Parametric Identification of Temporal Properties

2nd International Conference on
Runtime Verification,
September 28th, 2011

Eugène Asarin [1], **Alexandre Donzé**[2], Oded Maler[2] and Dejan Nickovic [3]

[1]LIAFA, Université Paris Diderot / CNRS, Paris, France

[2]Verimag, Université Joseph Fourier /CNRS, Gières, France

[3]IST Austria, Klosterneuburg, Austria

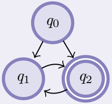
Context

- Properties monitoring of hybrid systems (embedded systems, mixed-signal circuits, etc)
- Behaviours have dense-time and real-valued components
- We specify and monitor properties using Signal Temporal Logic

In this work, we consider *Parametric* STL formulas, i.e., formulas with parameters

Hybrid System

$$\dot{x} = f_q(x, t) \;||$$



Goal: solving an inverse problem

For which values of parameters (e.g., p, s_1, s_2) does a given signal x satisfies φ ?

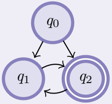
Context

- Properties monitoring of hybrid systems (embedded systems, mixed-signal circuits, etc)
- Behaviours have dense-time and real-valued components
- We specify and monitor properties using Signal Temporal Logic

In this work, we consider *Parametric* STL formulas, i.e., formulas with parameters

Hybrid System

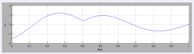
$\dot{x} = f_q(x,t) \; ||$



$\longrightarrow$

Simulation Trace

$q_0 \rightarrow q_1 \rightarrow \cdots$



$x(t)$

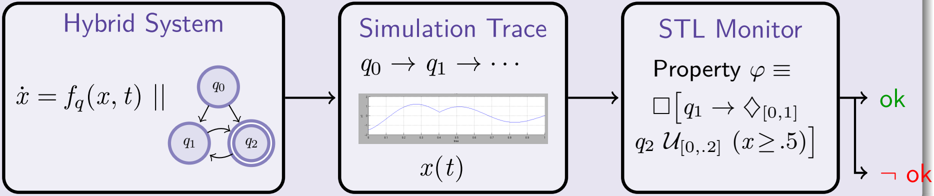
Goal: solving an inverse problem

For which values of parameters (e.g., p, s_1, s_2) does a given signal x satisfies φ ?

Context

- Properties monitoring of hybrid systems (embedded systems, mixed-signal circuits, etc)
- Behaviours have dense-time and real-valued components
- We specify and monitor properties using Signal Temporal Logic

In this work, we consider *Parametric* STL formulas, i.e., formulas with parameters



Goal: solving an inverse problem

For which values of parameters (e.g.,p, s_1, s_2) does a given signal x satisfies φ ?

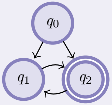
Context

- Properties monitoring of hybrid systems (embedded systems, mixed-signal circuits, etc)
- Behaviours have dense-time and real-valued components
- We specify and monitor properties using Signal Temporal Logic

In this work, we consider *Parametric* STL formulas, i.e., formulas with parameters

Hybrid System

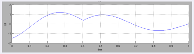
$\dot{x} = f_q(x, t) \; ||$



→

Simulation Trace

$q_0 \rightarrow q_1 \rightarrow \cdots$



$x(t)$

→

STL Monitor

Property $\varphi \equiv$

$\Box\big[q_1 \rightarrow \Diamond_{[0,s_1]}$
$q_2 \; \mathcal{U}_{[0,s_2]} \; (x \geq p)\big]$

→ ok

→ $\neg$ ok

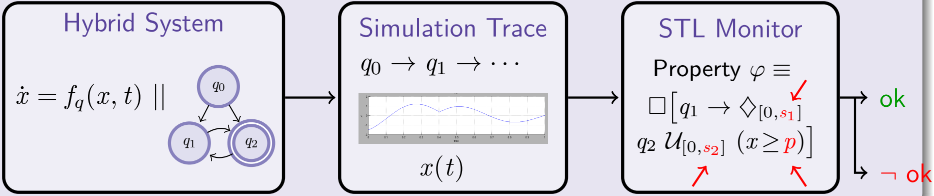
Goal: solving an inverse problem

For which values of parameters (e.g., p, s_1, s_2) does a given signal x satisfies φ ?

Context

- Properties monitoring of hybrid systems (embedded systems, mixed-signal circuits, etc)
- Behaviours have dense-time and real-valued components
- We specify and monitor properties using Signal Temporal Logic

In this work, we consider *Parametric* STL formulas, i.e., formulas with parameters



Goal: solving an inverse problem

For which values of parameters (e.g.,p, s_1, s_2) does a given signal x satisfies φ ?

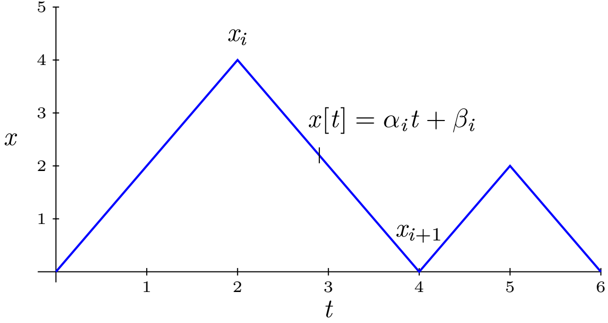
Outline

1. Parametric Signal Temporal Logic
2. Exact Computation of Validity Domains
3. Numerical Approximation of Validity Domains

Outline

Signals

We consider piecewise affine signals given by sets of points ($x_i = x[t_i]$)



In this talk, we consider 1-dimensional signals for simplicity.

Signal Temporal Logic

Goal specifying temporal properties of signals

Definition (STL Syntax)

An STL formula is given by:

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2$$

where μ is a predicate of the form $f(x) > p$.

Definition (STL Semantics)

The validity of a formula φ with respect to a signal x at time t is

$$\begin{array}{lcl}
(x,t) \models f(x) > p & \Leftrightarrow & f(x[t]) > p \\
(x,t) \models \varphi \wedge \psi & \Leftrightarrow & (x,t) \models \varphi \wedge (x,t) \models \psi \\
(x,t) \models \neg\varphi & \Leftrightarrow & \neg((x,t) \models \varphi) \\
(x,t) \models \varphi \, \mathcal{U}_{[a,b]} \psi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (x,t') \models \psi \wedge \\
& & \qquad \forall t'' \in [t,t'],\ (x,t'') \models \varphi \}
\end{array}$$

Signal Temporal Logic

Goal specifying temporal properties of signals

Definition (STL Syntax)

An STL formula is given by:

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2$$

where μ is a predicate of the form $f(x) > p$.

Definition (STL Semantics)

The validity of a formula φ with respect to a signal x at time t is

$$\begin{array}{lcl}
(x,t) \models f(x) > p & \Leftrightarrow & f(x[t]) > p \\
(x,t) \models \varphi \wedge \psi & \Leftrightarrow & (x,t) \models \varphi \wedge (x,t) \models \psi \\
(x,t) \models \neg\varphi & \Leftrightarrow & \neg((x,t) \models \varphi) \\
(x,t) \models \varphi \, \mathcal{U}_{[a,b]} \psi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (x,t') \models \psi \wedge \\
 & & \quad \forall t'' \in [t,t'],\ (x,t'') \models \varphi \}
\end{array}$$

Signal Temporal Logic

Goal specifying temporal properties of signals

Definition (STL Syntax)

An STL formula is given by:

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2$$

where μ is a predicate of the form $f(x) > p$.

