First-Order Temporal Properties of Continuous Signals

Thomas Ferrère, IST Austria

13 July 2018

Joint work with Alexey Bakhirkin,
Tom Henzinger, and Dejan Nickovic

A Brief History: Runtime Verification for Rigorous Systems Engineering

- LTL Monitoring (Kim & al. 1999, Havelund & Rosu 2001)
- Signal Temporal Logic (Maler & Nickovic 2004)
- LTL Robust Monitoring (Fainekos & Pappas 2006)
- STL Robust Monitoring (Fainekos & Pappas 2009, Donzé & Maler 2010)
- STL Parametric Identification (Asarin & al. 2011)
- Robustness-Based Falsification (Sankaranarayanan & Fainekos 2012)

Research Objective: Making RV 4 RISE Practical

- Efficient Robust Monitoring (Donzé & F & Maler 2013)
- Timed Pattern Matching (Ulus & al. 2014)
- Trace Diagnostics (F & Maler & Nickovic 2015)
- Pattern-Based Measurements and Robustness (F & al. 2015, Bakhirkin & al. 2017)
- Efficient Parametric Identification (Bakhirkin & F & Maler 2018)

Parametric STL

- PSTL Syntax

$$\varphi ::= f \sim c \mid \varphi_1 \vee \varphi_2 \mid F_{[a,b]}\, \varphi \mid \varphi_1 U \varphi_2$$

- PSTL Semantics

$$(w, v, t) \vDash \varphi$$

where

- w is trace
- v is valuation of parameters
- t is absolute time

Example

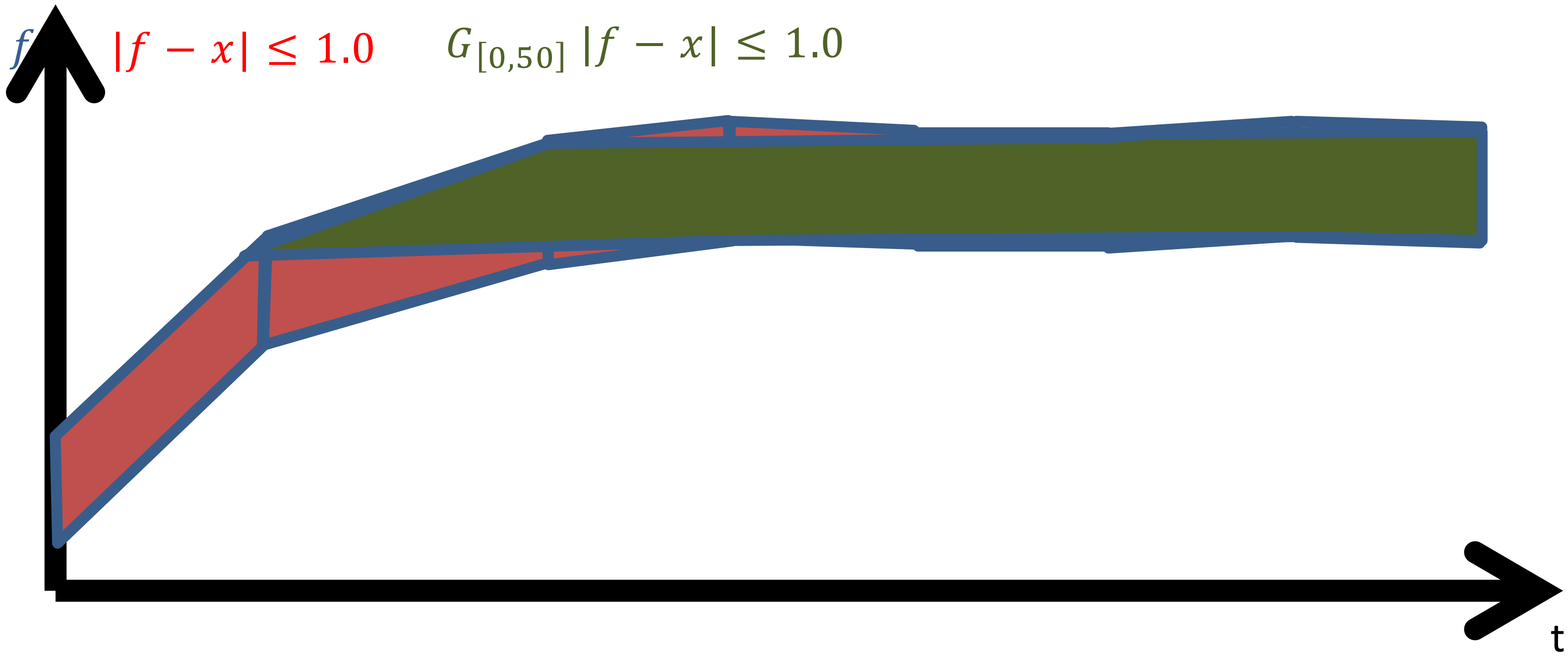
- PSTL Property:

