Cyber-Physical Systems

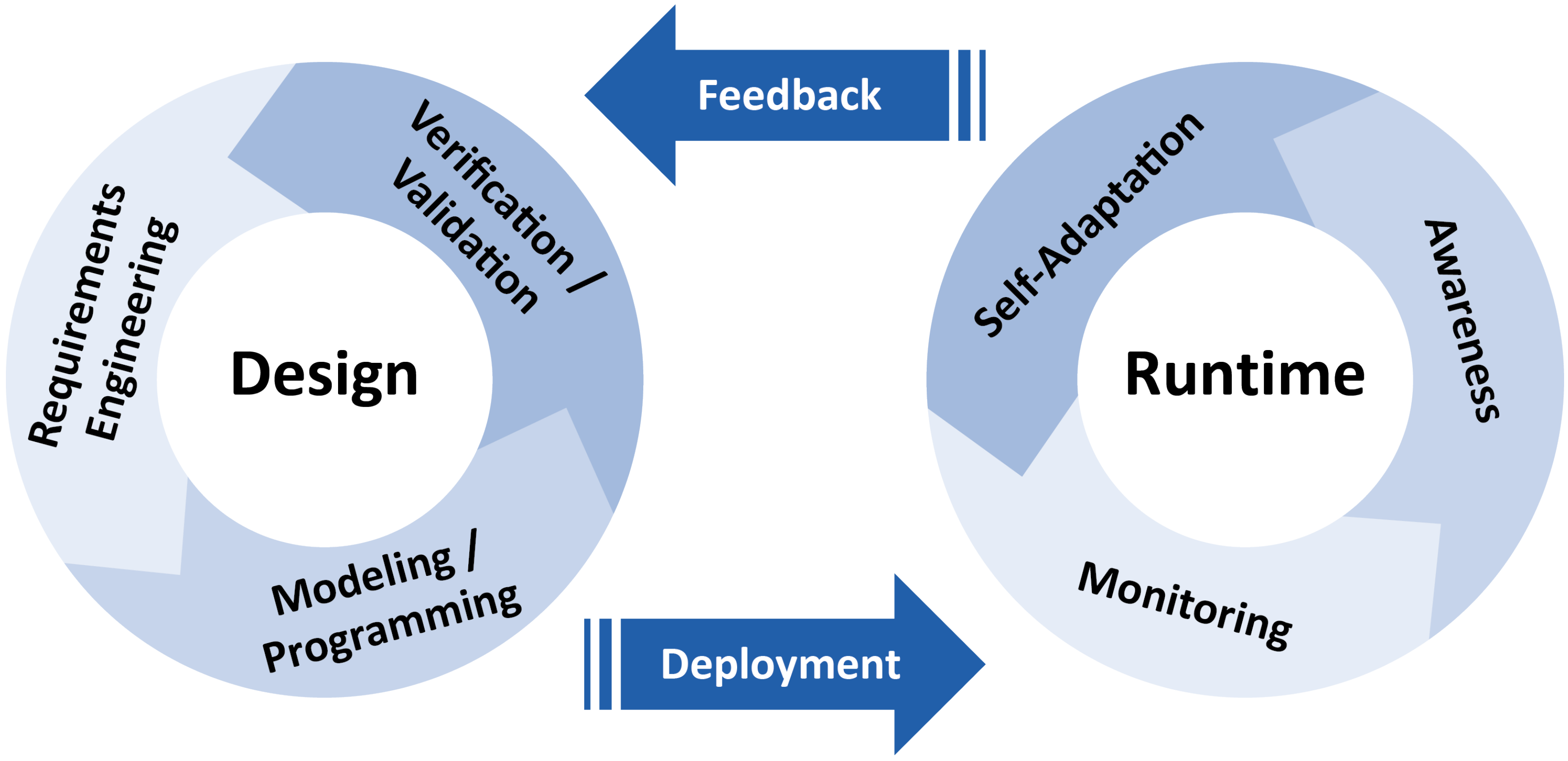
Laura Nenzi

Università degli Studi di Trieste
II Semestre 2019

Lecture 9: Signal Temporal Logic

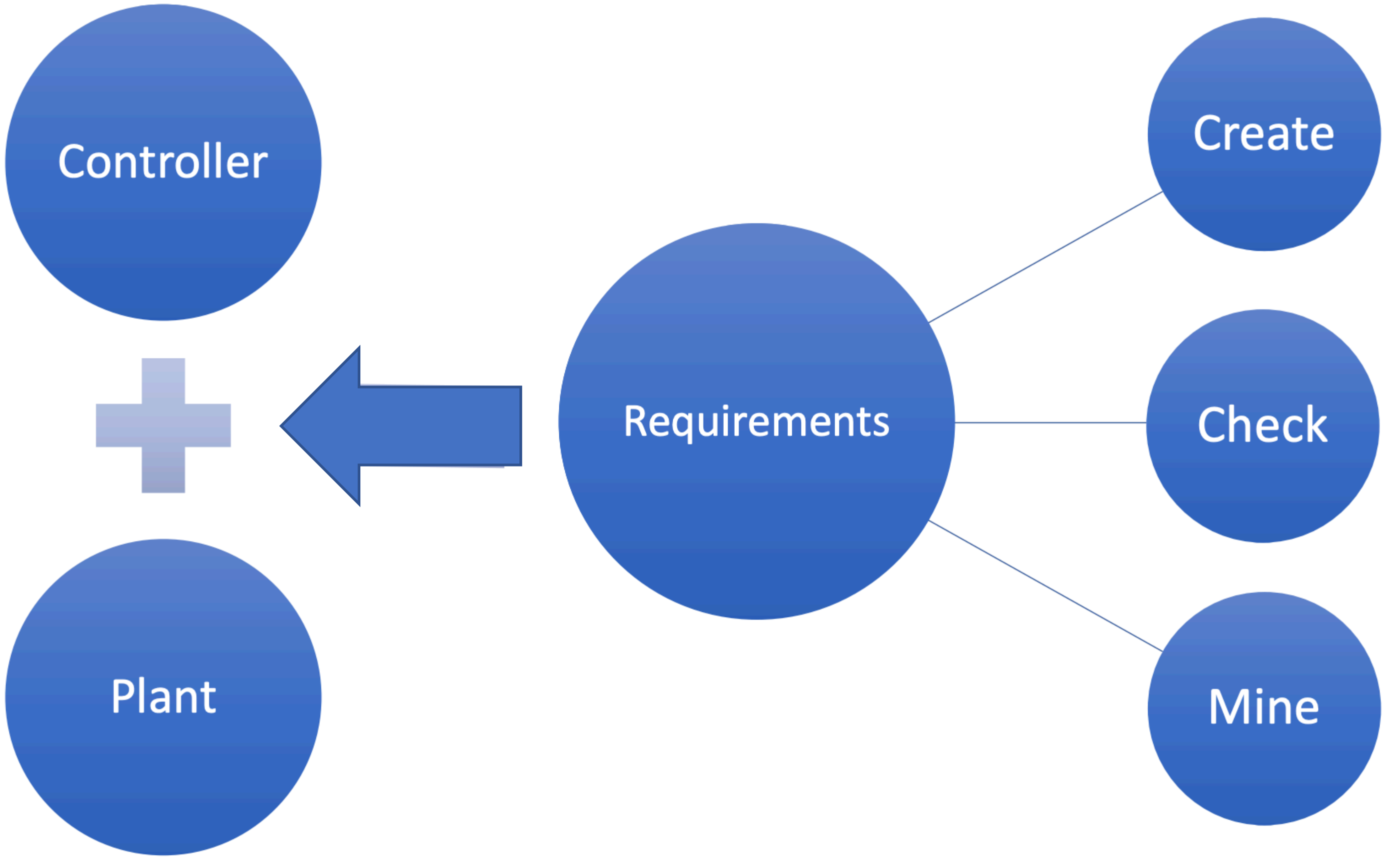
[Many Slides due to J. Deshmukh, A. Donzé]

