond_{[0,\theta_1]}(v > 100) \wedge (\omega \leq \theta_2))$

Timing

State

Parameter Mining

Parameter Mining Problem:

Given a parametric MTL formula $\phi[\vec{\theta}]$ with a vector of m unknown parameters and a system Σ, find the set $\Psi = \{\theta^* \in \Theta \mid \Sigma \nvDash \phi[\theta^*]\}$

Parameter Mining

Parameter Mining Problem:

Given a parametric MTL formula $\phi[\vec{\theta}]$ with a vector of m unknown parameters and a system Σ, find the set $\Psi = \{\theta^* \in \Theta \mid \Sigma \not\models \phi[\theta^*]\}$

Question:

Why don't we search for the set of parameters for which the system satisfies the specification?

Parameter Mining

Parameter Mining Problem:

Given a parametric MTL formula $\phi[\vec{\theta}]$ with a vector of m unknown parameters and a system Σ, find the set $\Psi = \{\theta^* \in \Theta \mid \Sigma \not\models \phi[\theta^*]\}$

Approximation possible ☺

Question:

Why don't we search for the set of parameters for which the system satisfies the specification?

Problem is undecidable [AL94] ☹.

[AL94]: Alur, Rajeev, et al. "The algorithmic analysis of hybrid systems." *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*. Springer Berlin Heidelberg, 1994.

Parameter Mining

Testing framework based on the theory of robustness of MTL formulas

Monotonicity properties of parametric MTL formulas.

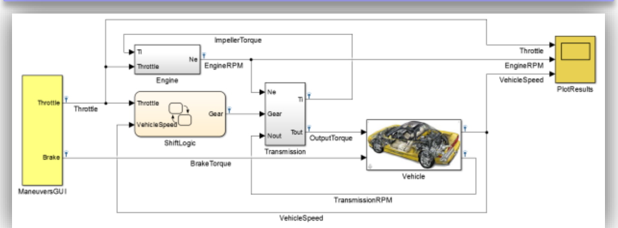
Parameter mining -> Optimization problem

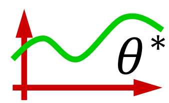
Output Trajectory Testing

For a specific parameter valuation θ^*:

System Σ

$y=\Delta(x_0,u)$ $x_0 \in X_0$ $u \in \mathbb{U}$





Output y

$\phi[\theta^*]$

Tester

$y \nvDash \phi[\theta^*]$?

If yes, we have a counterexample

proof that Σ does not satisfy $\phi[\theta^*]$

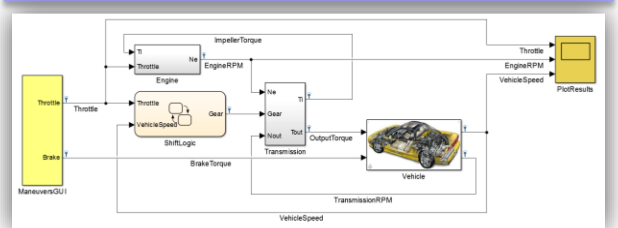
If no, generate another output

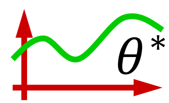
Output Trajectory Testing

For a specific parameter valuation θ^*:

System Σ

$y=\Delta(x_0,u)$ $x_0 \in X_0$, $u \in \mathbb{U}$





Output y

$\phi[\theta^*]$

Tester

$y \not\models \phi[\theta^*]$?

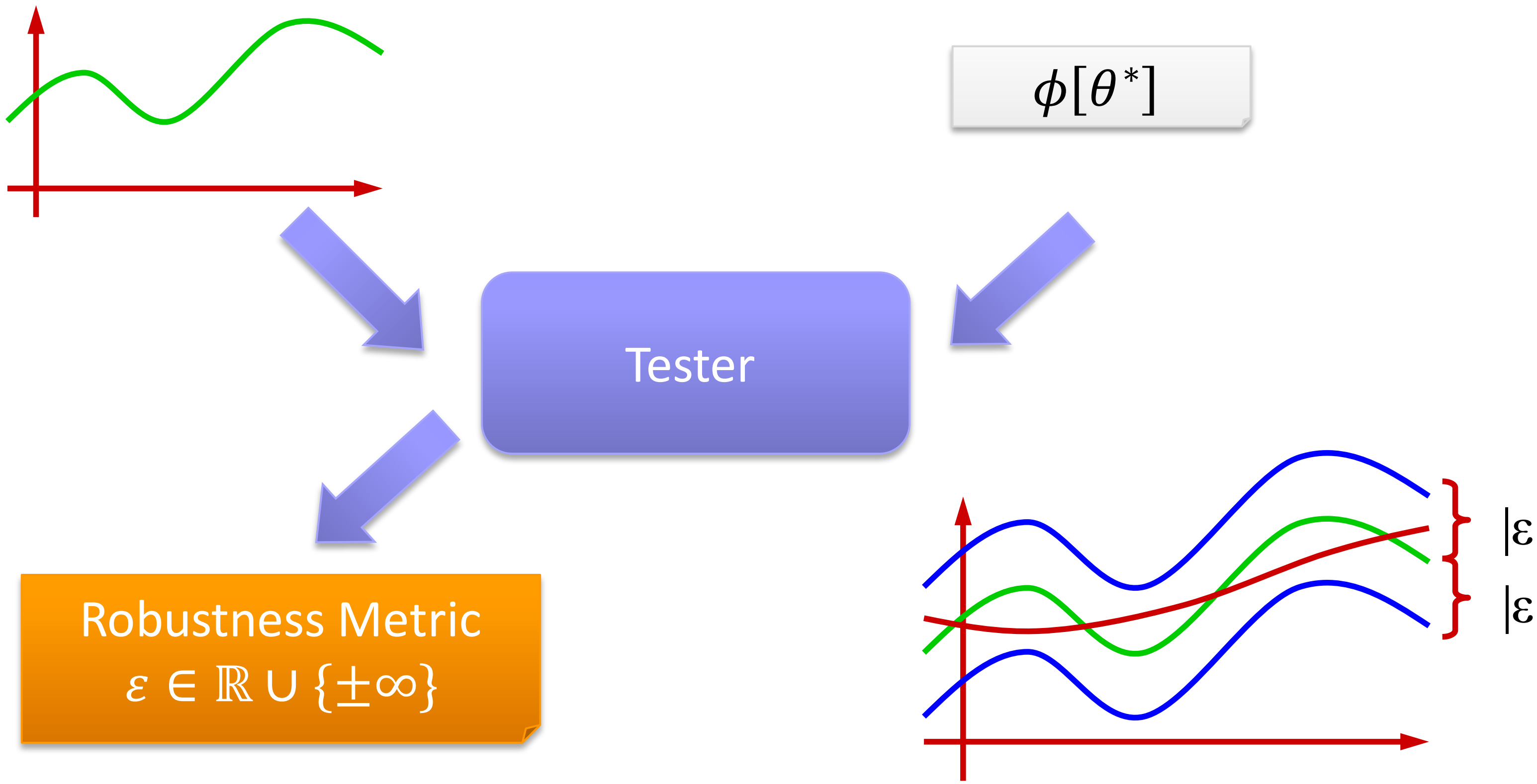
By how much?