Definition (STL Semantics)

The validity of a formula φ with respect to a signal x at time t is

$$\begin{array}{lcl}
(x,t) \models f(x) > p & \Leftrightarrow & f(x[t]) > p \\
(x,t) \models \varphi \wedge \psi & \Leftrightarrow & (x,t) \models \varphi \wedge (x,t) \models \psi \\
(x,t) \models \neg\varphi & \Leftrightarrow & \neg((x,t) \models \varphi) \\
(x,t) \models \varphi \, \mathcal{U}_{[a,b]} \psi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (x,t') \models \psi \, \wedge \\
 & & \qquad \forall t'' \in [t,t'],\ (x,t'') \models \varphi \}
\end{array}$$

Signal Temporal Logic

Goal specifying temporal properties of signals

Definition (STL Syntax)

An STL formula is given by:

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \,\mathcal{U}_{[a,b]}\, \varphi_2$$

where μ is a predicate of the form $f(x) > p$.

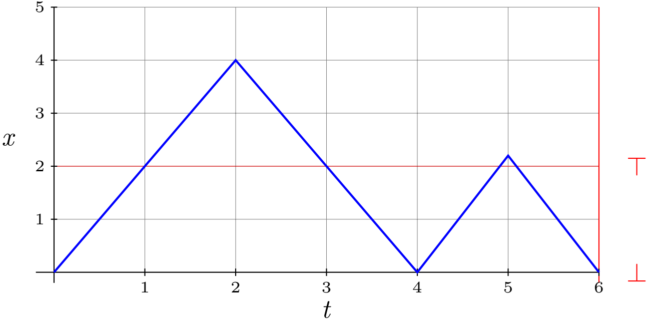
Definition (STL Semantics)

The validity of a formula φ with respect to a signal x at time t is

$$\begin{array}{lcl}
(x,t) \models f(x) > p & \Leftrightarrow & f(x[t]) > p \\
(x,t) \models \varphi \wedge \psi & \Leftrightarrow & (x,t) \models \varphi \wedge (x,t) \models \psi \\
(x,t) \models \neg\varphi & \Leftrightarrow & \neg((x,t) \models \varphi) \\
(x,t) \models \varphi\, \mathcal{U}_{[a,b]} \psi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (x,t') \models \psi \wedge \\
 & & \qquad\qquad \forall t'' \in [t,t'],\ (x,t'') \models \varphi\}
\end{array}$$

Additionally: $\Diamond_{[a,b]}\varphi = \top\, \mathcal{U}_{[a,b]}\varphi$ and $\Box_{[a,b]}\varphi = \varphi\, \mathcal{U}_{[a,b]}\bot$.

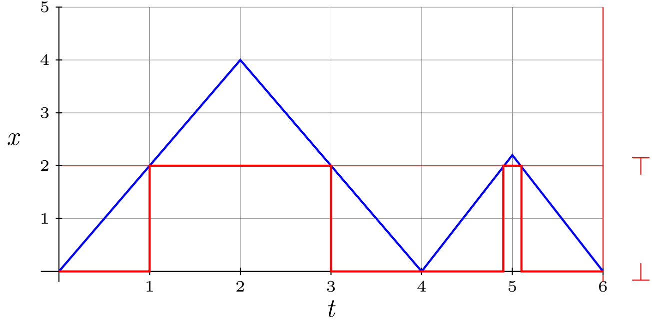
Examples



Truth value of :

- $\varphi = x > 2$

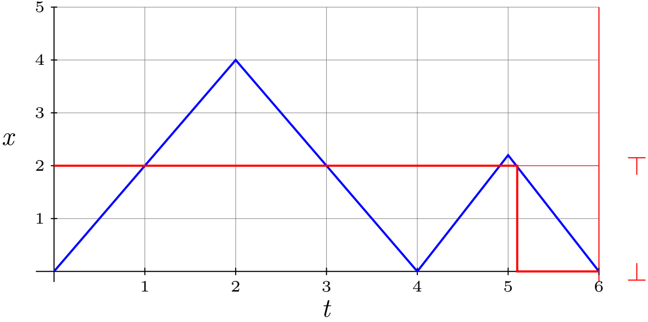
Examples



Truth value of :

- $\varphi = x > 2$

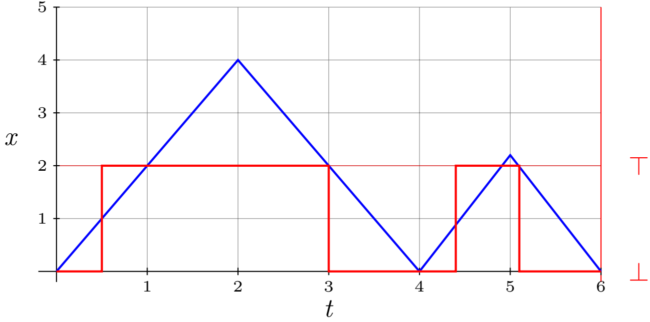
Examples



Truth value of :

- $\varphi = x > 2$
- $\varphi = \Diamond_{[0,\infty]}(x > 2)$

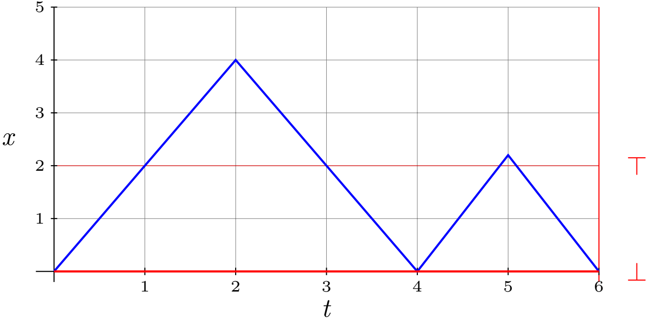
Examples



Truth value of :

- $\varphi = x > 2$
- $\varphi = \Diamond_{[0,.5]}(x > 2)$

Examples

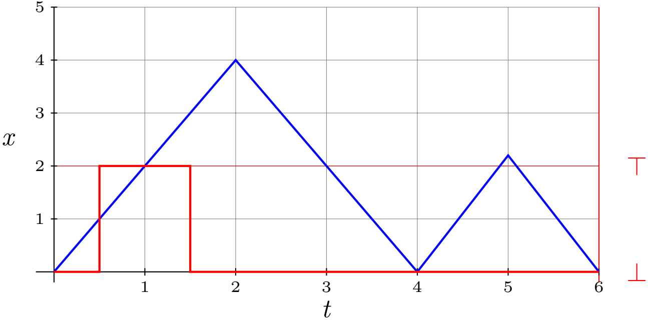


Truth value of :

- $\varphi = x > 2$

- $\varphi = \Box_{[0,\infty]}(x > 2)$

Examples



Truth value of :

- $\varphi = x > 2$

- $\varphi = \Box_{[0.5,1.5]}(x > 2)$

Outline

Validity Domains

Definition

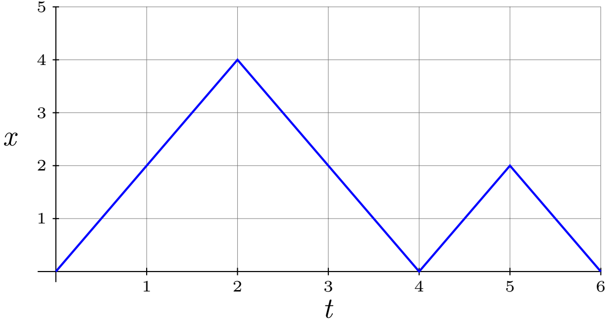
Let φ be a PSTL formula with amplitude parameters $p \in \mathcal{P}$ and timing parameters $s \in \mathcal{S}$.