Model-based Design Approach

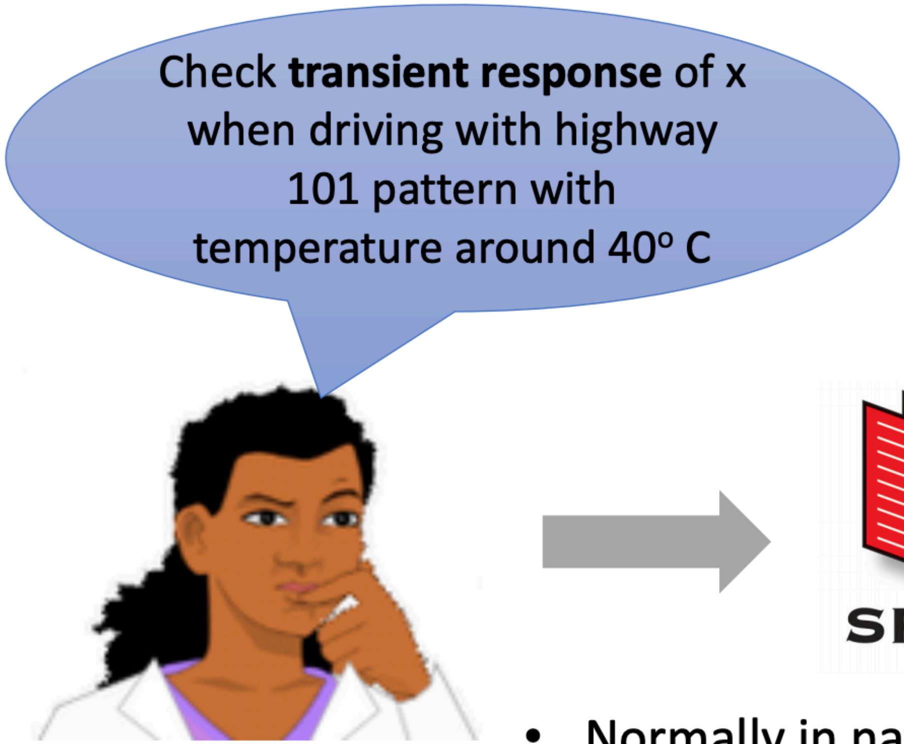


Requirements Driving Design

Requirements **formally** capture what it means for a system to operate correctly in its operating environment



Typical day in a control designer's life

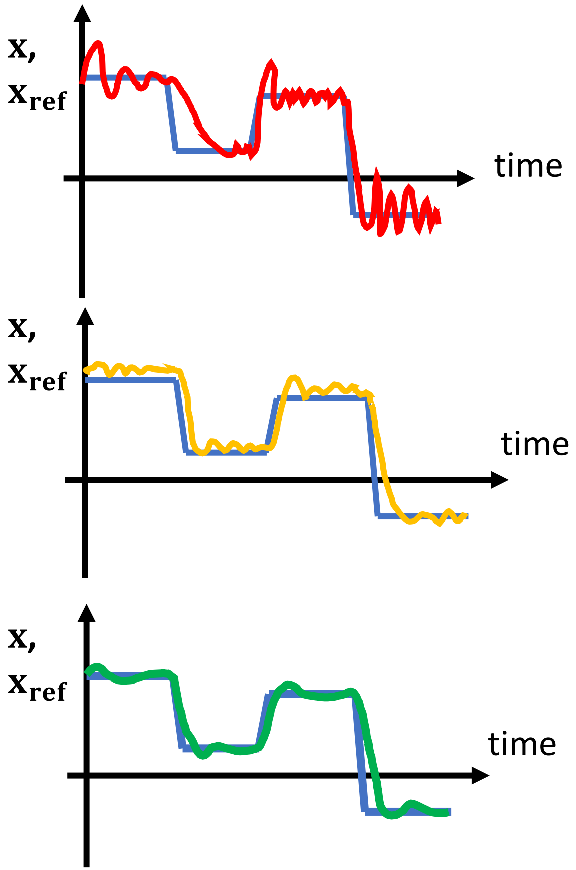


Chief Engineer

Control Designer

- Normally in natural language (ambiguous, error-prone)
- Sometime absent
- If you are LUCKY, they are written in English

Typical day in a control designer's life



Linear Temporal Logic (LTL) specification

It is a logic interpreted over infinite discrete-time traces

E.g. It is always true that the highest temperature will be below 75 degree and the lowest temperature will be above 60 degree

$\mathbf{G}(p \wedge q)$ $p = T<75, q=T>60$

Linear Temporal Logic (LTL) specification

It is a logic interpreted over infinite discrete-time traces

E.g. **For the next 3 days** the highest temperature will be below 75 degree and the lowest temperature will be above 60 degree

$X\ (p \wedge q) \wedge X\ X\ (p \wedge q) \wedge X\ X\ X\ (p \wedge q)$ with $p = T<75,\ q=T>60$

Metric Interval Temporal Logic (STL)

Invented by R. Alur, T.Feder, T.A. Henzinger (1991)
It extended LTL by adding **dense time intervals**:

$$G_{[0,3]}(\mathrm{p} \wedge q)$$

Signal Temporal Logic (STL)