$$G_{[0,50]}\ |f - x| \leq 1.0$$

- Meaning: signal f stays within 1.0 of value x for 50 time units
- Output of Parametric Identification: set of valuations of x that make the formula true on a given trace

Efficient Parametric Identification

- Compute the set of valuations $(v, t) \in \mathbb{R}^{k+1}$ such that $(w, v, t) \vDash \varphi$ as convex polyhedra
- Example:



Quantified Signal Temporal Logic

- Instead of measuring parameter x, just state
$$\exists x : G_{[0,5]} \, |f - x| \leq 1.0$$
- Meaning: there exists x such that f stabilizes within 1.0 of x for 5 time units
- Quantifiers can be nested with temporal operators
- Example: control
$$\exists x : G_{[0,10]} \, |f - x| \leq 1.0$$
$$\rightarrow \exists y : |x - y| \leq 2.0 \wedge G_{[5,10]} \, |g - y| \leq 1.0$$

QSTL Definitions

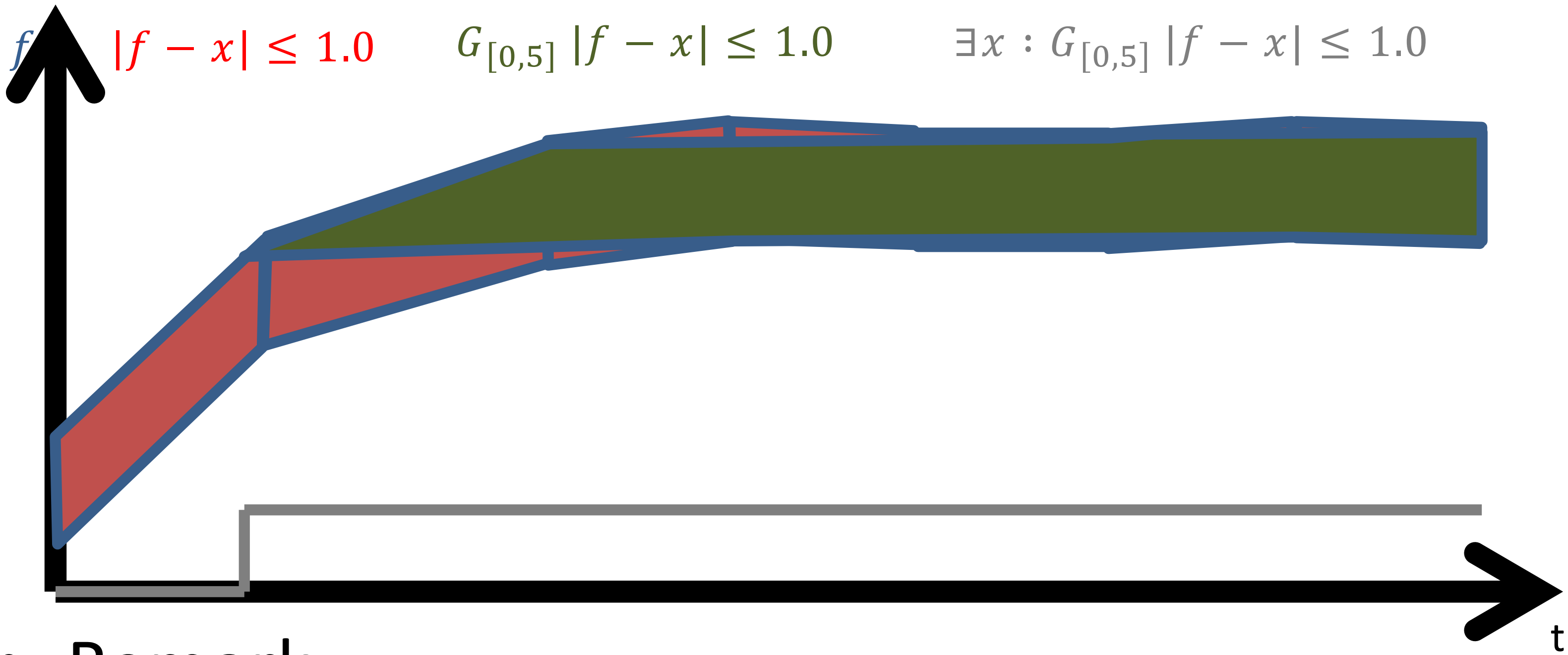
- Syntax:
  $\varphi ::= f \sim c \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid F_{[a,b]}\, \varphi$
  $\mid \varphi_1 U \varphi_2 \mid \exists x : \varphi$
- Here a, b, c, are constants or parameters
- Semantics:

$(w, v, t) \vDash \exists x : \varphi$ iff $(w, v[x \leftarrow u], t) \vDash \varphi$

- Other cases are as usual

Efficient (?) QSTL Monitoring

- Use same algorithm as for PSTL



- Remark:
  - piecewise-constant signals -> box polyhedra
  - Piecewise linear signals -> arbitrary polyhedra

Time Variables vs Temporal Operators

- Remark that $F_{[d,d]}\varphi$ holds at t iff φ holds at $t + d$
- Thus operator "until" becomes redundant:

$$\varphi U \psi$$
$$\Leftrightarrow$$
$$\exists d > 0 : \; F_{[d,d\,]}\psi \wedge \forall c \in (0, d) : F_{[c,c]}\varphi$$

- In fact, $F_{[d,d]}$ is the only operator we need
- But then, why use temporal logic?

The First-Order Logic of Signals

- Variables $x, y, \ldots$
- Function symbols $f, g, \ldots$
- SFO Syntax:
  $$\theta ::= n \mid x \mid f(\theta) \mid \theta_1 - \theta_2 \mid \theta_1 + \theta_2$$
  $$\varphi ::= \theta_1 < \theta_2 \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x : \varphi$$
- SFO Semantics: linear arithmetic over piecewise linear signal interpretations of function symbols

Examples of SFO Formulas

Example 3 (Rise Time). *Consider the following property:*

$$\varphi_3 \equiv \forall c : \left(\begin{array}{l} f(t) = 1 \wedge f(t+c) = 2 \\ \wedge\ \forall t' \in (t, t+c) : 1 < f(t') < 2 \end{array} \right)$$

$$\rightarrow \exists t' \exists d : \left(\begin{array}{l} t < t' \wedge 9c \leq 10d \leq 11c \\ \wedge\ \forall t'' \in (t, t') : g(t'') < 1 \\ \wedge\ g(t') = 1 \wedge g(t'+d) = 2 \\ \wedge\ \forall t'' \in (t', t'+d) : 1 < g(t'') < 2 \end{array} \right)$$

Formula φ_3 expresses the fact that if signal f has a positive edge (from 1.0 *to* 2.0*) then signal g subsequently has a positive edge whose rise time is within* 10% *of that of f.*

Basic Properties of SFO

- Satisfiability is undecidable
  - Over piecewise-linear signals
  - Over a bonded time domain
  - Restricted to linear order, or difference logic over Boolean signals
- Membership (of PWL signal in language of SFO formula) is decidable

Remark: membership of Signal Second-Order logic is undecidable

Complexity of Membership for PWL Signals

- Decidable in time $(m+n)^{2^{O(k+l)}}$
  - m = size of formula
  - n = size of trace
  - k = number of quantifiers
  - l = number of function symbols
- Proof:
  - translate signal into linear real arithmetic formula
  - conjunct with SFO formula
  - Eliminate quantifiers