The validity domain of a signal x for φ is the set of times and parameters values for which φ is satisfied by x

$$d(x, \varphi) = \{(p, s, t) \in \mathcal{P} \times \mathcal{S} \times \mathbb{R}^+ \text{ s.t. } (x, t) \models \varphi_{p,s}\}$$

Validity Domains of Predicates

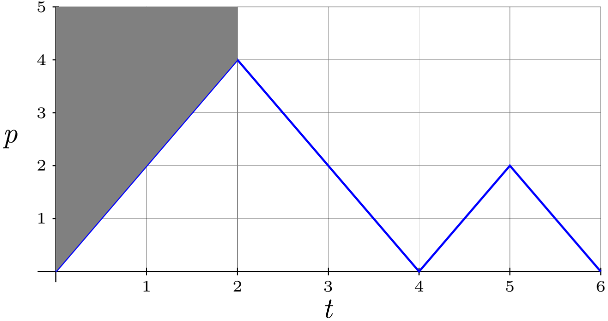
Consider again signal x and the predicate $x > p$.



$$
\begin{aligned}
d(x, x > p) \quad = \quad & (t \geq 0 \wedge t < 2 \wedge 2p > 4t) && \vee \\
& (t \geq 2 \wedge t < 4 \wedge 2p + 4t > 16) && \vee \\
& (t \geq 4 \wedge t < 5 \wedge p > 2t - 8) && \vee \\
& (t \geq 5 \wedge t < 6 \wedge p + 2t > 12)
\end{aligned}
$$

Validity Domains of Predicates

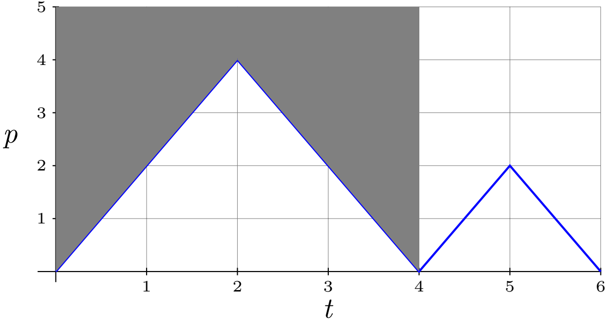
Consider again signal x and the predicate $x > p$.



$$
\begin{aligned}
d(x, x > p) \quad = \quad & (t \geq 0 \wedge t < 2 \wedge 2p > 4t) & \vee \\
& (t \geq 2 \wedge t < 4 \wedge 2p + 4t > 16) & \vee \\
& (t \geq 4 \wedge t < 5 \wedge p > 2t - 8) & \vee \\
& (t \geq 5 \wedge t < 6 \wedge p + 2t > 12) &
\end{aligned}
$$

Validity Domains of Predicates

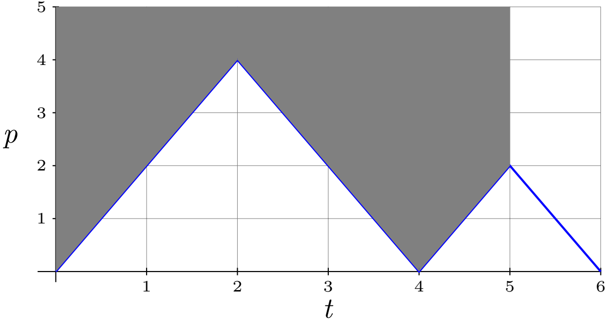
Consider again signal x and the predicate $x > p$.



$$
\begin{aligned}
d(x, x > p) \quad = \quad & (t \geq 0 \wedge t < 2 \wedge 2p > 4t) && \vee \\
& (t \geq 2 \wedge t < 4 \wedge 2p + 4t > 16) && \vee \\
& (t \geq 4 \wedge t < 5 \wedge p > 2t - 8) && \vee \\
& (t \geq 5 \wedge t < 6 \wedge p + 2t > 12) &&
\end{aligned}
$$

Validity Domains of Predicates

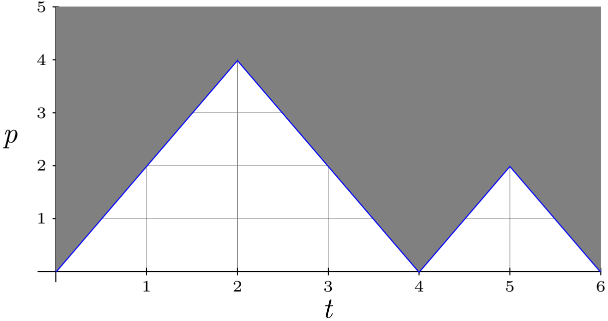
Consider again signal x and the predicate $x > p$.



$$
\begin{aligned}
d(x, x > p) \quad = \quad & (t \geq 0 \wedge t < 2 \wedge 2p > 4t) && \vee \\
& (t \geq 2 \wedge t < 4 \wedge 2p + 4t > 16) && \vee \\
& (t \geq 4 \wedge t < 5 \wedge p > 2t - 8) && \vee \\
& (t \geq 5 \wedge t < 6 \wedge p + 2t > 12) &&
\end{aligned}
$$

Validity Domains of Predicates

Consider again signal x and the predicate $x > p$.



$$
\begin{array}{rcll}
d(x, x > p) & = & (t \geq 0 \wedge t < 2 \wedge 2p > 4t) & \vee \\
 & & (t \geq 2 \wedge t < 4 \wedge 2p + 4t > 16) & \vee \\
 & & (t \geq 4 \wedge t < 5 \wedge p > 2t - 8) & \vee \\
 & & (t \geq 5 \wedge t < 6 \wedge p + 2t > 12) &
\end{array}
$$

Inductive Computation of Validity Domains

Validity domains can be computed inductively as follows:

$$
\begin{array}{lcl}
d(x, f(x) > p) & = & \{(t,p) : f(x[t]) > p\} \\
d(x, \varphi \wedge \psi) & = & d(x,\varphi) \cap d(x,\psi) \\
d(x, \neg\varphi) & = & \overline{d(x,\varphi)} \\
d(x, \varphi\, \mathcal{U}_{[a,b]} \psi) & = & \{(t,p,s) : \exists t' \in [t+a, t+b] \text{ s.t. } (t',p,s) \in d(x,\psi) \wedge \\
 & & \qquad \forall t'' \in [t,t'] (t'',p,s) \in d(x,\varphi)\}
\end{array}
$$

We use quantifier elimination (QE) to get inductively quantifier free (QF) formulas describing $d(x,\varphi)$.

Inductive Computation of Validity Domains

Validity domains can be computed inductively as follows:

$$
\begin{array}{lcl}
d(x, f(x) > p) & = & \{(t,p) : f(x[t]) > p\} \\
d(x, \varphi \wedge \psi) & = & d(x,\varphi) \cap d(x,\psi) \\
d(x, \neg\varphi) & = & \overline{d(x,\varphi)} \\
d(x, \varphi\, \mathcal{U}_{[a,b]}\psi) & = & \{(t,p,s) : \exists t' \in [t+a, t+b] \text{ s.t. } (t',p,s) \in d(x,\psi) \wedge \\
 & & \qquad \forall t'' \in [t,t'] (t'',p,s) \in d(x,\varphi)\}
\end{array}
$$

We use quantifier elimination (QE) to get inductively quantifier free (QF) formulas describing $d(x,\varphi)$.

Example

Validity domains of $\varphi = \Box_{[0,s_1]}(x < p)$?