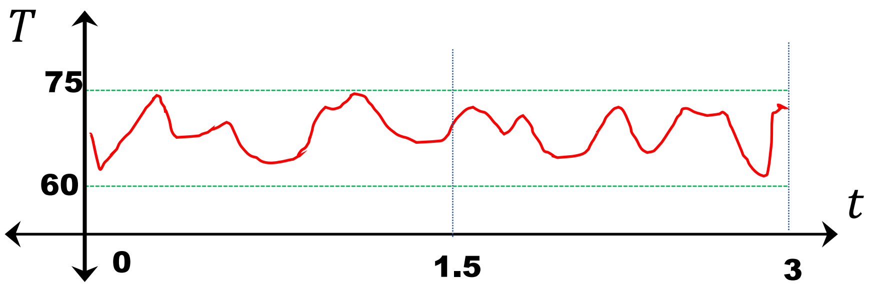
Invented by D. Nickovic and O. Maler from Verimag (2004)
It extended MITL by having **signal predicates over real values as atomic formulas:**

$$G_{[0,3]}(T < 75 \wedge T > 60)$$

Expressing specifications in STL

$\textbf{Always}_{[0,3]}\ (60 < T < 75)$

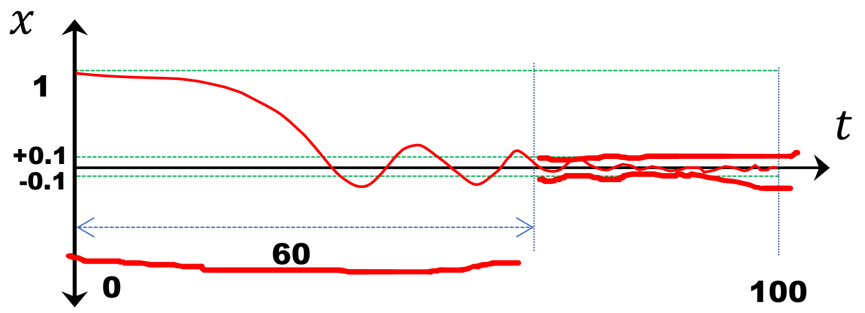
Always between time 0 and 3



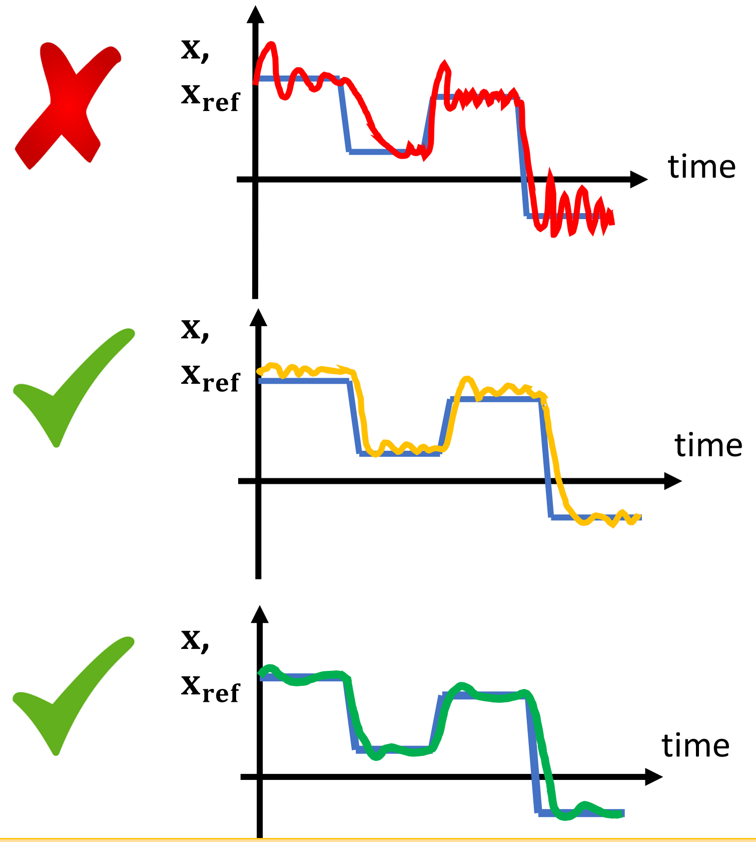
$\textbf{Eventually}_{[0,60]}(\textbf{Always}\ (|x| < 0.1)\ )$