If yes, we have a counterexample

proof that Σ does not satisfy $\phi[\theta^*]$

If no, generate another output

Robustness of Temporal Logics



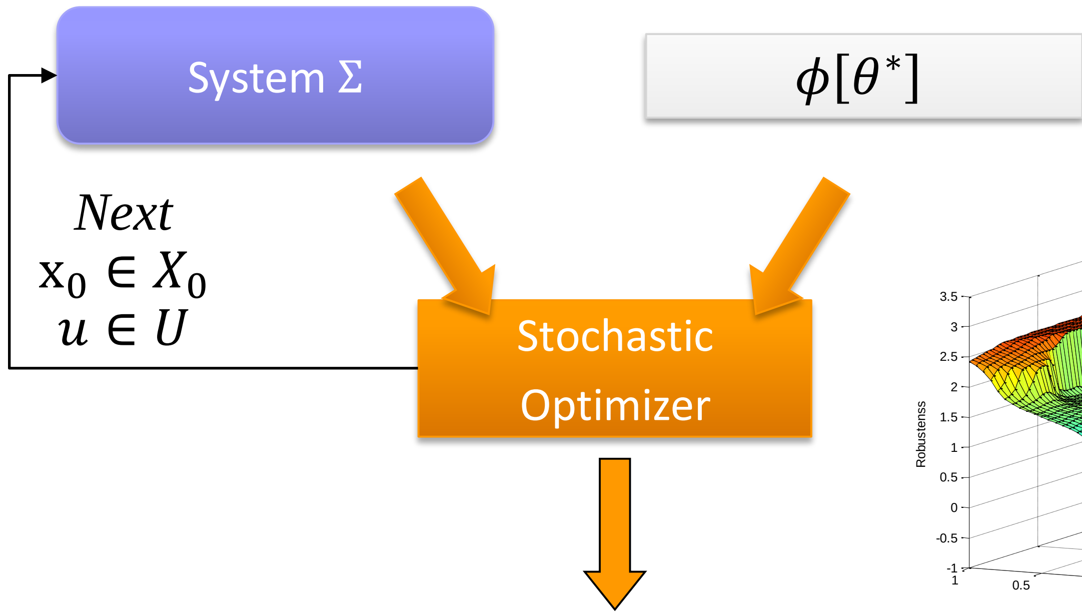
positive robustness → signal satisfies the formula

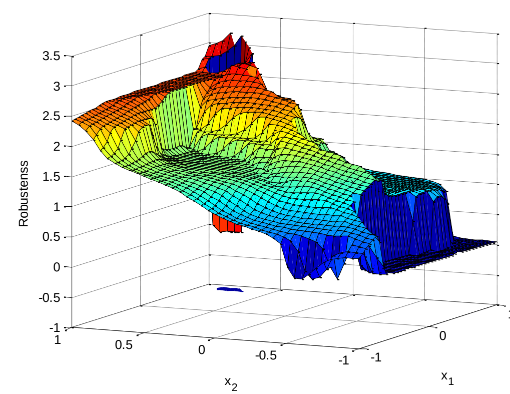
negative robustness → signal falsifies the formula

Fainekos and Pappas, *Robustness of temporal logic specifications for continuous-time signals,* Theoretical Computer Science, 2009

Falsification by optimization

The falsification method searches for counterexamples that prove that the system does not satisfy the specification





Minimum Robustness

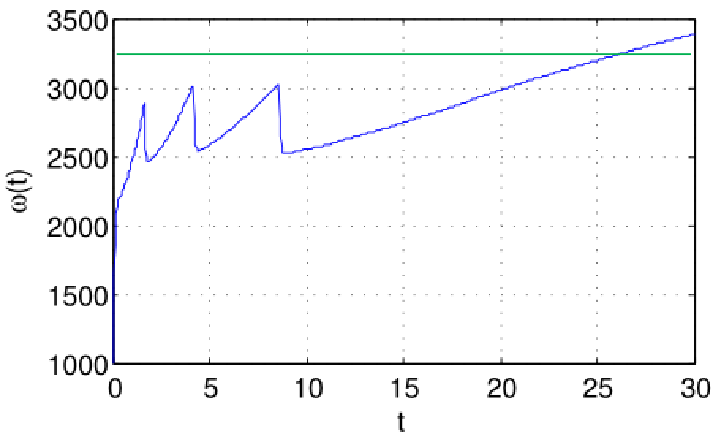
with corresponding input signal and initial conditions

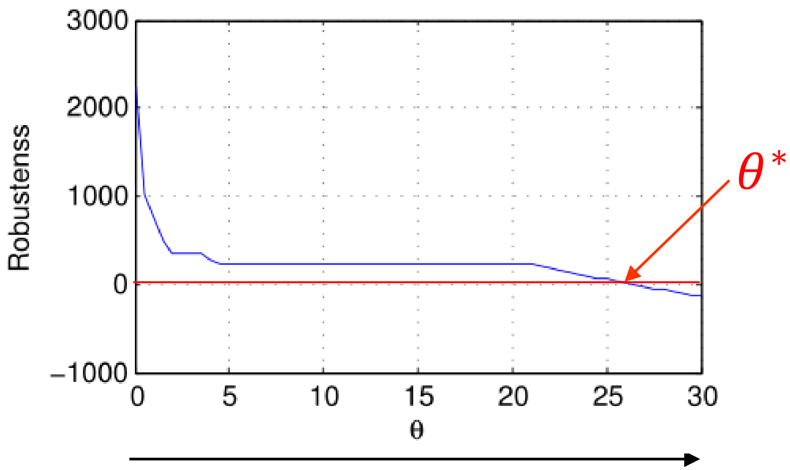
Abbas, et al, Probabilistic Temporal Logic Falsification of Cyber-Physical Systems, ACM TECS 2013

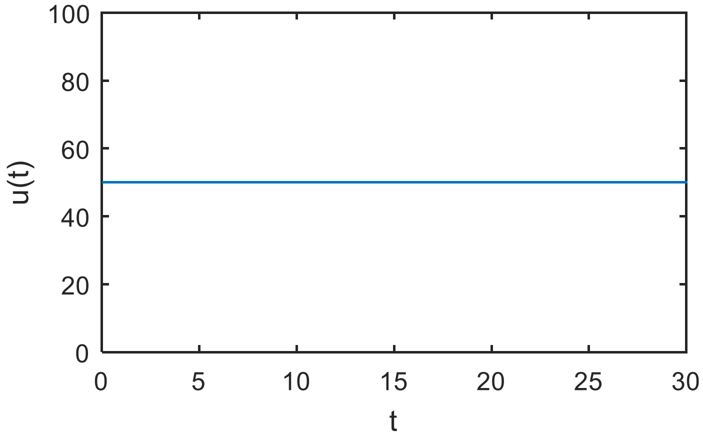
Monotonicity of parametric MTL specifications

NL: Always, from 0 to θ, the engine speed is less than 3250

$$\phi[\theta] = \Box_{[0,\theta]}(\omega \leq 3250)$$







As we increase θ, we can only increase the opportunity to find falsifying system behavior

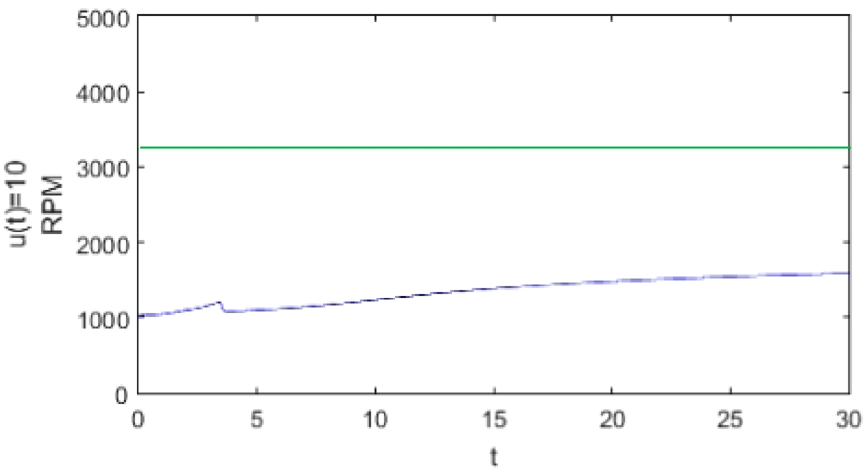
Non-Increasing robustness with respect to θ