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
&\quad = \{(t, s_1, p) \text{ s.t. } (x, t) \models \Box_{[0,s_1]}(x < p)\} \\
&\quad = \{(t, s_1, p) \text{ s.t. } \forall \tau,\ t \le \tau \le t + s_1 \implies (\tau, p) \in d(x, x < p)\}
\end{aligned}
$$

Quantifier-free (QF) formula, knowing a QF formula for $d(x, x < p)$:

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
&\quad = QE\Big(\forall \tau, t \le \tau \le t + s_1 \implies (\tau \ge 0 \wedge \tau < 2 \wedge 2p > 4\tau) \vee \ldots\Big) \\
&\quad = p + 2s_1 + 2t < 12 \vee p + 2t > 12 \vee p > 0 \vee p \le 0) \wedge \\
&\qquad (p + 2s_1 + 2t < 8 \vee p + 2t > 8 \vee p + 4 \le 0 \vee p > 4) \wedge \\
&\qquad (s_1 + t \ge 6 \vee (p - 2s_1 - 2t > 0 \wedge s_1 + t < 2) \vee \\
&\qquad \ldots
\end{aligned}
$$

Example

Validity domains of $\varphi = \Box_{[0,s_1]}(x < p)$?

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
& \quad = \ \{(t, s_1, p) \text{ s.t. } (x,t) \models \Box_{[0,s_1]}(x < p)\} \\
& \quad = \ \{(t, s_1, p) \text{ s.t. } \forall \tau,\ t \le \tau \le t + s_1 \implies (\tau, p) \in d(x, x < p)\}
\end{aligned}
$$

Quantifier-free (QF) formula, knowing a QF formula for $d(x, x < p)$:

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
& \quad = \ QE\Big(\forall \tau, t \le \tau \le t + s_1 \implies (\tau \ge 0 \wedge \tau < 2 \wedge 2p > 4\tau) \vee \ldots\Big) \\
& \quad = \ p + 2s_1 + 2t < 12 \vee p + 2t > 12 \vee p > 0 \vee p \le 0) \wedge \\
& \qquad (p + 2s_1 + 2t < 8 \vee p + 2t > 8 \vee p + 4 \le 0 \vee p > 4) \wedge \\
& \qquad (s_1 + t \ge 6 \vee (p - 2s_1 - 2t > 0 \wedge s_1 + t < 2) \vee \\
& \qquad \ldots
\end{aligned}
$$

Example

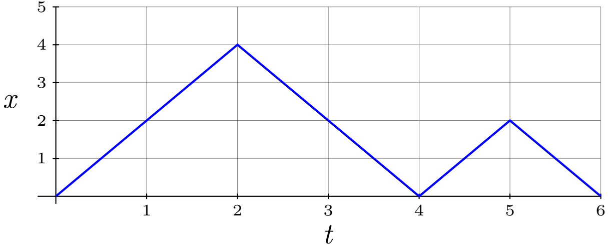
Validity domains of $\varphi = \Box_{[0,s_1]}(x < p)$?

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
& \quad = \{(t, s_1, p) \text{ s.t. } (x, t) \models \Box_{[0,s_1]}(x < p)\} \\
& \quad = \{(t, s_1, p) \text{ s.t. } \forall \tau,\ t \le \tau \le t + s_1 \implies (\tau, p) \in d(x, x < p)\}
\end{aligned}
$$

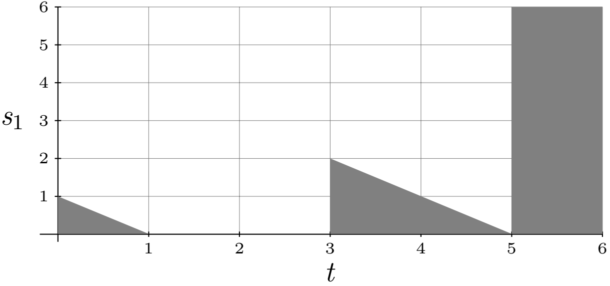
Quantifier-free (QF) formula, knowing a QF formula for $d(x, x < p)$:

$$
\begin{aligned}
& d(x, \Box_{[0,s_1]}(x < p)) \\
& \quad = QE\Big(\forall \tau, t \le \tau \le t + s_1 \implies (\tau \ge 0 \land \tau < 2 \land 2p > 4\tau) \lor \dots\Big) \\
& \quad = p + 2s_1 + 2t < 12 \lor p + 2t > 12 \lor p > 0 \lor p \le 0) \land \\
& \qquad (p + 2s_1 + 2t < 8 \lor p + 2t > 8 \lor p + 4 \le 0 \lor p > 4) \land \\
& \qquad (s_1 + t \ge 6 \lor (p - 2s_1 - 2t > 0 \land s_1 + t < 2) \lor \\
& \qquad \dots
\end{aligned}
$$

Example



Validity domain $d(x, \Box_{[0,s_1]}(x < 2))$



Experimental Results

Typical stabilization property

$$\varphi_{st} : \Box((x > p) \to \Diamond_{[0,s_2]}\Box_{[0,s_1]}(x < p)).$$

which we decompose into subformulas

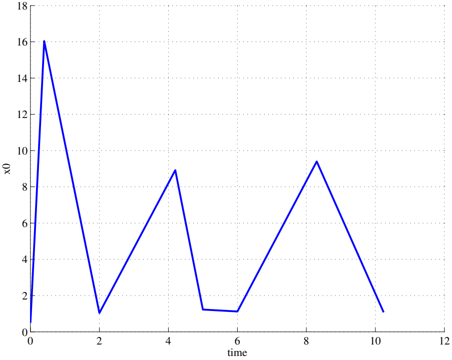
$$\begin{array}{ll} \varphi_1 : & \Box_{[0,s_1]}(x < p) \\ \varphi_2 : & \Diamond_{[0,s_2]}\Box_{[0,s_1]}(x < p) \\ \varphi_3 : & (x \geq p) \to \Diamond_{[0,s_2]}\Box_{[0,s_1]}(x < p) \end{array}$$

We tested the computation of validity domains using QE for a signal with different number of samples.

Experimental Results

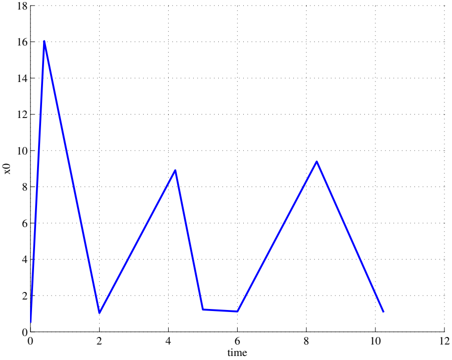
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



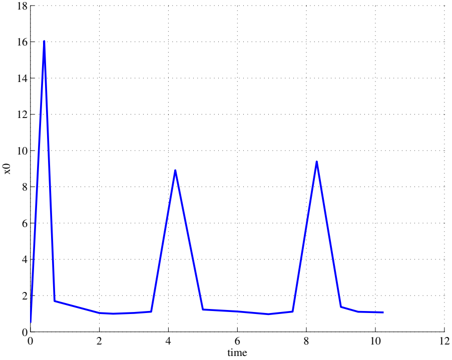
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



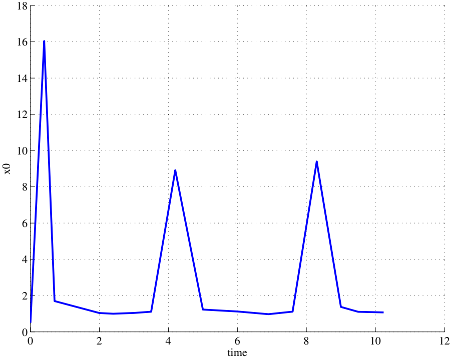
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