Eventually at **some time** ***t*** between time 0 and 60

From that time ***t*****,** always till the end of the signal trace

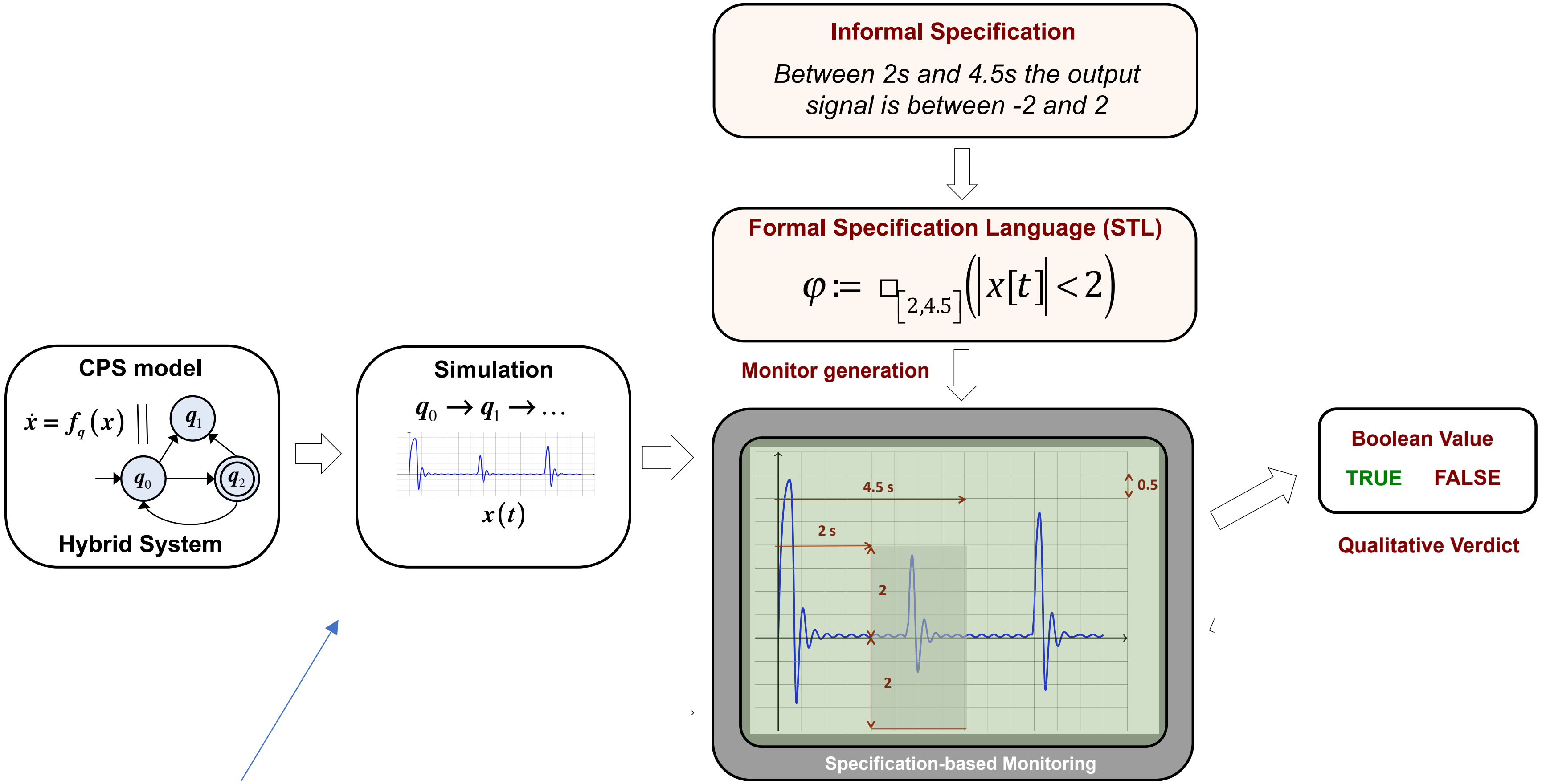


Can we express our engineer's requirements?



$$\varphi \equiv \mathrm{Alw}_{[0,10]}\left(\mathrm{step} \Rightarrow \mathrm{Alw}_{[0,2]}\left(\left|x - x_{ref}\right| < 0.05\right)\right)$$

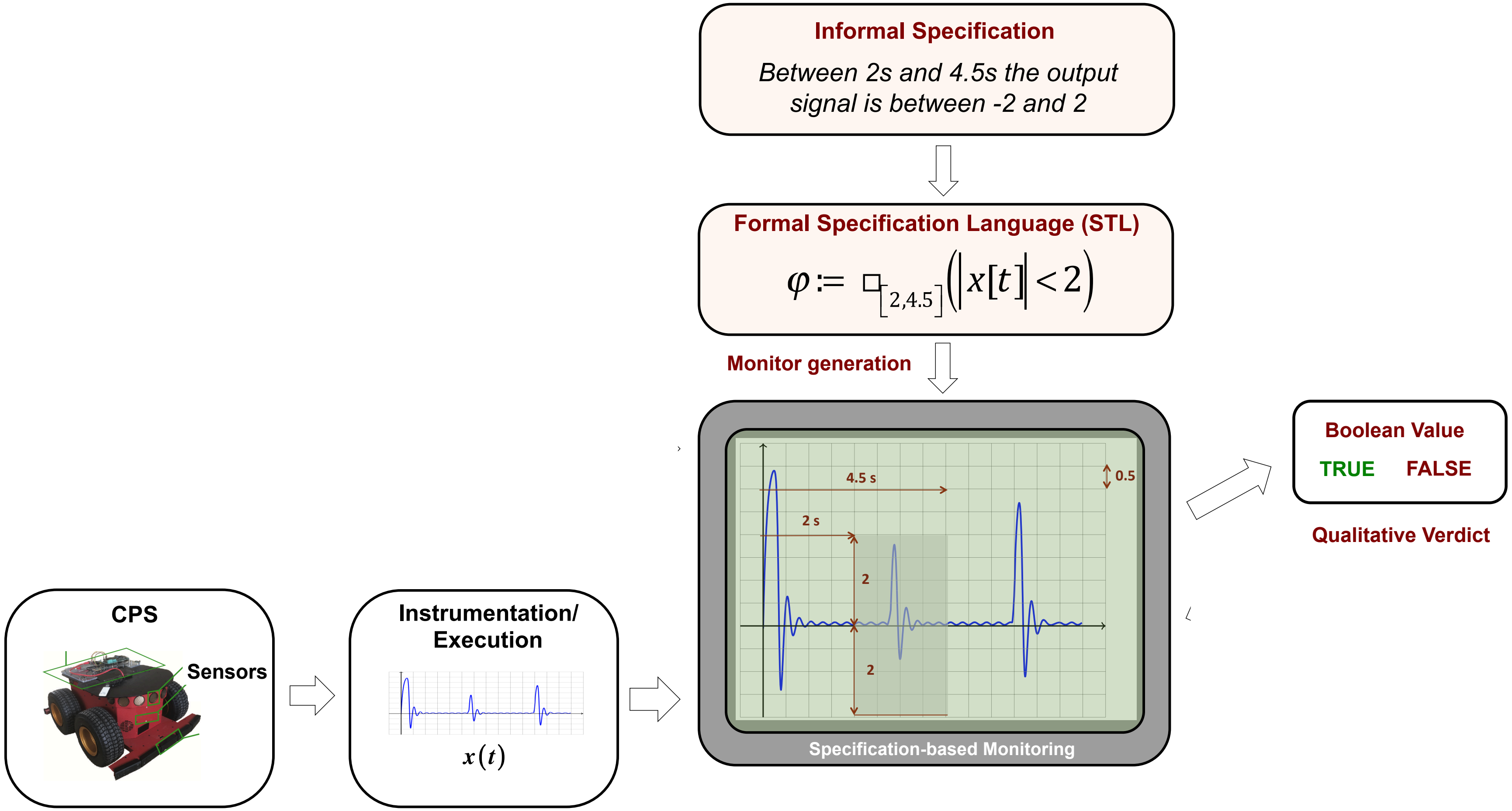
Specification-based Monitoring



Complex behaviours

Low dimensional vectors

Specification-based Monitoring



STL Syntax

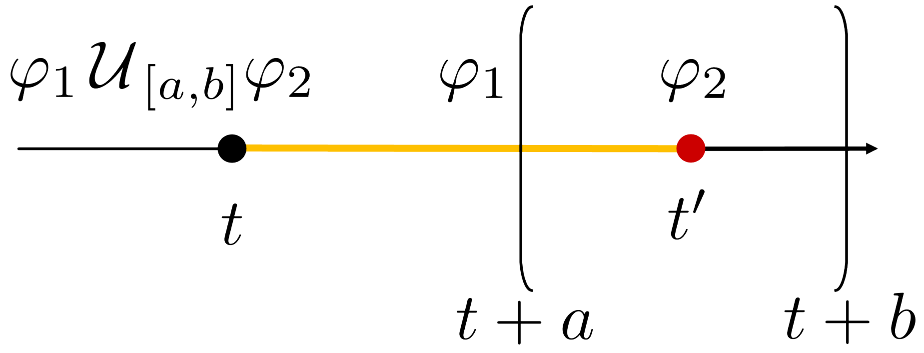
Syntax of STL		
$\varphi \ ::= \quad f(\mathbf{x}) \sim 0$	|	$f: \mathbb{D} \to \mathbb{R}$ is a function over the signal $\mathbf{x}$: $\mathbb{T} \to \mathbb{D}$, $\sim \in \{\leq, <, >, \geq, =, \neq\}$
$\neg\varphi$	|	Negation
$\varphi \wedge \varphi$	|	Conjunction
$\mathbf{F}_{[a,b]}\varphi$	|	At some **F**uture step in the interval $[a, b]$
$\mathbf{G}_{[a,b]}\varphi$	|	**G**lobally in all times in the interval $[a, b]$
$\varphi\ \mathbf{U}_{[a,b]}\ \varphi$	|	In all steps **U**ntil in interval $[a, b]$
$\varphi\ \boldsymbol{S}_{[a,b]}\ \varphi$	|	In all steps **S**ince in interval $[a, b]$