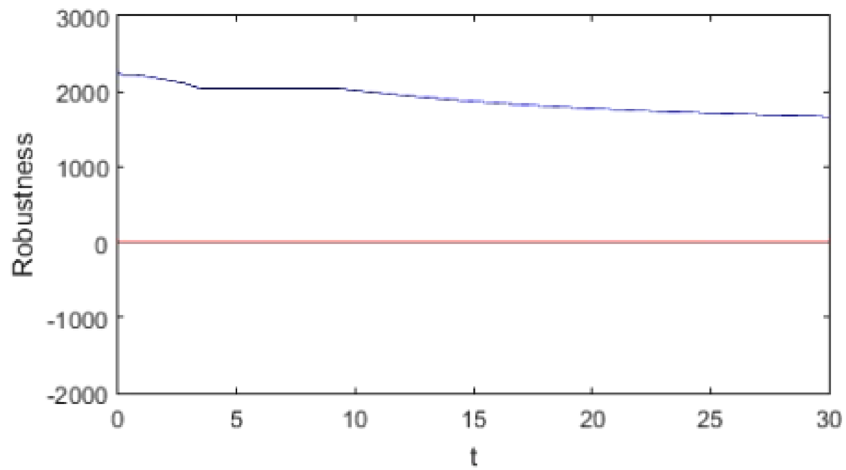
Arizona State University

CPSLab

Monotonicity of parametric MTL specifications

$$\phi[\theta] = \Box_{[0,\theta]}(\omega \leq 3250)$$



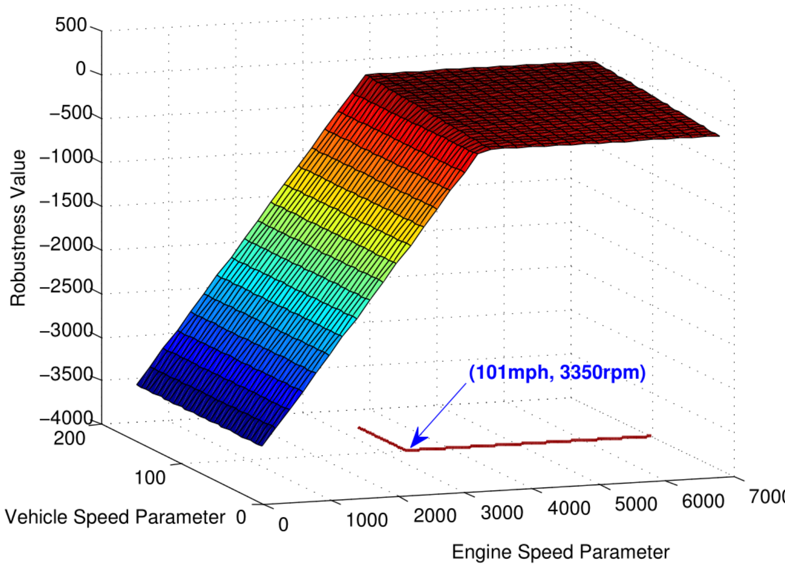


Monotonicity results formalized in
[Hoxha, Dokhanchi, and Fainekos, arXiv:1512.07956]

Monotonicity of parametric MTL specifications

NL: Always, vehicle speed is less than θ_1 and engine speed is less than θ_2

$$\phi_1[\theta] = \Box((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$



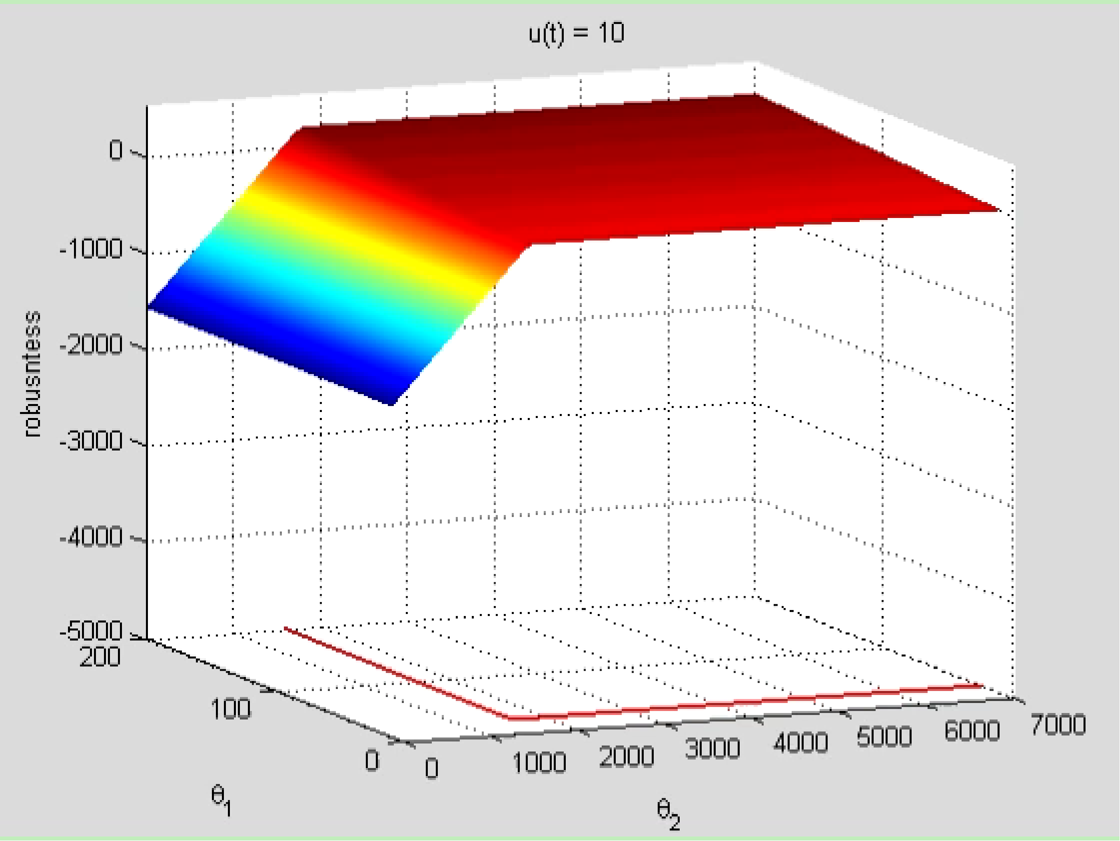
As we increase θ_1 and θ_2, we can only decrease the opportunity to find falsifying system behavior

Non-Decreasing robustness with respect to $f(\vec{\theta})$

Monotonicity results formalized in
[Hoxha, Dokhanchi, and Fainekos, arXiv:1512.07956]

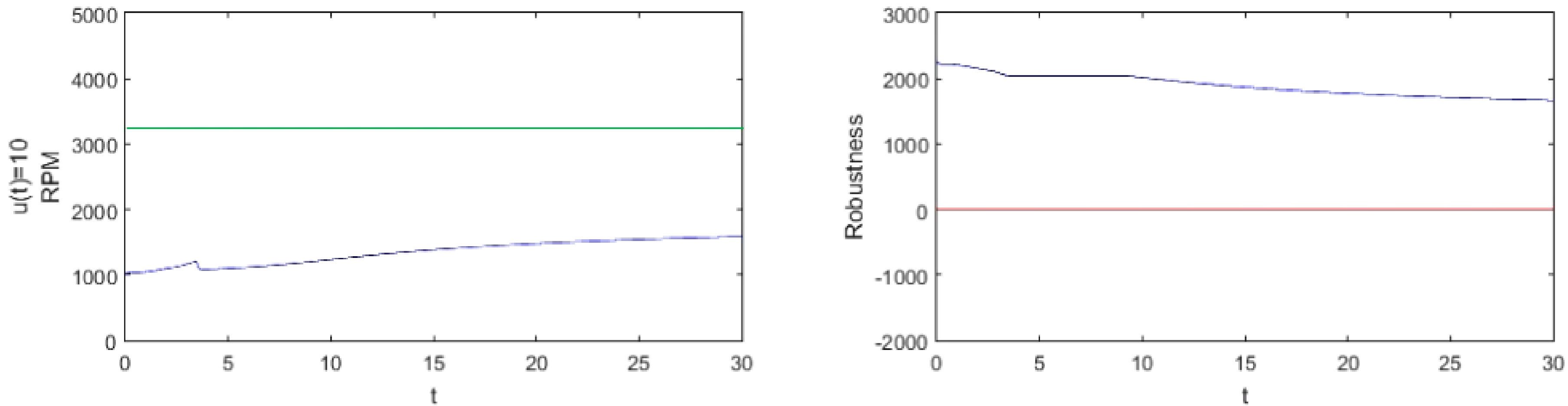
Monotonicity of parametric MTL specifications