May not scale with **large signals**

Bounded-Time Formulas

- Separate variables between
  - absolute time variable t
  - delays variables $d \in D$
  - space variables $r \in R$
- Syntax

$$\delta ::= d \mid n \mid \delta_1 - \delta_2 \mid \delta_1 + \delta_2$$

$$\rho ::= r \mid f(t + \delta) \mid n \mid \rho_1 - \rho_2 \mid \rho_1 + \rho_2$$

$$\varphi ::= \delta_1 < \delta_2 \mid \rho_1 < \rho_2 \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid$$

$$\exists\, d \in I : \varphi \mid \exists r : \varphi$$

Monitoring

- Problem: compute the Boolean *satisfaction signal* of some formula φ relative to a piecewise-linear signal w
- Solution:
  - Inductively on the formula structure
  - Represent satisfaction set as list of polytopes
  - Order polytopes in time

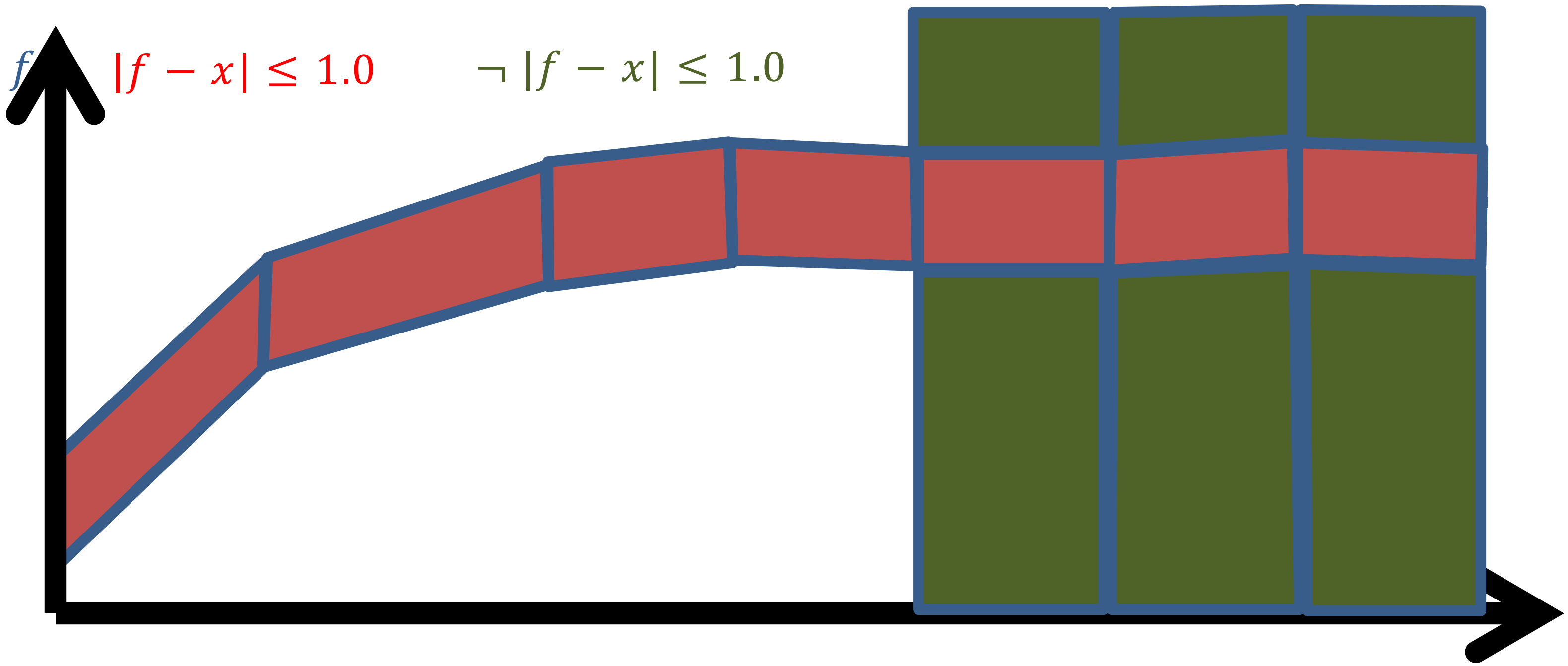
Efficient when in every slice of time the satisfaction set has few non-convex parts