Recursive Boolean Semantics of STL

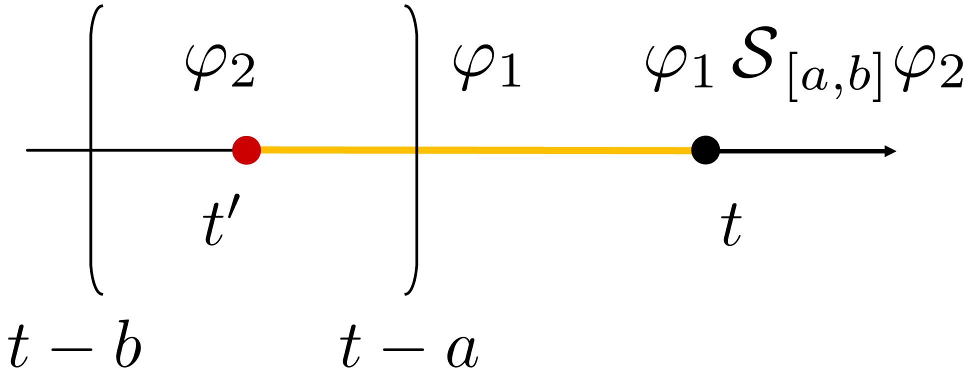
φ	$\beta(\varphi, \mathbf{x}, t)$
$f(\mathbf{x}) \sim 0$	$f(\mathbf{x}(t)) \sim 0$, $\quad \sim \in \{\leq, <, >, \geq, =, \neq\}$
$\neg\varphi$	$\neg\beta(\varphi, \mathbf{x}, t)$
$\varphi_1 \wedge \varphi_2$	$\beta(\varphi_1, \mathbf{x}, \mathrm{t}) \wedge \beta(\varphi_2, \mathbf{x}, \mathrm{t})$
$\mathbf{F}_{[a,b]}\varphi$	$\exists \tau \in [t+a, t+b]\ \beta(\varphi, \mathbf{x}, \tau)$
$\mathbf{G}_{[a,b]}\varphi$	$\forall \tau \in [t+a, t+b]\ \beta(\varphi, \mathbf{x}, \tau)$
$\varphi\ \mathbf{U}_{[a,b]}\ \psi$	$\exists \tau \in [t+a, t+b]\ \left(\beta(\psi, \mathbf{x}, \tau) \wedge \forall \tau' \in [t, \tau)\ \beta(\varphi, x, \tau')\right)$
$\varphi\ \mathbf{S}_{[a,b]}\ \psi$	$\exists \tau \in [t-a, t-b]\ \left(\beta(\psi, \mathbf{x}, \tau) \wedge \forall \tau' \in (\tau, t]\ \beta(\varphi, x, \tau')\right)$

Since and Until Operators

- Until



- Since

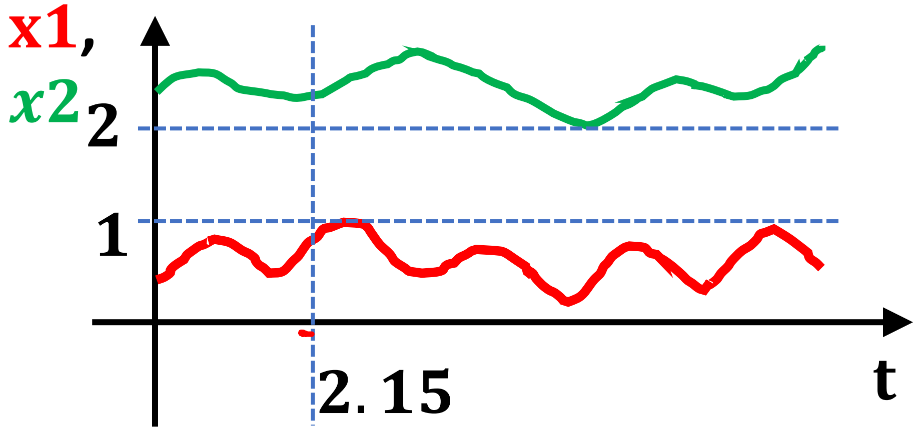


STL semantics

- Semantics of STL specified recursively over a signal $\mathbf{x}: \mathbb{T} \to \mathbb{D}$ at each time,

For each STL formula φ, here's how we define it's semantics:

- If φ is the signal predicate $\mu = f(\mathbf{x}) > 0$, then $\beta(\varphi, \mathbf{x}, t) = true$ iff $f\big(\mathbf{x}(t)\big) > 0$