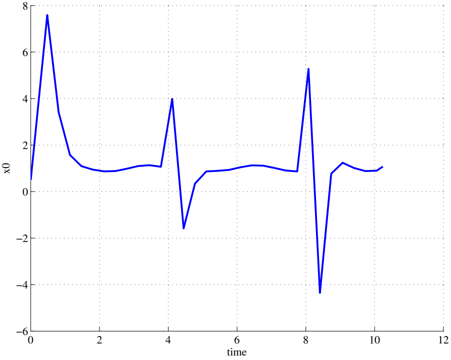
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



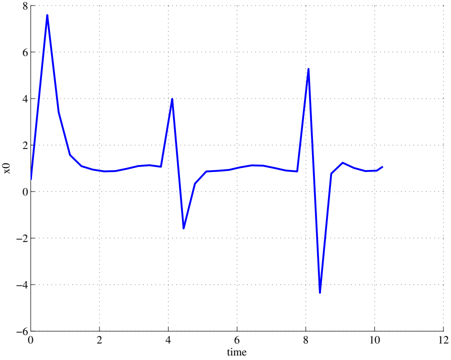
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



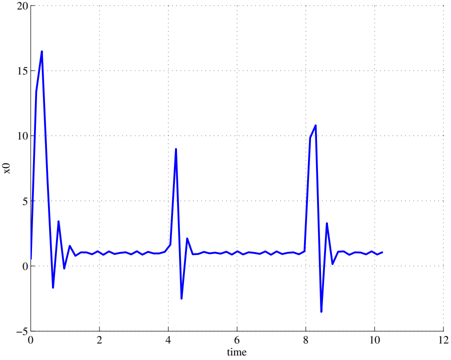
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



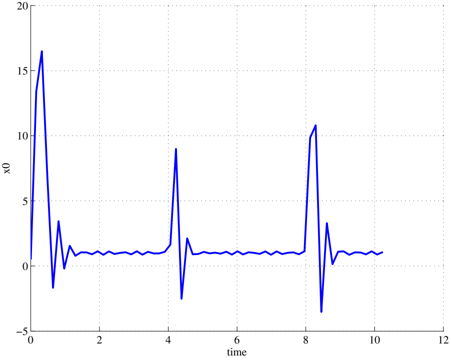
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



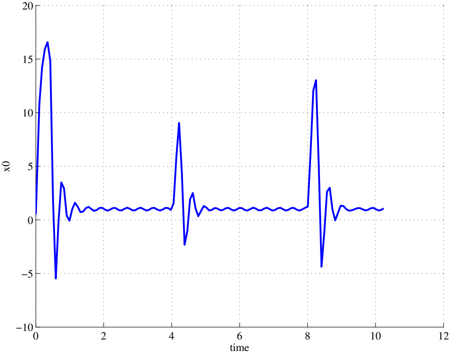
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	$*$	$*$
64	4.16	144	341.95	23103	308.93	10258	$*$	$*$
128	68.29	895	$*$	$*$	$*$	$*$	$*$	$*$
256	386.72	3098	$*$	$*$	$*$	$*$	$*$	$*$

Experimental Results



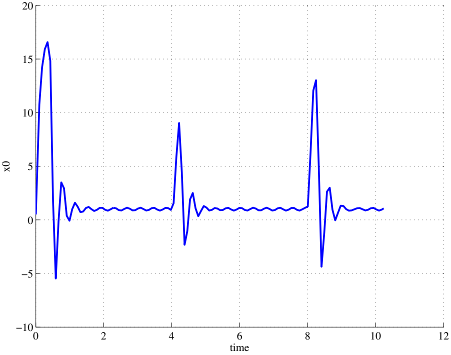
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	$*$	$*$
64	4.16	144	341.95	23103	308.93	10258	$*$	$*$
128	68.29	895	$*$	$*$	$*$	$*$	$*$	$*$
256	386.72	3098	$*$	$*$	$*$	$*$	$*$	$*$

Experimental Results



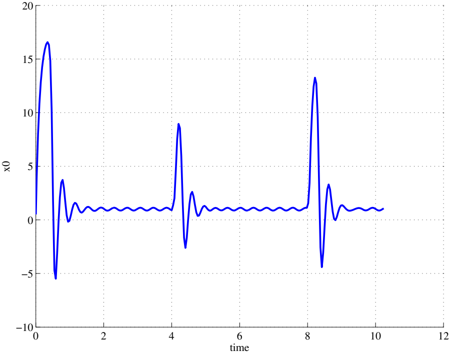
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	$*$	$*$
64	4.16	144	341.95	23103	308.93	10258	$*$	$*$
128	68.29	895	$*$	$*$	$*$	$*$	$*$	$*$
256	386.72	3098	$*$	$*$	$*$	$*$	$*$	$*$

Experimental Results



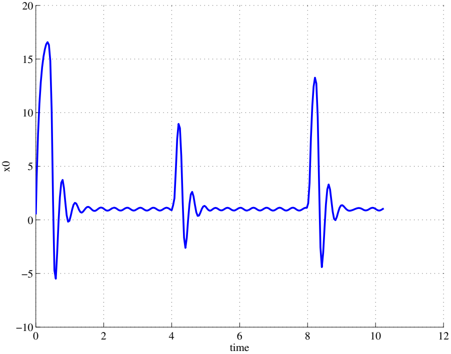
	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results



	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	*	*
64	4.16	144	341.95	23103	308.93	10258	*	*
128	68.29	895	*	*	*	*	*	*
256	386.72	3098	*	*	*	*	*	*

Experimental Results

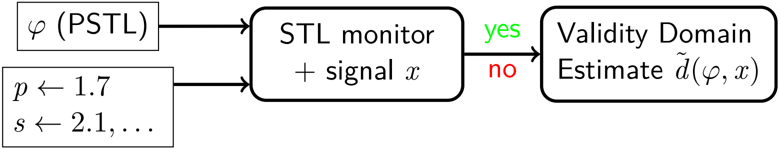


	φ_1		φ_2		φ_3		φ_{st}	
k	time(s)	size	time(s)	size	time(s)	size	time(s)	size
8	0.02	38	0.11	197	0.17	207	3	4219
16	0.10	66	0.81	855	0.74	375	83.79	37709
32	0.26	86	19.07	6553	18.27	2885	$*$	$*$
64	4.16	144	341.95	23103	308.93	10258	$*$	$*$
128	68.29	895	$*$	$*$	$*$	$*$	$*$	$*$
256	386.72	3098	$*$	$*$	$*$	$*$	$*$	$*$

Outline

General Approach

Estimate the set $d(\varphi, x)$ using a finite number of queries to an STL monitor with different values for the parameters

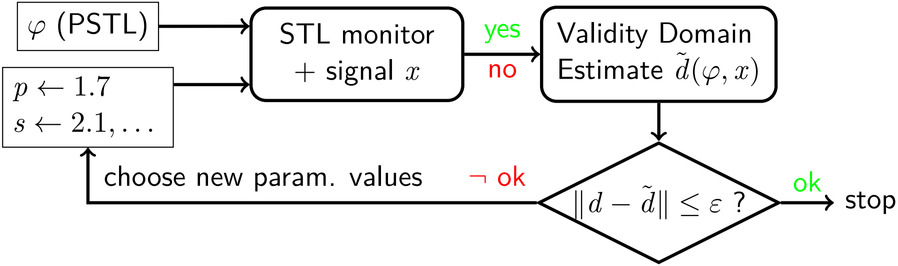


Motivations

- Scalability issues with the QE based approach
- STL (i.e., instantiated PSTL), formulas can be monitored efficiently

General Approach

Estimate the set $d(\varphi, x)$ using a finite number of queries to an STL monitor with different values for the parameters