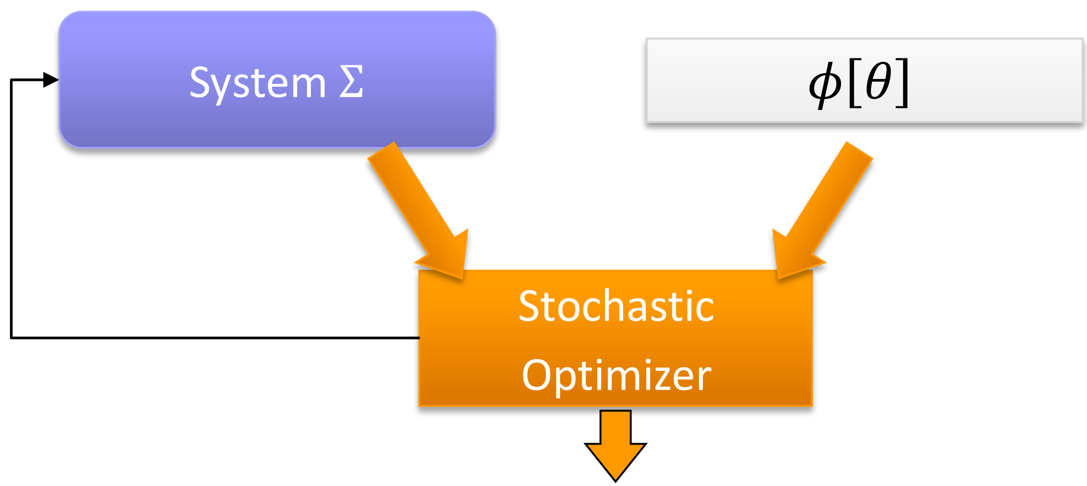
$$\phi_1[\theta] = \Box((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$



Monotonicity of parametric MTL specifications

$$\phi[\theta] = \square_{[0,\theta]}(\omega \leq 3250)$$

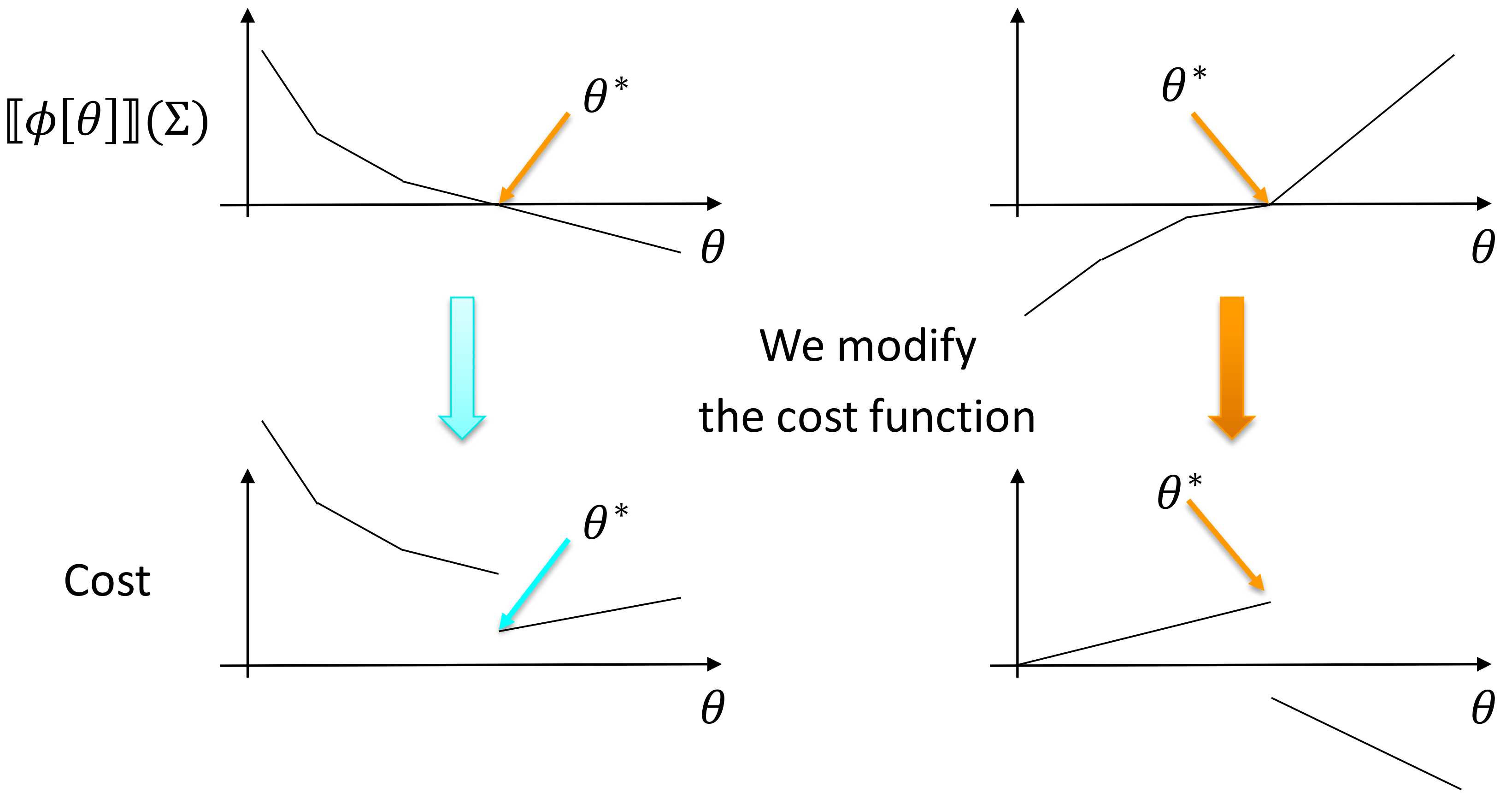




Solution to the Parameter Mining Problem.