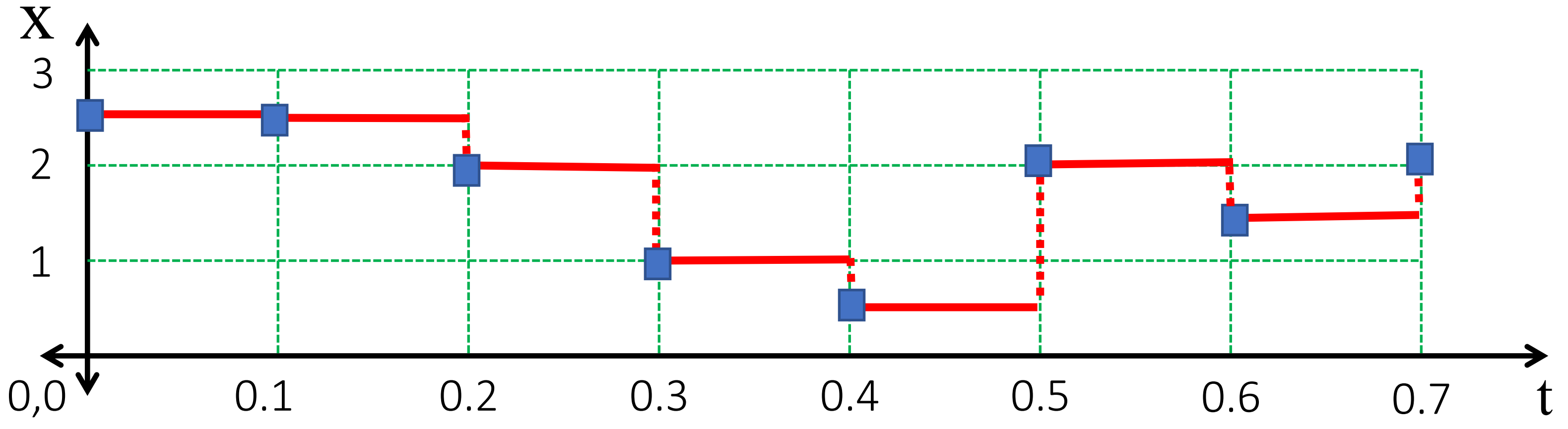
$$\mathbf{x} = (x1, x2)$$

$$\boldsymbol{f} = x2 - x1 - 1$$

$$\beta(f(\mathbf{x}) > 0, \mathbf{x}, 2.15)?$$

Recursive Boolean Semantics of STL

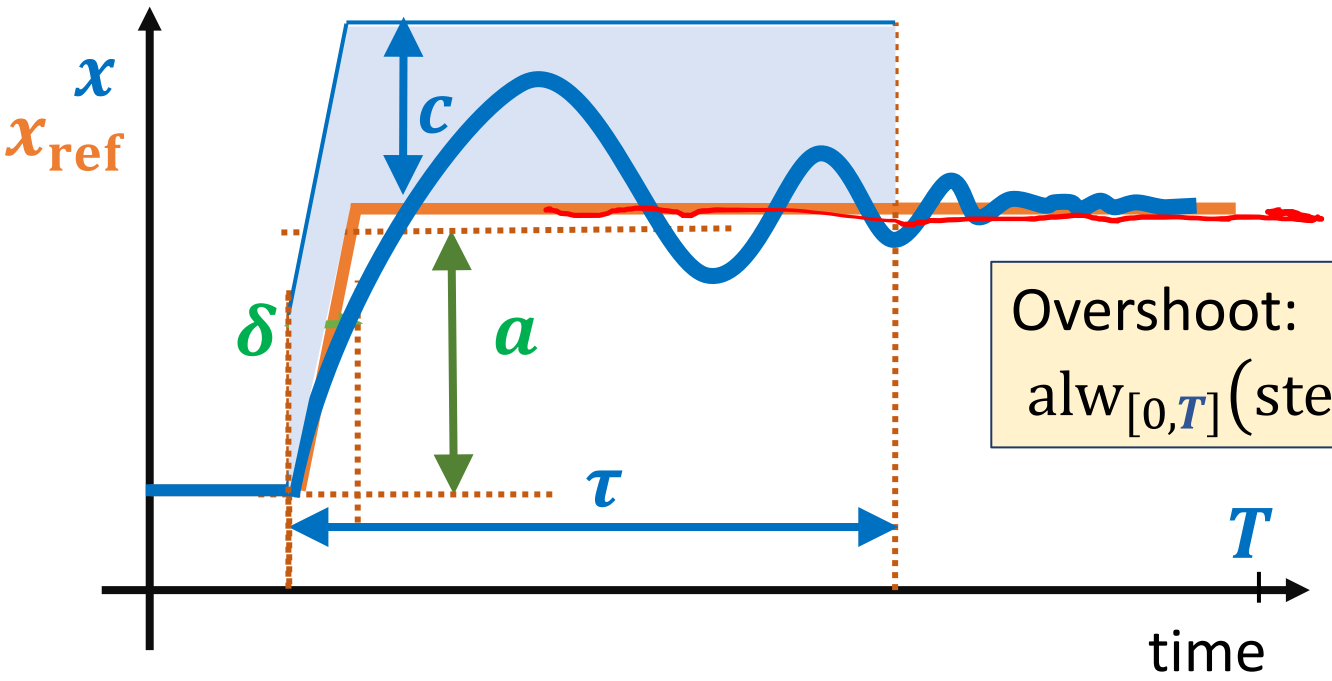
$\varphi \equiv \mathbf{F}_{[0,0.2]}(x(t) \geq 1.5)$



$x(t) - 1.5 > 0$	T	T	T	F	F	T	T	T
$\mathbf{F}_{[0,0.2]}\ \mu$	T	T	T	T	T	T		
$\mathbf{G}_{[0,0.7]}\mathbf{F}_{[0,0.2]}\ \mu$	T							

μ

Example STL formulas: Overshoot



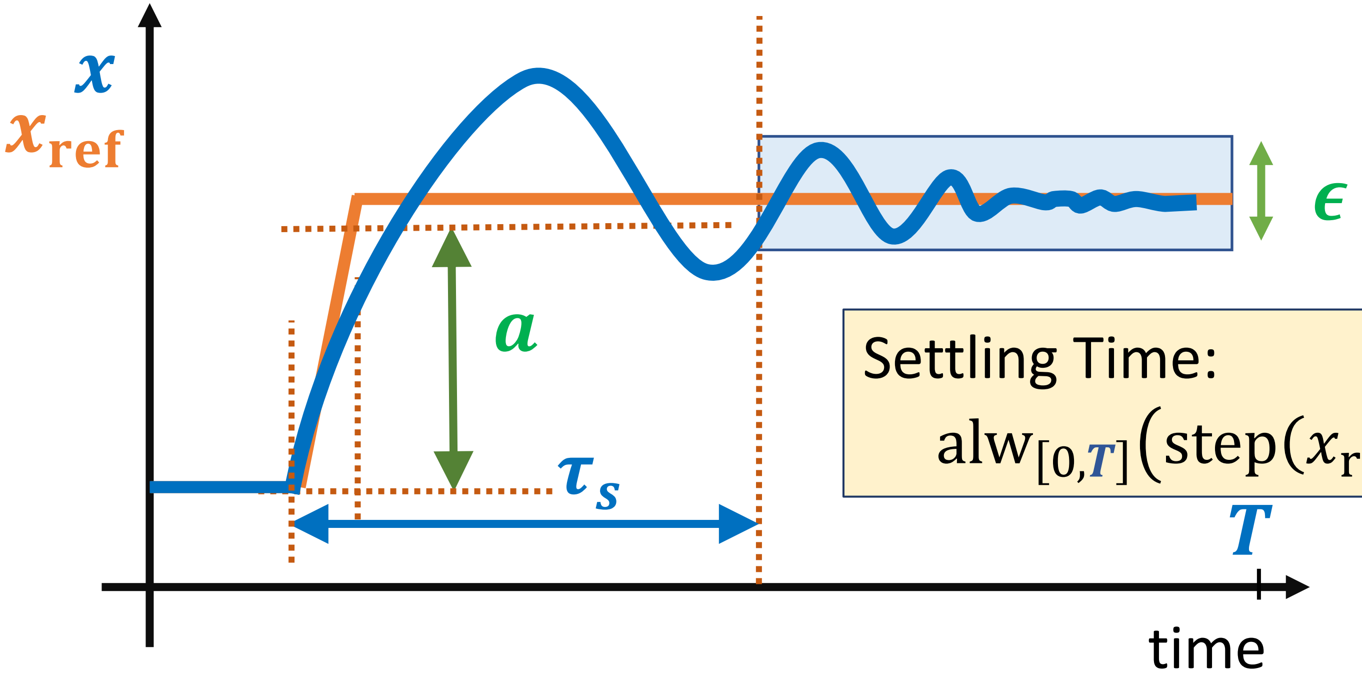
Step:

$$\text{step}(y,t) \coloneqq y(t+\tau) - y(t) > a$$

Overshoot:

$$\text{alw}_{[0,T]}\big(\text{step}(x_{\text{ref}},t) \Rightarrow \text{alw}_{[0,\tau]}(x(t) - x_{\text{ref}}(t) < c)\ \big)$$

Example STL formulas: Settling Time



Step:

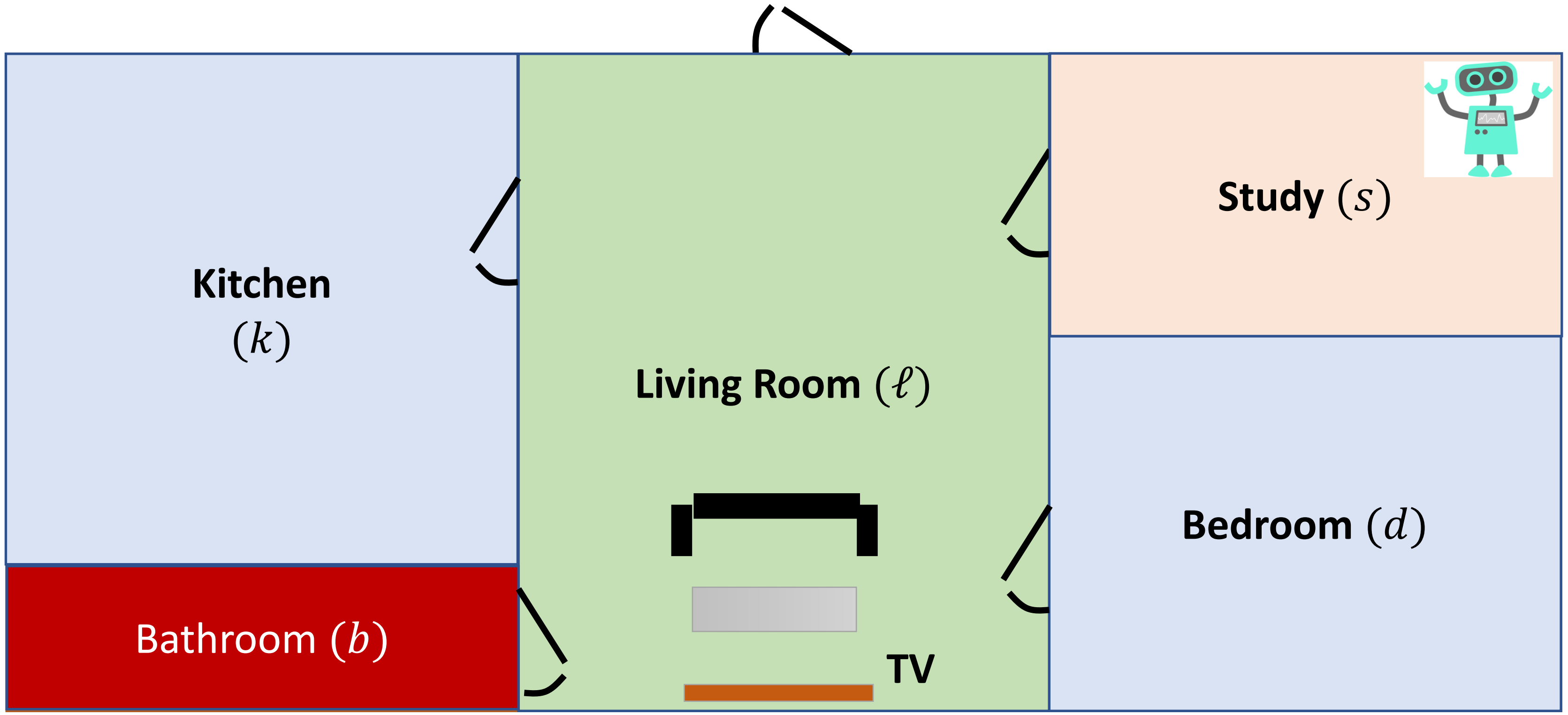
$\mathrm{step}(y, t) := y(t + \delta) - y(t) > a$

Settling Time:

$\mathrm{alw}_{[0,T]}\big(\mathrm{step}(x_{\mathrm{ref}}, t) \Rightarrow \mathrm{alw}_{[\tau_s,\infty]}(|x(t) - x_{\mathrm{ref}}(t)| < \epsilon)\big)$