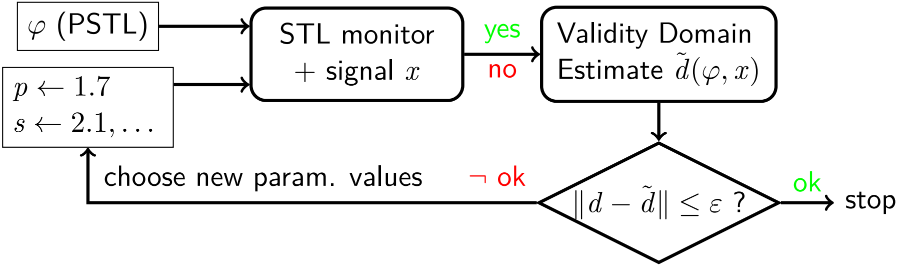


Motivations

- Scalability issues with the QE based approach
- STL (i.e., instantiated PSTL), formulas can be monitored efficiently

General Approach

Estimate the set $d(\varphi, x)$ using a finite number of queries to an STL monitor with different values for the parameters



Motivations

- Scalability issues with the QE based approach
- STL (i.e., instantiated PSTL), formulas can be monitored efficiently

Monitoring Computational Cost[1]

(a) Signal x, formula $\varphi = (x > 0) \underbrace{\mathcal{U}_{[0,1]}\ (x > 0)\ \mathcal{U}_{[0,1]}\ (x > 0) \ldots}_{i \text{ times}}$

(b) Formula φ_{st} with different number of samples

(a)

i	time(s)
1	0.34747
2	0.46335
3	0.60599
4	0.76067
5	0.89201
6	1.03761

(b)

Nb of samples	time(s)
31416	0.18402
345566	0.40761
659716	0.75508
973866	1.09268
1288016	1.4587

- However, the approximation scales exponentially with the number of queries.
- In certain cases, efficient heuristics are possible

[1] `http://www-verimag.imag.fr/~donze/breach_page.html`

Monitoring Computational Cost[1]

(a) Signal x, formula $\varphi = (x > 0)\ \underbrace{\mathcal{U}_{[0,1]}\ (x > 0)\ \mathcal{U}_{[0,1]}\ (x > 0) \ldots}_{i \text{ times}}$

(b) Formula φ_{st} with different number of samples

(a)

i	time(s)
1	0.34747
2	0.46335
3	0.60599
4	0.76067
5	0.89201
6	1.03761

(b)

Nb of samples	time(s)
31416	0.18402
345566	0.40761
659716	0.75508
973866	1.09268
1288016	1.4587

- However, the approximation scales exponentially with the number of queries.
- In certain cases, efficient heuristics are possible

[1] `http://www-verimag.imag.fr/~donze/breach_page.html`

Monitoring Computational Cost[1]

(a) Signal x, formula $\varphi = (x > 0) \underbrace{\mathcal{U}_{[0,1]}\ (x > 0)\ \mathcal{U}_{[0,1]}\ (x > 0) \ldots}_{i \text{ times}}$

(b) Formula φ_{st} with different number of samples

(a)

i	time(s)
1	0.34747
2	0.46335
3	0.60599
4	0.76067
5	0.89201
6	1.03761

(b)

Nb of samples	time(s)
31416	0.18402
345566	0.40761
659716	0.75508
973866	1.09268
1288016	1.4587

- However, the approximation scales exponentially with the number of queries.
- In certain cases, efficient heuristics are possible

[1] `http://www-verimag.imag.fr/~donze/breach_page.html`

Polarity

Definition (informal)

The *polarity* $\pi(p, \varphi)$ of a parameter p with respect to a formula φ is positive if it is easier to satisfy φ as we increase the value of p and negative otherwise

Polarity

Definition (informal)

The *polarity* $\pi(p, \varphi)$ of a parameter p with respect to a formula φ is positive if it is easier to satisfy φ as we increase the value of p and negative otherwise

Examples and remarks

- Magnitude parameters satisfy:

$$\pi(p, f(x) < p) = + \qquad \pi(p, f(x) > p) = -$$

- Temporal parameters satisfy:

$$\pi(s, \Diamond_{[a,s]}\mu) = + \qquad \pi(s, \Box_{[a,s]}\mu) = -$$

- Polarity can be deduced inductively on the structure of formulas
- It can be undetermined, e.g. in $\pi(p, (f(x) > p) \wedge (g(x) < p))$

Polarity

Definition (informal)

The *polarity* $\pi(p, \varphi)$ of a parameter p with respect to a formula φ is positive if it is easier to satisfy φ as we increase the value of p and negative otherwise

Examples and remarks

- Magnitude parameters satisfy:

$$\pi(p, f(x) < p) = + \qquad \pi(p, f(x) > p) = -$$

- Temporal parameters satisfy:

$$\pi(s, \Diamond_{[a,s]}\mu) = + \qquad \pi(s, \Box_{[a,s]}\mu) = -$$

- Polarity can be deduced inductively on the structure of formulas
- It can be undetermined, e.g. in $\pi(p, (f(x) > p) \wedge (g(x) < p))$

Polarity

Definition (informal)

The *polarity* $\pi(p, \varphi)$ of a parameter p with respect to a formula φ is positive if it is easier to satisfy φ as we increase the value of p and negative otherwise

Examples and remarks

- Magnitude parameters satisfy:

$$\pi(p, f(x) < p) = + \qquad \pi(p, f(x) > p) = -$$

- Temporal parameters satisfy:

$$\pi(s, \Diamond_{[a,s]}\mu) = + \qquad \pi(s, \Box_{[a,s]}\mu) = -$$

- Polarity can be deduced inductively on the structure of formulas
- It can be undetermined, e.g. in $\pi(p, (f(x) > p) \wedge (g(x) < p))$

Polarity

Definition (informal)

The *polarity* $\pi(p, \varphi)$ of a parameter p with respect to a formula φ is positive if it is easier to satisfy φ as we increase the value of p and negative otherwise

Examples and remarks

- Magnitude parameters satisfy:

$$\pi(p, f(x) < p) = + \qquad \pi(p, f(x) > p) = -$$

- Temporal parameters satisfy:

$$\pi(s, \Diamond_{[a,s]}\mu) = + \qquad \pi(s, \Box_{[a,s]}\mu) = -$$

- Polarity can be deduced inductively on the structure of formulas
- It can be undetermined, e.g. in $\pi(p, (f(x) > p) \wedge (g(x) < p))$

Monotonic Validity Domains

Definition

A subset $V \subseteq \mathcal{P} \times \mathcal{S}$ is monotonic if for every i, whenever a parameter valuation $(v_1, \ldots, v_i, \ldots, v_n)$ is in V so is any $(v_1, \ldots, v_i', \ldots, v_n) \in \mathcal{P} \times \mathcal{S}$ satisfying $v_i' > v_i$ (when $\pi(p_i, \varphi) = +$) or $v_i' < v_i$ (when $\pi(p_i, \varphi) = -$).

- If φ has only one monotonic param. p then $d(x, \varphi) \subset [p^*, \infty)$ for some p^*
- More generally the *boundary* of $d(x, \varphi)$ has the same properties as *Pareto fronts*

Monotonic validity domains can be estimated using heuristics generalizing dichotomic search in dim ≥ 1

In the following, we sketch a methodology adapted from

J. Legriel, C. Le Guernic, S. Cotton, O. Maler, *Approximating the Pareto Front of Multi-Criteria Optimization Problems*, TACAS 2010

Monotonic Validity Domains

Definition

A subset $V \subseteq \mathcal{P} \times \mathcal{S}$ is monotonic if for every i, whenever a parameter valuation $(v_1, \ldots, v_i, \ldots, v_n)$ is in V so is any $(v_1, \ldots, v_i', \ldots, v_n) \in \mathcal{P} \times \mathcal{S}$ satisfying $v_i' > v_i$ (when $\pi(p_i, \varphi) = +$) or $v_i' < v_i$ (when $\pi(p_i, \varphi) = -$).