Namely, set $\Psi = \{\theta^* \in \Theta \mid \Sigma \nvDash \phi[\theta^*]\}$

Parameter Bound Computation

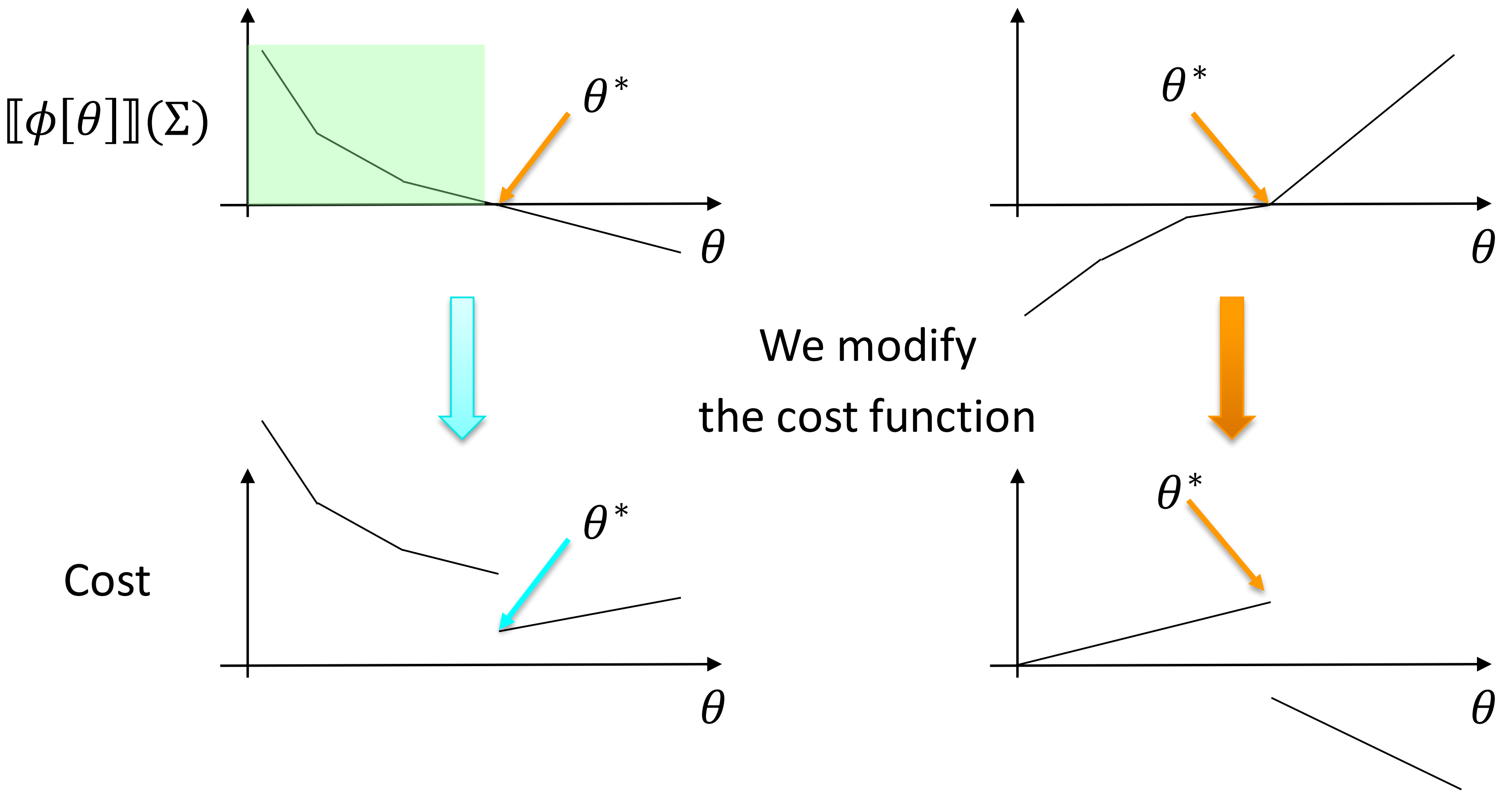


Non-Increasing robustness with respect to θ	Non-Decreasing robustness with respect to θ
Minimize	**Maximize**

$$\min_{\theta\in\Theta}\ \min_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma+[\![\phi[\theta]]\!](\mu) & \\ & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

$$\max_{\theta\in\Theta}\ \max_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma-[\![\phi[\theta]]\!](\mu) & \\ & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

Parameter Bound Computation



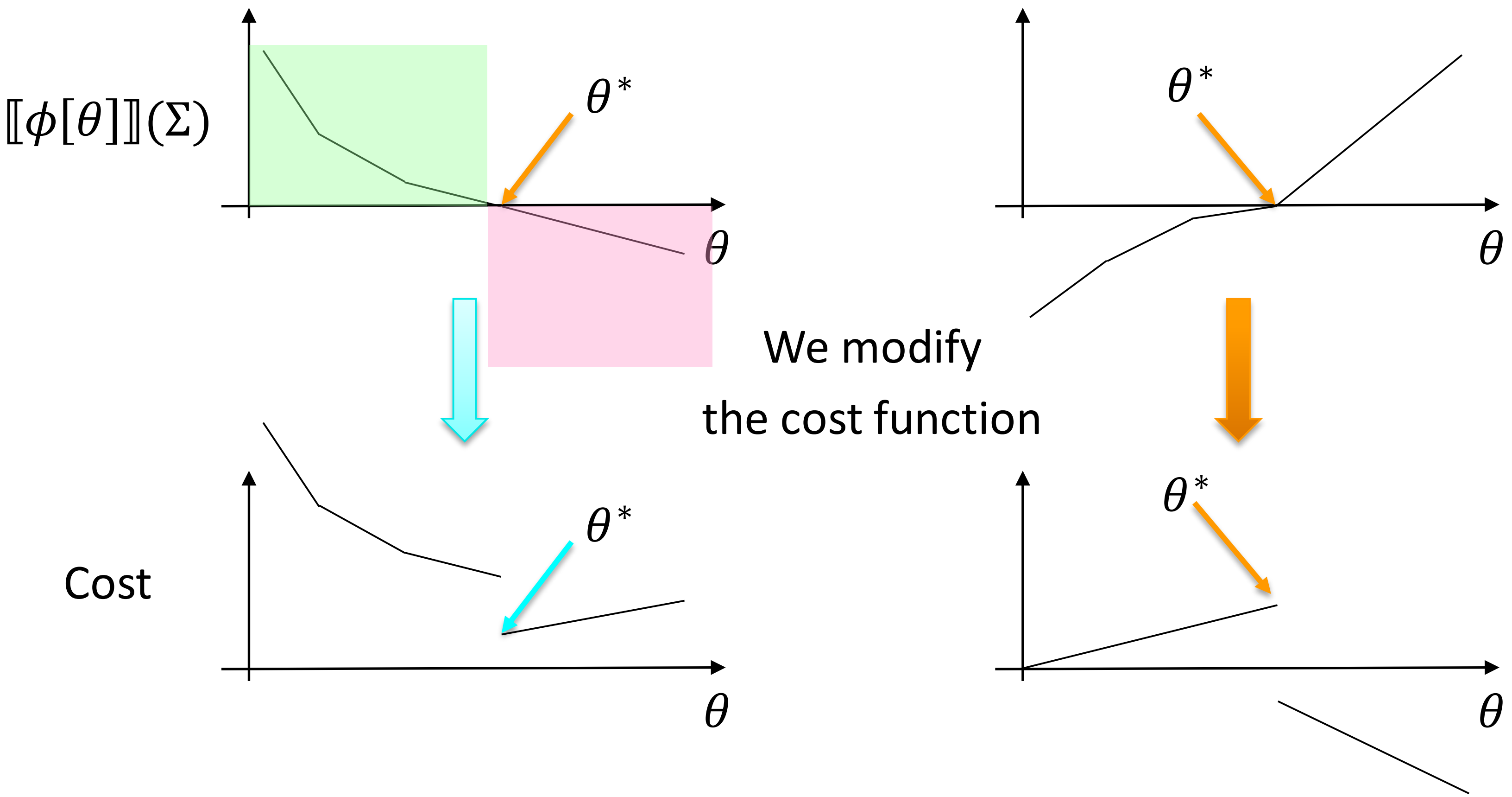
We modify the cost function

Non-Increasing robustness with respect to θ	Non-Decreasing robustness with respect to θ
Minimize	Maximize

$$\min_{\theta\in\Theta}\ \min_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma+[\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

$$\max_{\theta\in\Theta}\ \max_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma-[\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

Parameter Bound Computation



Non-Increasing robustness with respect to θ

Minimize

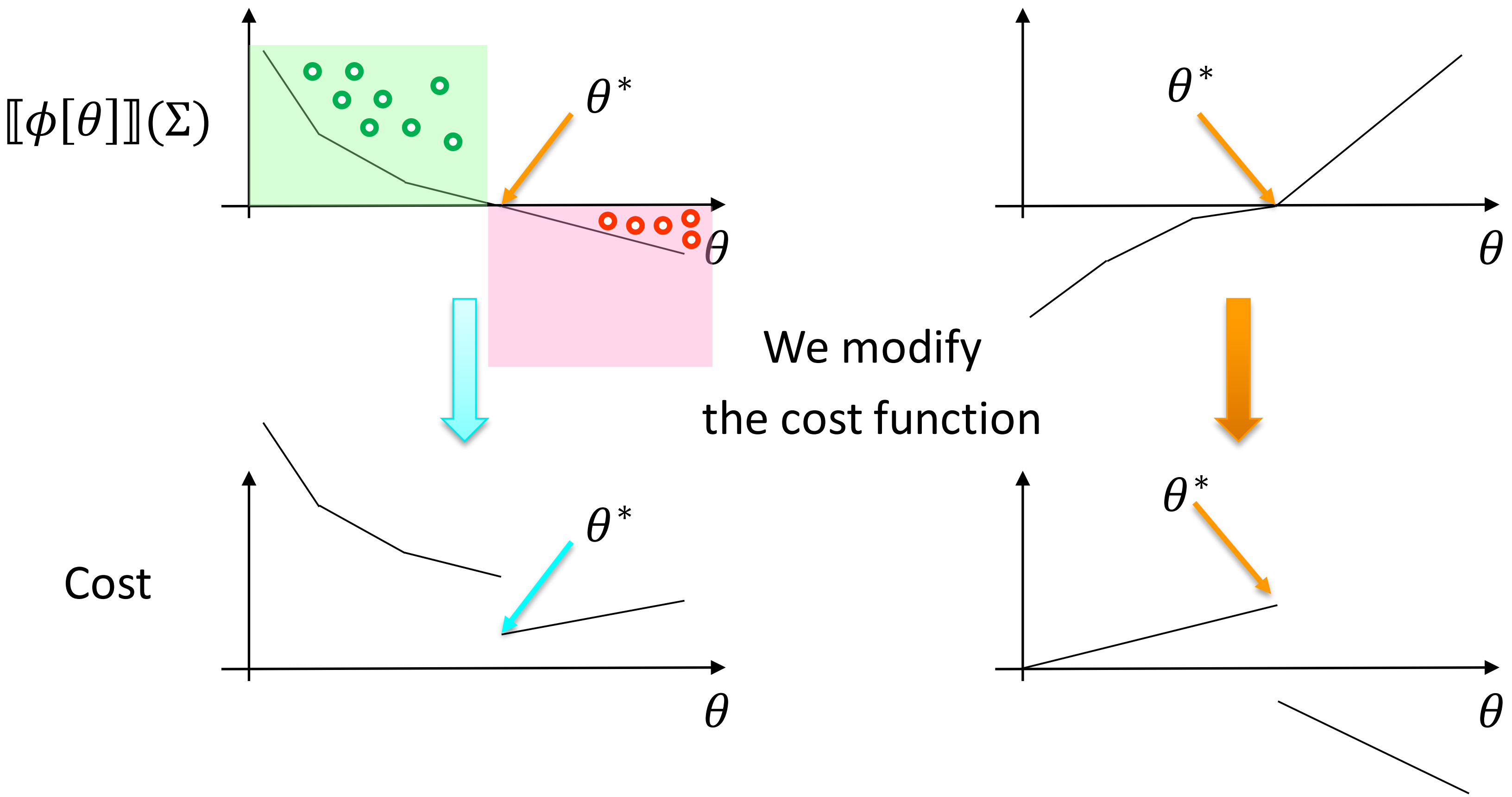
$$\min_{\theta\in\Theta}\ \min_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma+[\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