Algorithms

function $formula(\varphi, P_{\text{diff}})$
  if $\varphi = \rho < \rho_2$ **then**
    $\mathcal{P}, v \leftarrow term(\rho_1 - \rho_2, P_{\text{diff}})$
    return $eliminate(v, \mathcal{P} \sqcap (v < 0))$
  else if $\varphi = \neg\varphi'$ **then**
    $\mathcal{P} \leftarrow formula(\varphi', P_{\text{diff}})$
    return $P_{\text{diff}} \sqcap complement(\mathcal{P})$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    $\mathcal{P}_1 \leftarrow formula(\varphi_1, P_{\text{diff}})$
    $\mathcal{P}_2 \leftarrow formula(\varphi_2, P_{\text{diff}})$
    return $\mathcal{P}_1 \sqcup \mathcal{P}_2$
  else if $\varphi = \exists x.\ \varphi'$ **then**
    $\mathcal{P} \leftarrow formula(\varphi', P_{\text{diff}})$
    return $eliminate(x, \mathcal{P})$
  else if $\varphi = \exists d \in I.\ \varphi'$ **then**
    $\mathcal{P} \leftarrow formula(\varphi', P_{\text{diff}} \sqcap (d \in I))$
    return $eliminate(d, \mathcal{P})$
  end
end

Complementation

- Idea: proceed in time-ordered manner
- Complement each polytope over its time footprint



Complexity on Bounded-Time SFO

- Decidable in time $n2^{(m+h)^{2^{O(k+l)}}}$
  - n = size of trace
  - m = size of formula
  - k = number of quantifiers
  - l = number of function symbols
  - h = variability

DDR2 Case Study

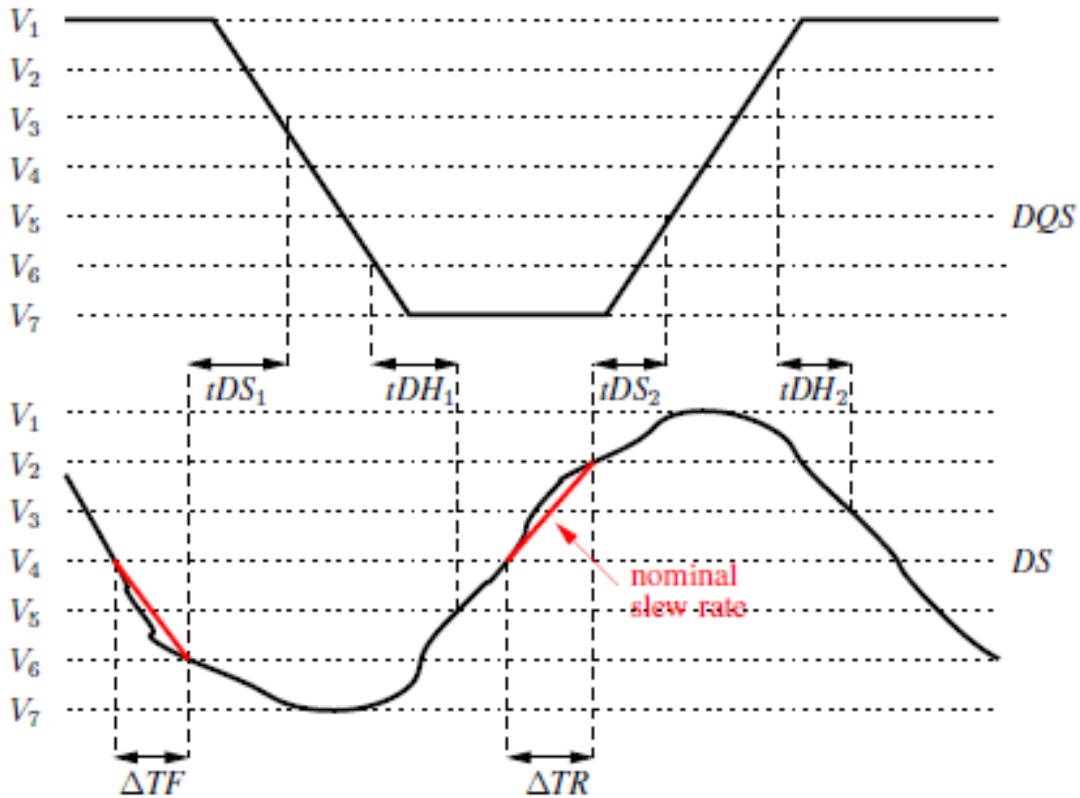
- Memory interface
- Requirement: alignment between data signal DS and data signal strobe DQS
- Involves setup times tDS and tDH
- Alignment defined according to crossing of thresholds:

Threshold name	Threshold id	Value (V)
V_{DDQ}	V_1	1.800
$V_{IH(AC)_{min}}$	V_2	1.250
$V_{IH(DC)_{min}}$	V_3	1.025
$V_{REF(DC)}$	V_4	0.900
$V_{IL(DC)_{max}}$	V_5	0.775
$V_{IL(AC)_{max}}$	V_6	0.650

- Falling edge of DQS = crossing V3 from above
- Falling edge of DS = crossing V6 from above

Alignment Property

Whenever DQS is on its falling edge, the distance from the previous falling edge in DS is at least tDS time



Formalization

- In STL:

$$\psi(t) \equiv \downarrow(DQS, V_3)(t) \rightarrow \boxminus_{[0,tDS]} \neg\downarrow(DS, V_6)(t)$$

- Problem: tDS is **not constant**

$$\Delta tDS = tDS - tDS(base)$$

- Should be linearly interpolated according to:

		DQS slew rate (V/ns)				
		2	1.5	1	0.9	0.8
DS slew rate (V/ns)	2	188	146	63	–	–
	1.5	167	125	42	43	–
	1	125	83	0	−2	−13
	0.9	–	69	−14	−13	−27
	0.8	–	–	−31	−30	−44

Formalization

- In SFO:

$$\psi'(t) \equiv \downarrow(DQS, V_3)(t) \rightarrow \exists \Delta TF_{DQS}, \Delta TF_{DS} :$$
$$\ominus^{V_6,V_4}_{\Delta TF_{DS}}(DS)(t) \wedge \bigcirc^{V_6,V_4}_{\Delta TF_{DQS}}(DQS)(t) \wedge$$
$$tDS = \delta(\Delta TF_{DQS}, \Delta TF_{DS}) \wedge \boxminus_{[0,tDS]} \neg \downarrow(DS, V_6)(t)$$

where

$$\ominus^{c,c'}_{T}(s)(t) \equiv \exists t', t'' : t'' < t' < t \wedge \forall \tau \in (t', t) :$$
$$s(\tau) < c \wedge s(t') = c \wedge \forall \tau \in (t'', t') :$$
$$s(\tau) \in (c, c') \wedge s(t'') = c' \wedge t' - t'' = T$$

$$dqs\_fall \equiv \downarrow(DQS \geq V_3) \qquad dq\_fall \equiv \downarrow(DS \geq V_6)$$

Conclusion

- A powerful formalism for specifying properties of real-valued signals: First-Order Logic with Linear Arithmetic and uninterpreted unary funtion symbols.
  - Undecidable satisfiability problem
  - Decidable (doubly exponential) membership problem
  - Bounded-time fragment can be monitored in linear time relative to trace length
  - Captures complex requirements of analog circuits not monitored in practice

Conclusion

- Prototype C++ implementation using PPLite, a new open-source software library re-implementing functionality of the Parma Polyhedra Library
- Preliminary experiments: can monitor signals up to 1000 samples in a few seconds
- Open questions:
  - Scalability on real examples
  - Theoretical and practical expressiveness relative to temporal logic (and regular expression) variants