- If φ has only one monotonic param. p then $d(x, \varphi) \subset [p^*, \infty)$ for some p^*
- More generally the *boundary* of $d(x, \varphi)$ has the same properties as *Pareto fronts*

Monotonic validity domains can be estimated using heuristics generalizing dichotomic search in dim ≥ 1

In the following, we sketch a methodology adapted from

J. Legriel, C. Le Guernic, S. Cotton, O. Maler, *Approximating the Pareto Front of Multi-Criteria Optimization Problems*, TACAS 2010

Monotonic Validity Domains

Definition

A subset $V \subseteq \mathcal{P} \times \mathcal{S}$ is monotonic if for every i, whenever a parameter valuation $(v_1, \ldots, v_i, \ldots, v_n)$ is in V so is any $(v_1, \ldots, v_i', \ldots, v_n) \in \mathcal{P} \times \mathcal{S}$ satisfying $v_i' > v_i$ (when $\pi(p_i, \varphi) = +$) or $v_i' < v_i$ (when $\pi(p_i, \varphi) = -$).

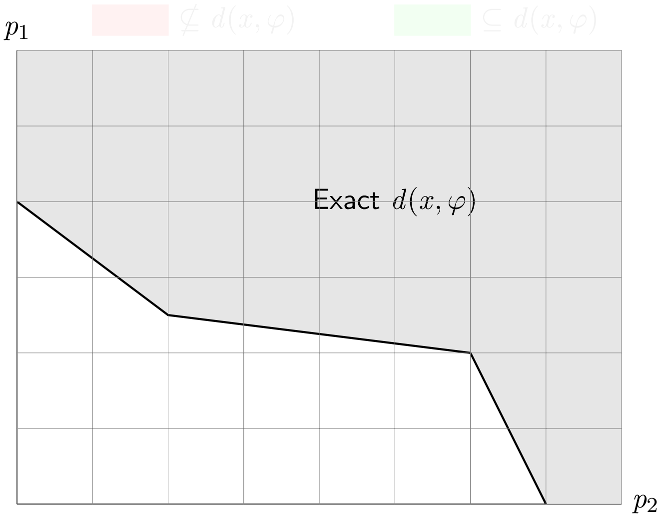
- If φ has only one monotonic param. p then $d(x, \varphi) \subset [p^*, \infty)$ for some p^*
- More generally the *boundary* of $d(x, \varphi)$ has the same properties as *Pareto fronts*

Monotonic validity domains can be estimated using heuristics generalizing dichotomic search in dim ≥ 1

In the following, we sketch a methodology adapted from

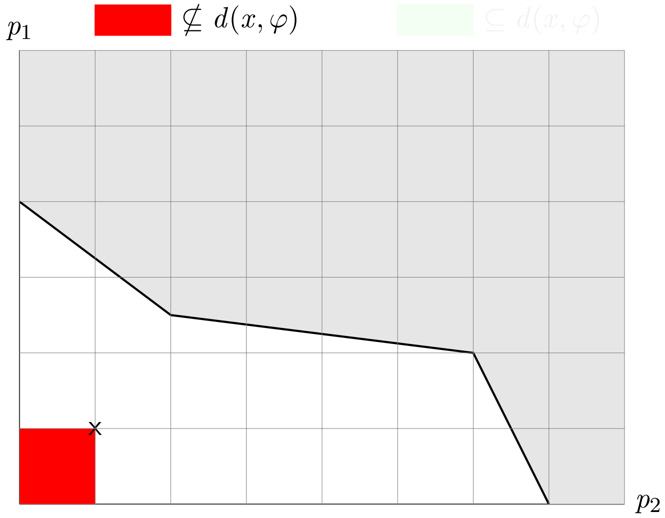
J. Legriel, C. Le Guernic, S. Cotton, O. Maler, *Approximating the Pareto Front of Multi-Criteria Optimization Problems*, TACAS 2010

Approximating Monotonic Validity Domains, Illustration



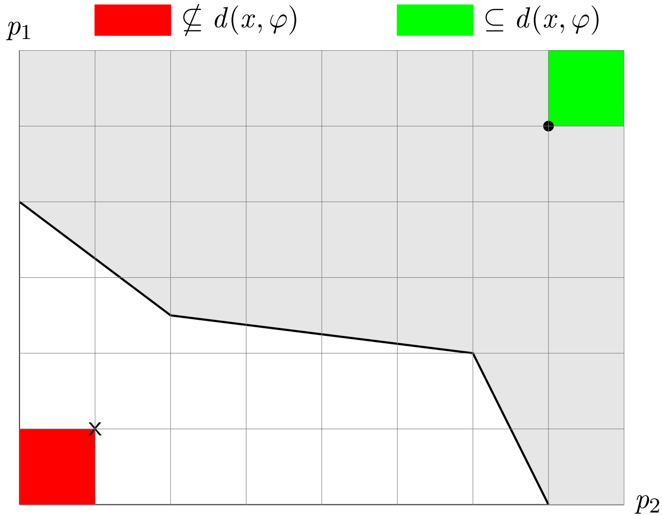
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



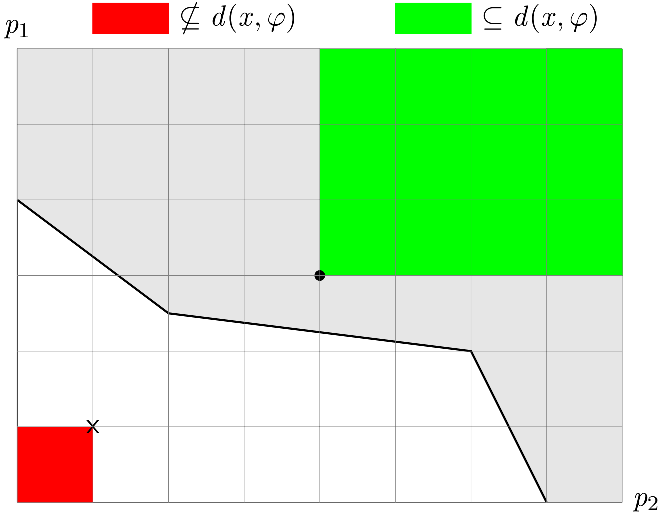
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



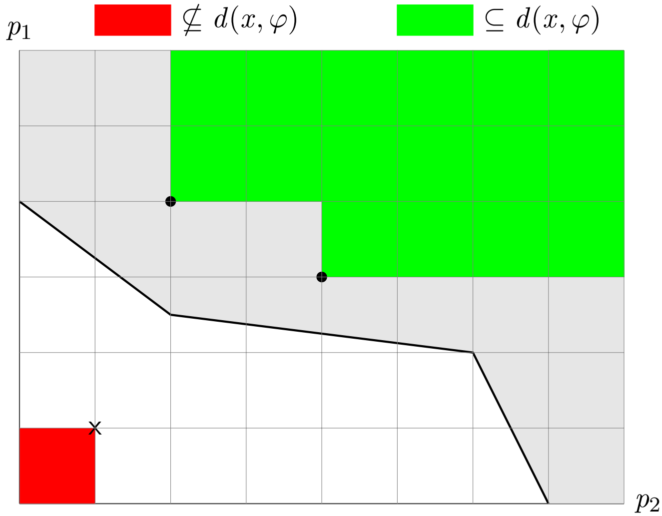
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