Non-Decreasing robustness with respect to θ

Maximize

$$\max_{\theta\in\Theta}\ \max_{\mu\in\mathcal{L}_\tau(\Sigma)}\left(f(\theta)+\begin{cases}\gamma-[\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0\\ 0 & \text{otherwise}\end{cases}\right)$$

Parameter Bound Computation



Non-Increasing robustness with respect to θ

Non-Decreasing robustness with respect to θ

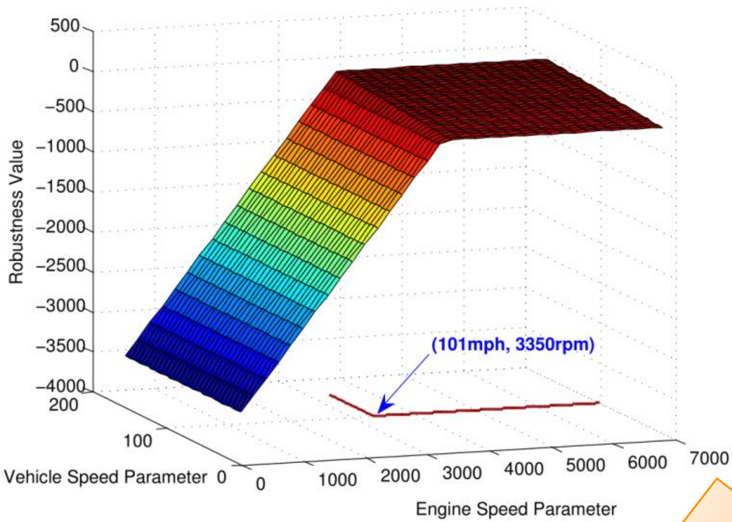
Minimize

$$\min_{\theta \in \Theta} \min_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma + [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

Maximize

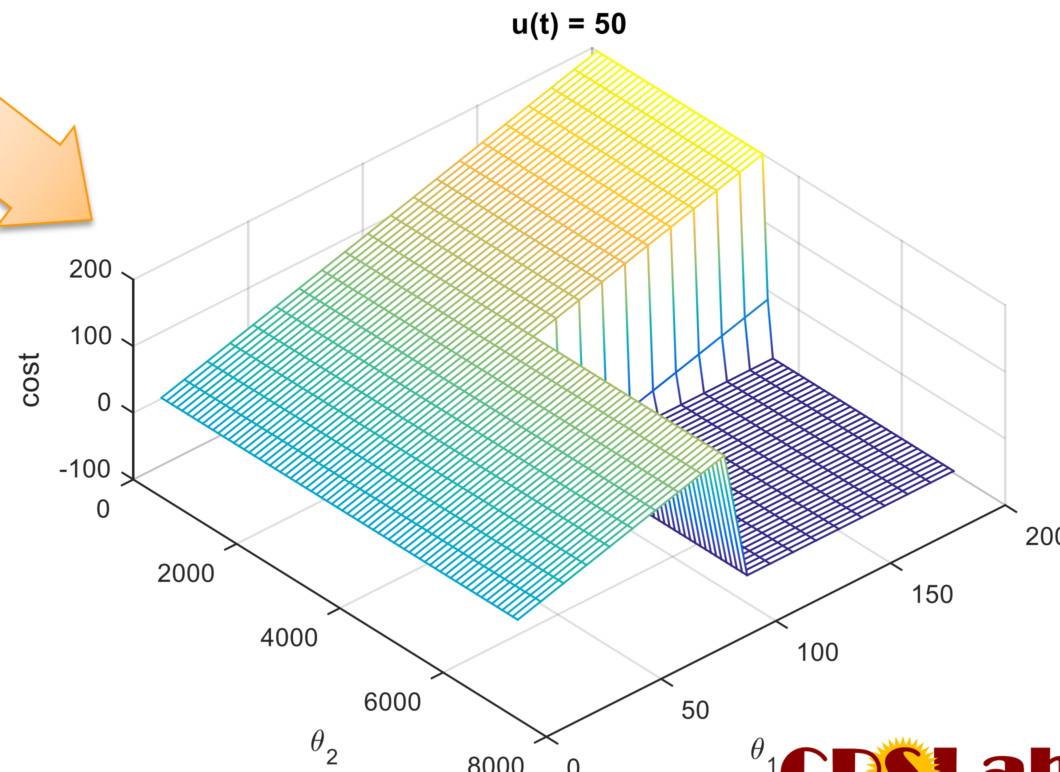
$$\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

Parameter Bound Computation



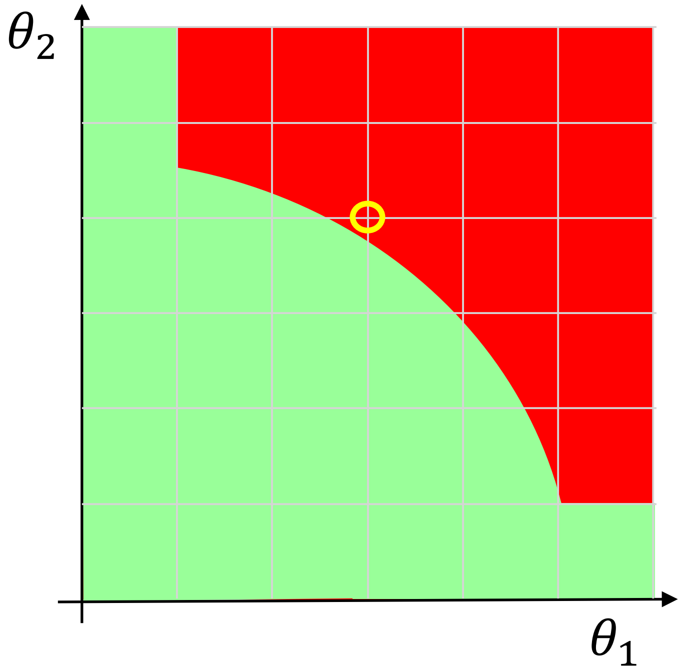
Non-Decreasing robustness with respect to $f(\vec{\theta})$

$$\max_{\theta\in\Theta}\ \max_{\mu\in\mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$



