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 $\Sigma$

$y = \Delta(x_0, u)$  \hspace{0.5cm} $x_0 \in X_0$

$u \in U$
What is the **shortest time** that the engine speed can exceed 3200RPM?

System $\Sigma$

$$y = \Delta(x_0, u) \quad x_0 \in X_0 \quad u \in U$$
What is the shortest time that the engine speed can exceed 3200RPM?

The vehicle speed is always less than parameter $\theta_1$ and the engine speed is always less than $\theta_2$. 

System $\Sigma$

$$y = \Delta(x_0,u) \quad x_0 \in X_0 \quad u \in U$$
What is the **shortest time** that the engine speed can exceed 3200RPM?

The vehicle speed is always less than parameter $\theta_1$ and the engine speed is always **less than $\theta_2$**.

If I **increase/decrease $\theta_1$** by a specific amount, how much do I have to **increase/decrease $\theta_2$** 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.

System $\Sigma$

$$y = \Delta(x_0, u) \quad x_0 \in X_0$$
$$u \in U$$
Preliminaries – Running Example

Automotive Transmission Simulink Demo
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 \( \omega \) is always below 4500RPM
Preliminaries - Metric Temporal Logic

Syntax: Boolean connectives with temporal operators

\[ \phi ::= T | \neg \phi | \phi_1 \vee \phi_2 | G \phi | F \phi | \phi_1 U_t \phi_2 \]

- \( G a \) - always a
  - a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a

- \( F a \) - eventually a
  - * \rightarrow * \rightarrow * \rightarrow a \rightarrow * \rightarrow *

- \( a U b \) - a until b
  - a \rightarrow a \rightarrow a \rightarrow b \rightarrow * \rightarrow *

- \( a U_{[1,1.5]} b \) - a until b
  - a \rightarrow a \rightarrow a \rightarrow b \rightarrow * \rightarrow *

Other notation: \( Ga \equiv \Box a \) and \( Fa \equiv \Diamond a \)
Parameter Mining

The vehicle speed is always less than parameter $\theta_1$ and the engine speed is always less than $\theta_2$.

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

PMTL formulas may contain state and/or timing parameters

Ex. $\phi_2[\vec{\theta}] = \neg(\diamond_{[0,\theta_1]}(v > 100) \land (\omega \leq \theta_2))$
Parameter Mining Problem:

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

Parameter Mining Problem:
Given a parametric MTL formula $\phi[\tilde{\theta}]$ with a vector of $m$ unknown parameters and a system $\Sigma$, 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 $\phi[\tilde{\theta}]$ with a vector of $m$ unknown parameters and a system $\Sigma$, find the set $\Psi = \{\theta^* \in \Theta | \Sigma \not\models \phi[\theta^*]\}$

Approximation possible 😊

Question:

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

Problem is undecidable [AL94] 😞.

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 $\theta^*$:

System $\Sigma$

$y = \Delta(x_0, u)$  \hspace{0.5cm} $x_0 \in X_0$  \hspace{0.5cm} $u \in U$

Output $y$

$\phi[\theta^*]$?

If yes, we have a counterexample proof that $\Sigma$ does not satisfy $\phi[\theta^*]$?

If no, generate another output
Output Trajectory Testing

For a specific parameter valuation $\theta^*$:

System $\Sigma$

\[
y = \Delta(x_0, u) \quad x_0 \in X_0 \quad u \in U
\]

If yes, we have a counterexample proof that $\Sigma$ does not satisfy $\phi[\theta^*]$?

If no, generate another output

Tester $y \not\equiv \phi[\theta^*]$?

By how much?

If yes, we have a counterexample proof that $\Sigma$ does not satisfy $\phi[\theta^*]$?
Robustness of Temporal Logics

Robustness Metric
\( \varepsilon \in \mathbb{R} \cup \{ \pm \infty \} \)

positive robustness \( \rightarrow \) signal satisfies the formula

negative robustness \( \rightarrow \) signal falsifies the formula

Falsification by optimization

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

\[ \phi[\theta^*] \]

\[ \text{System } \Sigma \]

Next
\[ x_0 \in X_0 \]
\[ u \in U \]

Stochastic Optimizer

Minimum Robustness
with corresponding input signal and initial conditions

Monotonicity of parametric MTL specifications

NL: Always, from 0 to \( \theta \), the engine speed is less than 3250

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

As we increase \( \theta \), we can only increase the opportunity to find falsifying system behavior.

Non-Increasing robustness with respect to \( \theta \)
Monotonicity of parametric MTL specifications

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

Monotonicity results formalized in
[Hoxha, Dokhanchi, and Fainekos, arXiv:1512.07956]
Monotonicity of parametric MTL specifications

NL: Always, vehicle speed is less than $\theta_1$ and engine speed is less than $\theta_2$

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

As we increase $\theta_1$ and $\theta_2$, we can only decrease the opportunity to find falsifying system behavior

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

Monotonicity results formalized in

[Hoxha, Dokhanchi, and Fainekos, arXiv:1512.07956]
Monotonicity of parametric MTL specifications

\[ \phi_1[\theta] = \square((v \leq \theta_1) \land (\omega \leq \theta_2)) \]
Monotonicity of parametric MTL specifications

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

Solution to the Parameter Mining Problem.