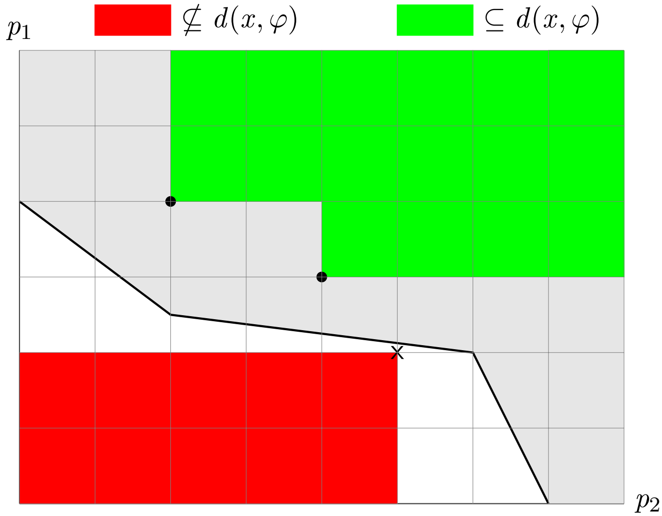
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



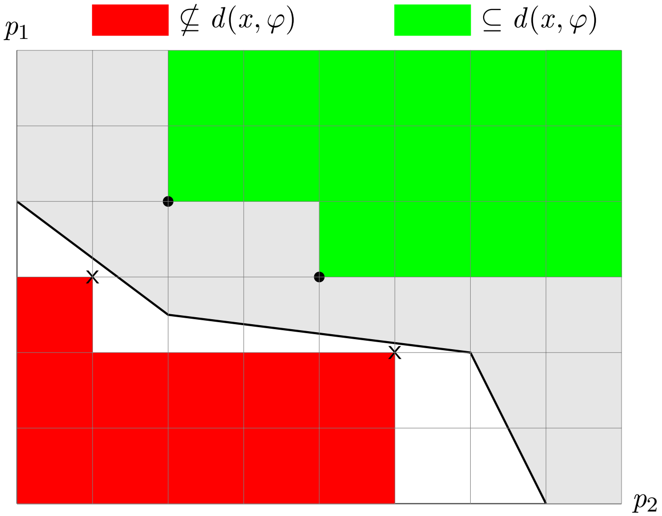
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



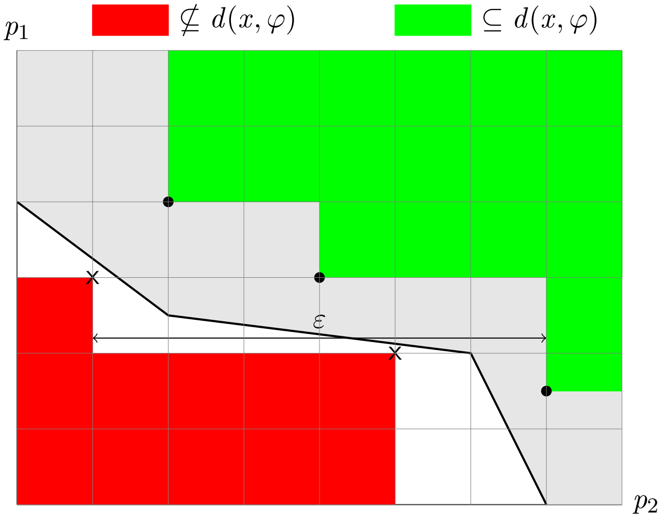
Different heuristics for reducing ϵ, distance between and .

Approximating Monotonic Validity Domains, Illustration



Different heuristics for reducing ϵ, distance between and .

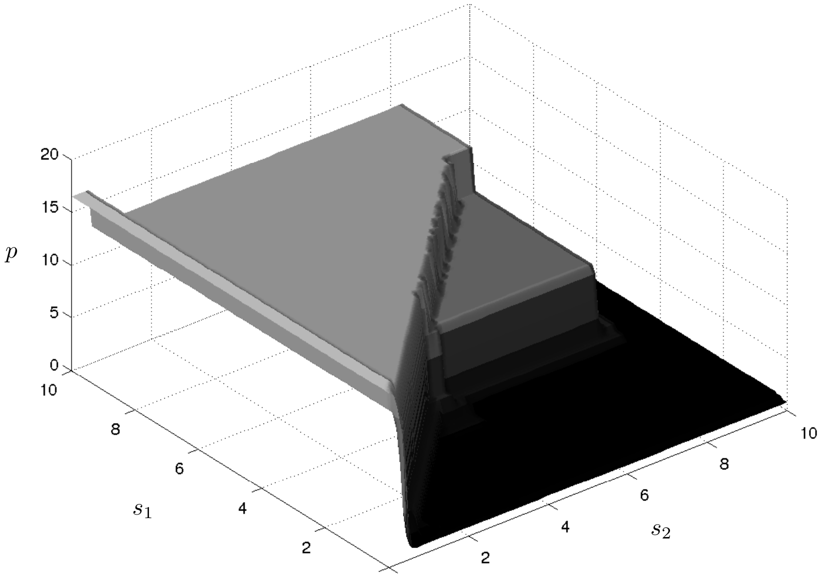
Approximating Monotonic Validity Domains, Illustration



Different heuristics for reducing ϵ, distance between [green] and [red].

Example

Stabilization property, $d(x, \varphi_{st})$ for x with 1024 samples



Conclusion

A problem potentially useful in many situations : given simulation traces, what properties do the system satisfy ?

Exact computation of validity domains using quantifier elimination

- Elegant but scalability problems with the size of x and φ
- What can be done beyond our "naive" approach ?

Numerical approximation using efficient monitoring

- Scales in $|x|$ and $|\varphi|$ but exponential in number of parameters
- Efficient heuristics in the case monotonic domains
- Applicable in the general case (including generic parameters, non-linear predicates, etc) but only to find *local* boundaries

Thanks for your attention !

Conclusion

A problem potentially useful in many situations : given simulation traces, what properties do the system satisfy ?

Exact computation of validity domains using quantifier elimination

- Elegant but scalability problems with the size of x and φ
- What can be done beyond our "naive" approach ?

Numerical approximation using efficient monitoring

- Scales in $|x|$ and $|\varphi|$ but exponential in number of parameters
- Efficient heuristics in the case monotonic domains
- Applicable in the general case (including generic parameters, non-linear predicates, etc) but only to find *local* boundaries

Thanks for your attention !

Conclusion

A problem potentially useful in many situations : given simulation traces, what properties do the system satisfy ?

Exact computation of validity domains using quantifier elimination

- Elegant but scalability problems with the size of x and φ
- What can be done beyond our "naive" approach ?

Numerical approximation using efficient monitoring

- Scales in $|x|$ and $|\varphi|$ but exponential in number of parameters
- Efficient heuristics in the case monotonic domains
- Applicable in the general case (including generic parameters, non-linear predicates, etc) but only to find *local* boundaries

Thanks for your attention !

Conclusion

A problem potentially useful in many situations : given simulation traces, what properties do the system satisfy ?

Exact computation of validity domains using quantifier elimination

- Elegant but scalability problems with the size of x and φ
- What can be done beyond our "naive" approach ?

Numerical approximation using efficient monitoring

- Scales in $|x|$ and $|\varphi|$ but exponential in number of parameters
- Efficient heuristics in the case monotonic domains
- Applicable in the general case (including generic parameters, non-linear predicates, etc) but only to find *local* boundaries

Thanks for your attention !