Parameter Falsification Domain

Non-Increasing robustness with respect to θ

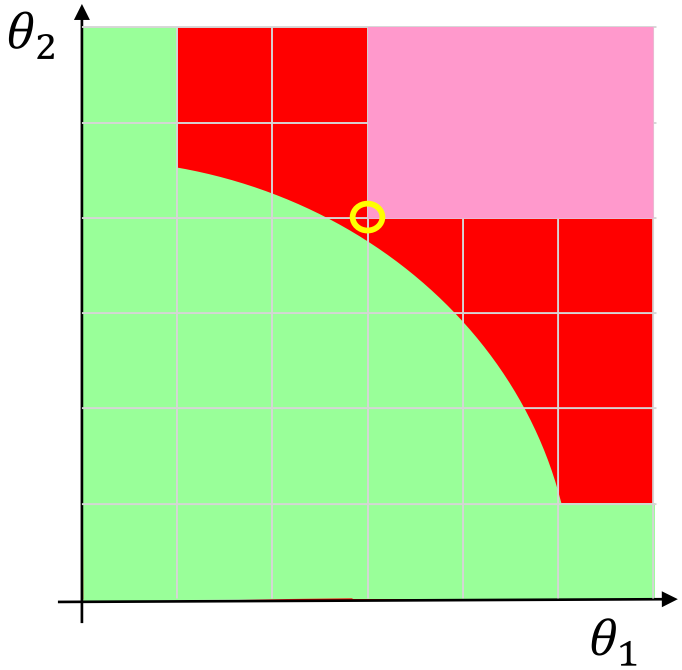


System fails the specification with θ_1 and θ_2

System satisfies the specification with θ_1 and θ_2

Parameter Falsification Domain

Non-Increasing robustness with respect to θ

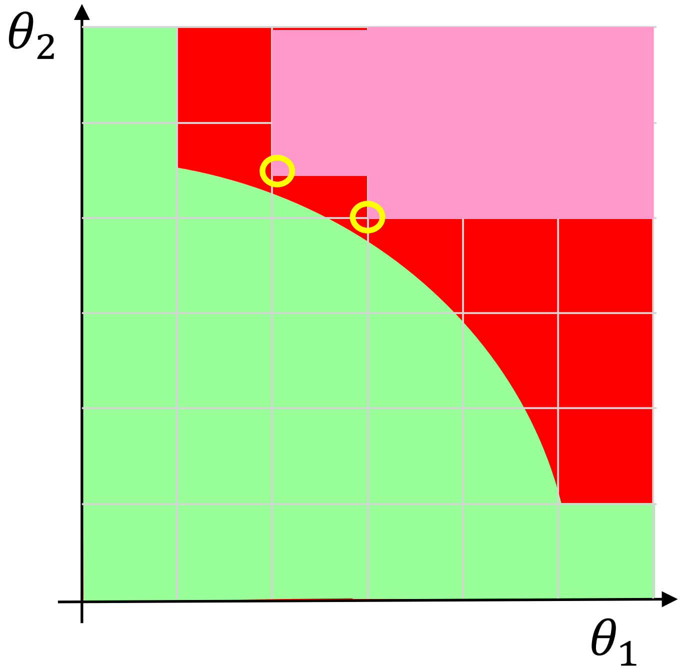


System fails the specification with θ_1 and θ_2

System satisfies the specification with θ_1 and θ_2

Parameter Falsification Domain

Non-Increasing robustness with respect to θ

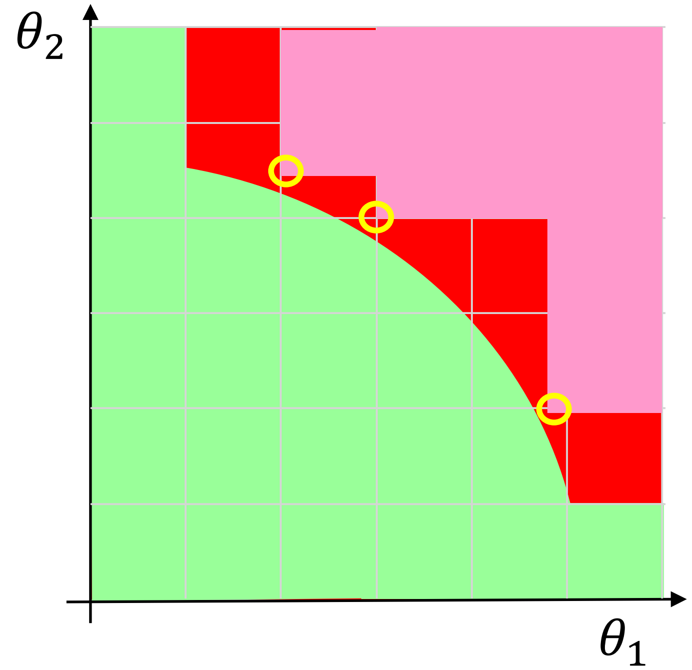


System fails the specification with θ_1 and θ_2

System satisfies the specification with θ_1 and θ_2

Parameter Falsification Domain

Non-Increasing robustness with respect to θ



System fails the specification with θ_1 and θ_2

System satisfies the specification with θ_1 and θ_2

Parameter Falsification Domain

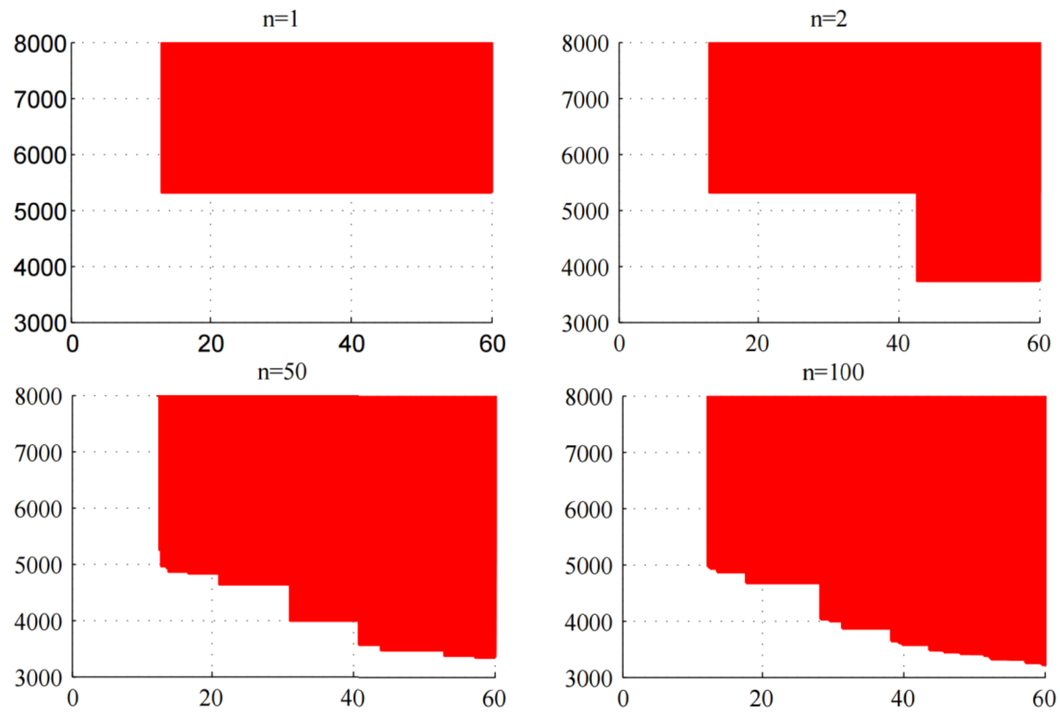
Alg 1: Robustness Guided Parameter Falsification Domain Algorithm

$$\phi[\theta] = \neg(\Diamond_{[0,\theta_1]}(v \geq 100) \wedge \Box(\omega \leq \theta_2))$$

Non-Increasing robustness with respect to f(θ)

In each iteration, shift weights of the priority function $f(\theta) = \sum w_i \theta_i$, which shifts the minimum of the cost function

$$\min_{\theta \in \Theta} \min_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma + [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$