Namely, set \( \Psi = \{\theta^* \in \Theta \mid \Sigma \not\models \phi[\theta^*]\} \)
Parameter Bound Computation

\[ [\phi[\theta]](\Sigma) \]

\[ \theta^* \]

Cost

\[ \theta^* \]

We modify the cost function

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

Minimize

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

Non-Decreasing robustness with respect to \( \theta \)

Maximize

\[
\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_r(\Sigma)} \left( f(\theta) + \begin{cases} \gamma - [\phi[\theta]](\mu) & \text{if } [\phi[\theta]](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)
\]
Parameter Bound Computation

\[ \left[ \phi[\theta] \right](\Sigma) \]

\[ \theta^* \]

Cost

We modify the cost function

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

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

Non-Decreasing robustness with respect to \( \theta \)

\[ \max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}^r(\Sigma)} \left( f(\theta) + \begin{cases} \gamma - \left[ \phi[\theta] \right](\mu) & \text{if } \left[ \phi[\theta] \right](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right) \]
Parameter Bound Computation

\[ \phi[\theta](\Sigma) \]

\[ \theta^* \]

We modify the cost function

\[ \text{Cost} \]

\[ \theta^* \]

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

Minimize

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

Non-Decreasing robustness with respect to \( \theta \)

Maximize

\[ \max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_r(\Sigma)} \left( f(\theta) + \begin{cases} \gamma - \phi[\theta](\mu) & \text{if } \phi[\theta](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right) \]
Parameter Bound Computation

\[ [\phi[\theta]](\Sigma) \]

\[ \theta^* \]

We modify the cost function

Cost

\[ \theta \]

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

Minimize

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

Non-Decreasing robustness with respect to \( \theta \)

Maximize

\[
\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_r(\Sigma)} \left( f(\theta) + \begin{cases} 
\gamma - [\phi[\theta]](\mu) & \text{if } [\phi[\theta]](\mu) \geq 0 \\
0 & \text{otherwise}
\end{cases} \right)
\]
Parameter Bound Computation

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

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

$u(t) = 50$
Parameter Falsification Domain

Non-Increasing robustness with respect to $\theta$

- System fails the specification with $\theta_1$ and $\theta_2$
- System satisfies the specification with $\theta_1$ and $\theta_2$
Parameter Falsification Domain

Non-Increasing robustness with respect to $\theta$

The diagram shows the parameter falsification domain in the $\theta_1$-$\theta_2$ plane.

- **Red Area**: System fails the specification with $\theta_1$ and $\theta_2$
- **Green Area**: System satisfies the specification with $\theta_1$ and $\theta_2$
Parameter Falsification Domain

Non-Increasing robustness with respect to $\theta$

- Red: System fails the specification with $\theta_1$ and $\theta_2$
- Green: System satisfies the specification with $\theta_1$ and $\theta_2$
Parameter Falsification Domain

Non-Increasing robustness with respect to $\theta$

- System fails the specification with $\theta_1$ and $\theta_2$
- System satisfies the specification with $\theta_1$ and $\theta_2$
Parameter Falsification Domain

Alg 1: Robustness Guided Parameter Falsification Domain Algorithm

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

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

In each iteration, shift weights of the priority function

\[ f(\theta) = \sum w_i \theta_i, \]

which shifts the minimum of the cost function

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

Red Colored Set represents the parameter falsification domain
Parameter Falsification Domain

Alg 1: Robustness Guided Parameter Falsification Domain Algorithm

\[
\phi[\theta] = \neg(\lozenge_{[0,\theta_1]}(v \geq 100) \land \square(\omega \leq \theta_2))
\]

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

Alg 2: Structured Parameter Falsification Domain Algorithm

\[ \phi[\theta] = \square((v \leq \theta_1) \land (\omega \leq \theta_2)) \]

Non-Decreasing robustness with respect to \( f(\tilde{\theta}) \):

\[
\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_r(\Sigma)} \left( f(\theta) + \begin{cases} 
\gamma - \|\phi[\theta]\|(\mu) & \text{if } \|\phi[\theta]\|(\mu) \geq 0 \\
0 & \text{otherwise}
\end{cases} \right)
\]
Parameter Falsification Domain

Alg 2: Structured Parameter Falsification Domain Algorithm

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

Non-Decreasing robustness with respect to \(f(\hat{\theta})\)
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:
  - Jin et al. Mining requirements from closed loop control models, 2013.
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:
NSF 1116136 and NSF 1350420

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.

Thank you!
Questions?