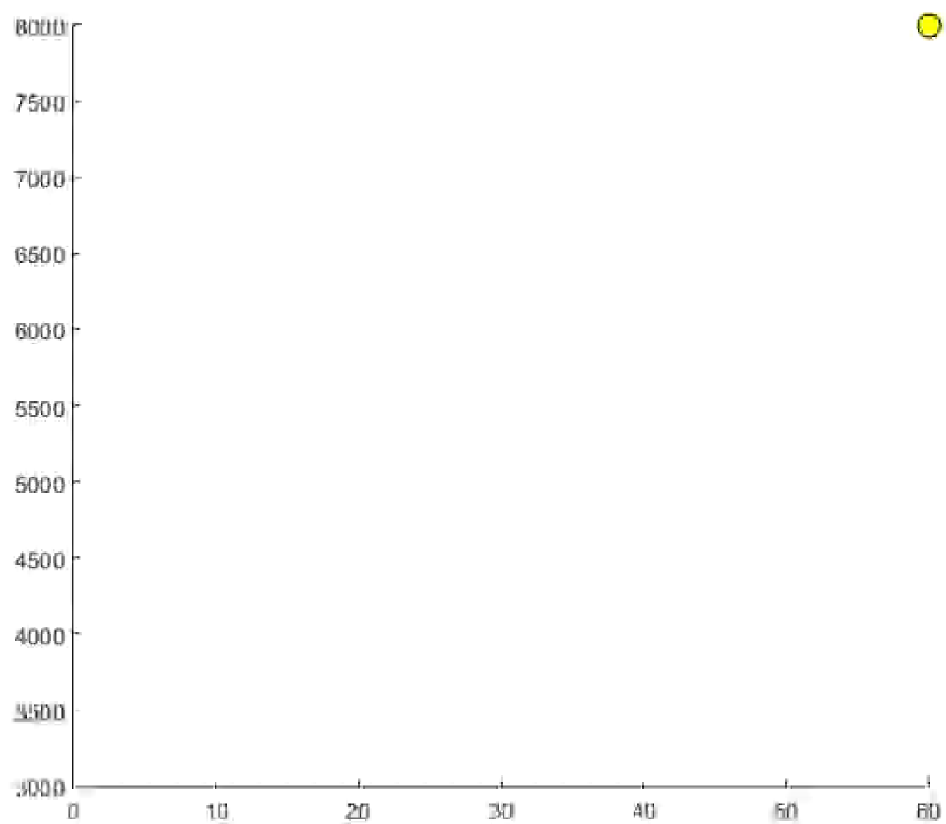
Red Colored Set represents the parameter falsification domain

Parameter Falsification Domain

Alg 1: Robustness Guided Parameter Falsification Domain Algorithm

$$\phi[\theta] = \neg(\Diamond_{[0,\theta_1]}(v \geq 100) \wedge \Box(\omega \leq \theta_2))$$

Non-Increasing robustness with respect to $f(\theta)$

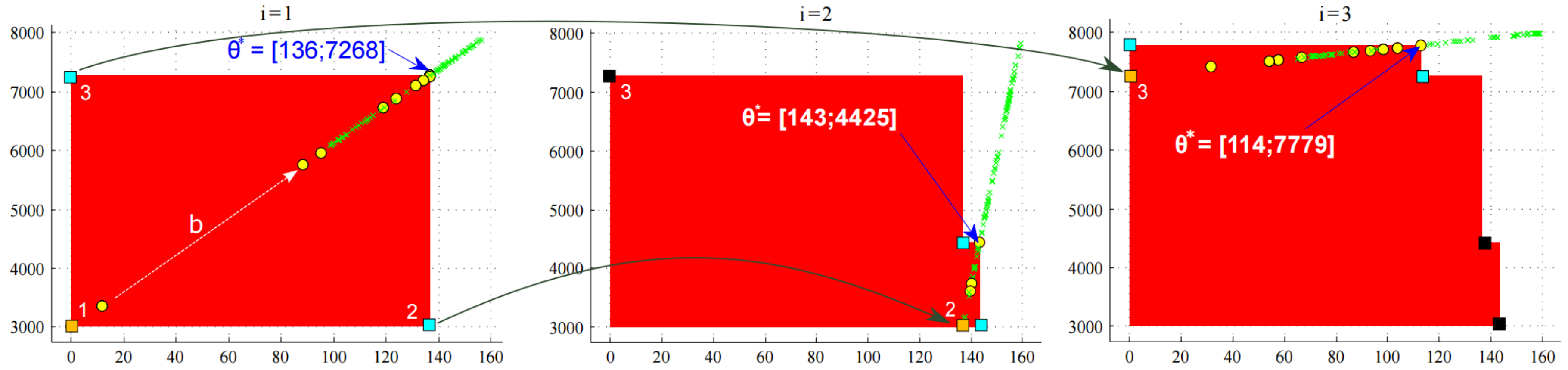


Parameter Falsification Domain

Alg 2: Structured Parameter Falsification Domain Algorithm

$$\phi[\theta] = \Box((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$

Non-Decreasing robustness with respect to $f(\vec{\theta})$



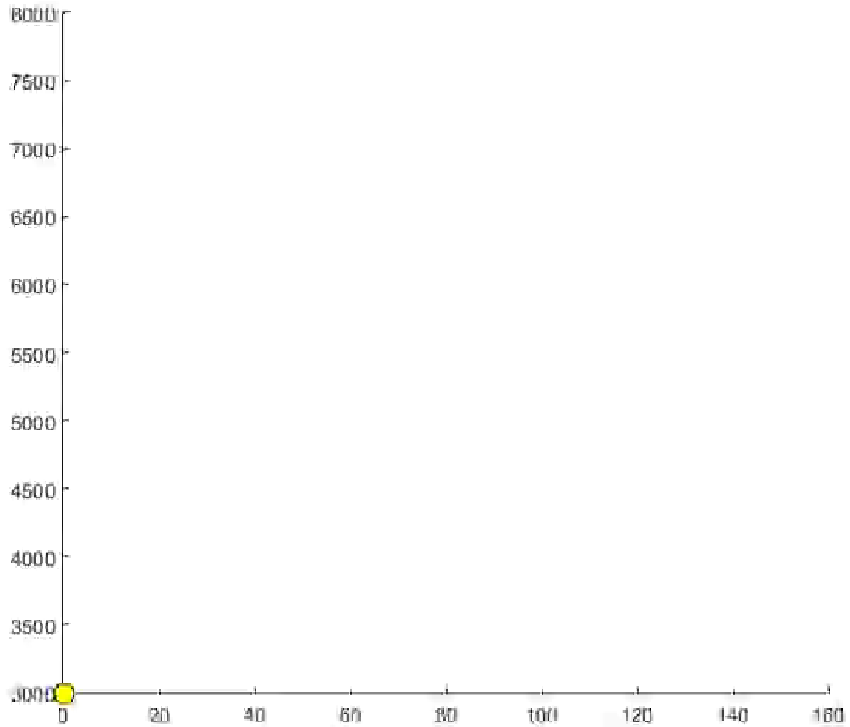
$$\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

Parameter Falsification Domain

Alg 2: Structured Parameter Falsification Domain Algorithm

$$\phi[\theta] = \Box((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$

Non-Decreasing robustness with respect to $f(\vec{\theta})$



Related Works

Parametric temporal logics over:

- Finite State Machines:
  - Alur et al. Parametric temporal logic for model measuring, 2001
- Timed Automata:
  - R. Alur et al. Parametric real-time reasoning, 1993
  - Bozzelli and La Torre. Decision problems for lower/upper bound parametric timed automata, 2009
- Hybrid Systems:
  - Asarin et al. Parametric identification of temporal properties, 2012.
  - Jin et al. Mining requirements from closed loop control models, 2013.

Conclusions

Conclusions

- We extend and generalize the parameter mining problem presented in [Yang, Hoxha and Fainekos, Querying Parametric Temporal Logic Properties on Embedded Systems, 2012].

Conclusions

- We extend and generalize the parameter mining problem presented in [Yang, Hoxha and Fainekos, Querying Parametric Temporal Logic Properties on Embedded Systems, 2012].
- We present two algorithms to explore the Pareto front of parametric MTL with multiple parameters.

Conclusions

- We extend and generalize the parameter mining problem presented in [Yang, Hoxha and Fainekos, Querying Parametric Temporal Logic Properties on Embedded Systems, 2012].
- We present two algorithms to explore the Pareto front of parametric MTL with multiple parameters.
- The algorithms presented in this work are publicly available through our toolbox S-TaLiRo.

Acknowledgements

Awards:
CNS-1319560, CNS-1116136,
IIP-1454143, IIP-1361926

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ARIZONA STATE UNIVERSITY

CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center

Thank you!

Questions?