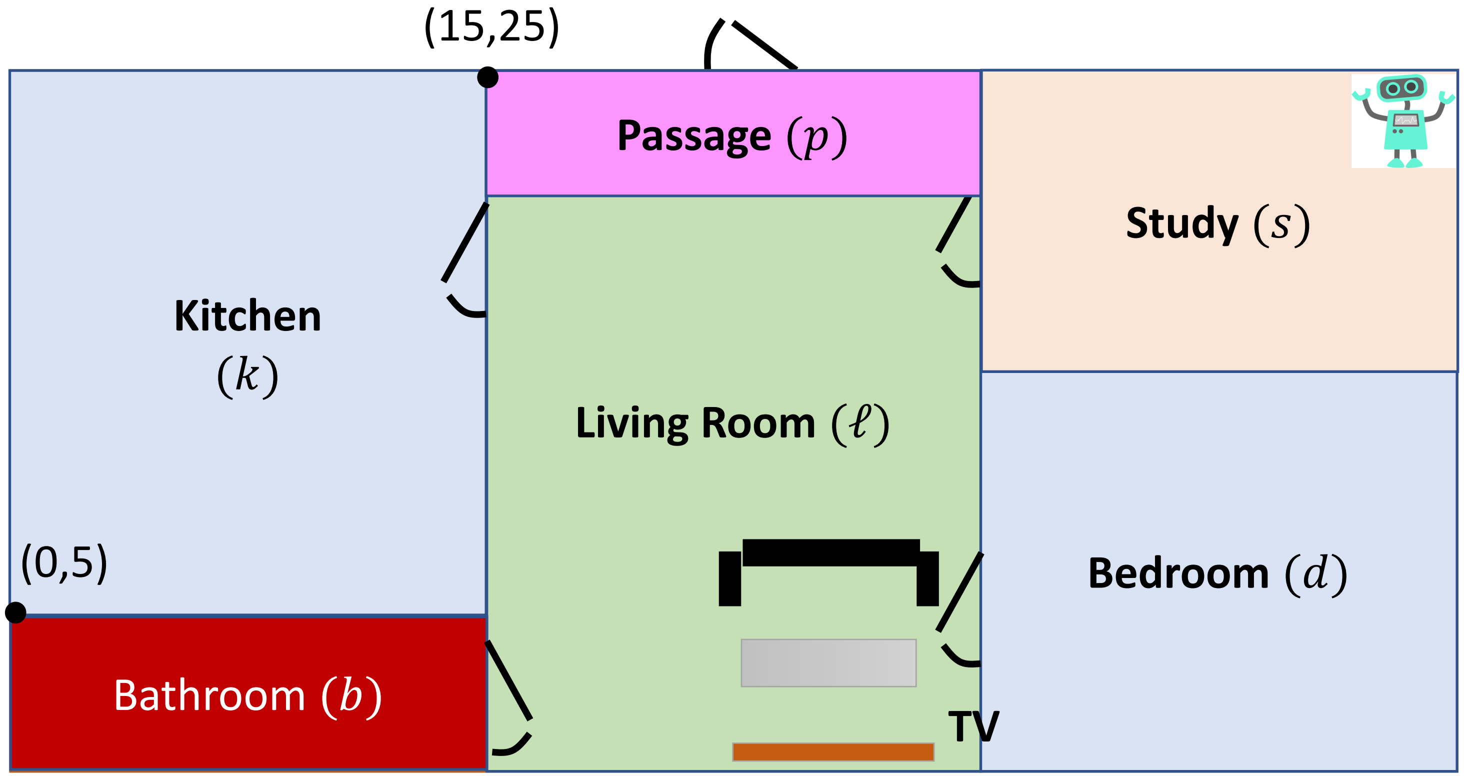
Example specifications in LTL

- Suppose you are designing a robot that has to do a number of missions



- Whenever the robot visits the kitchen, it should visit the bedroom after.
  $$\mathbf{G}(k_r \Rightarrow \mathbf{F}\ d_r)$$
- Robot should never go to the bathroom.
  $$\mathbf{G}\neg b_r$$
- The robot should keep working until its battery becomes low
  $$working\ \mathbf{U}\ low\_battery$$

Robot Path Specification



$$p(t) \in B_k \ : (0 < p_x(t) < 15) \wedge \left(5 \ < p_y(t) < 25\right)$$

- Whenever the robot visits the kitchen, it should visit the bedroom within the next 15 mins.

$$\mathbf{G}\Big(\big(p(t) \in B_k\big) \Rightarrow \mathbf{F}_{[0,15]}\big(p(t) \in B_b\big)\Big)$$

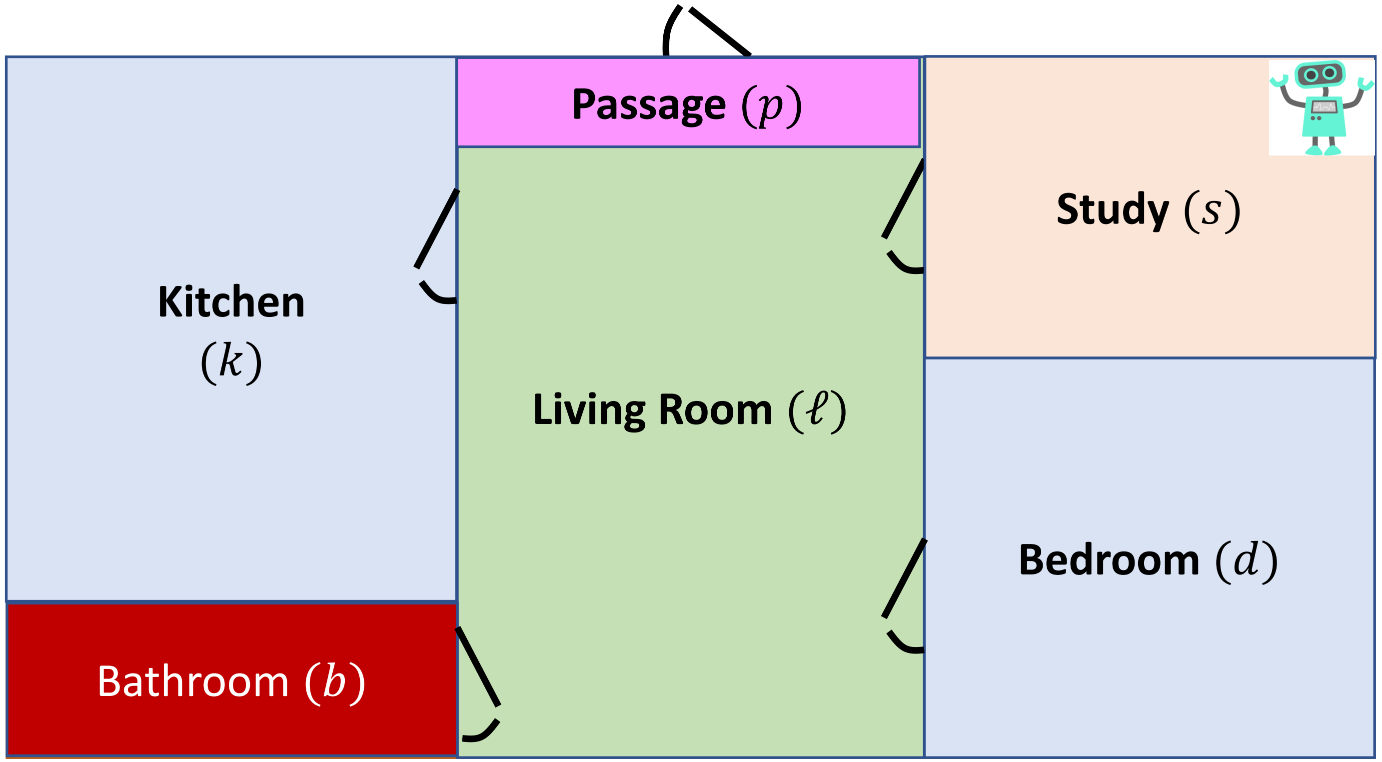
B_r: Box describing room r

$p(t)$: Position of robot at time t

- Robot should not go to the bathroom in the first 60 mins.

$$\mathbf{G}_{[0,60]}\big(p(t) \notin B_{bath}\big)$$

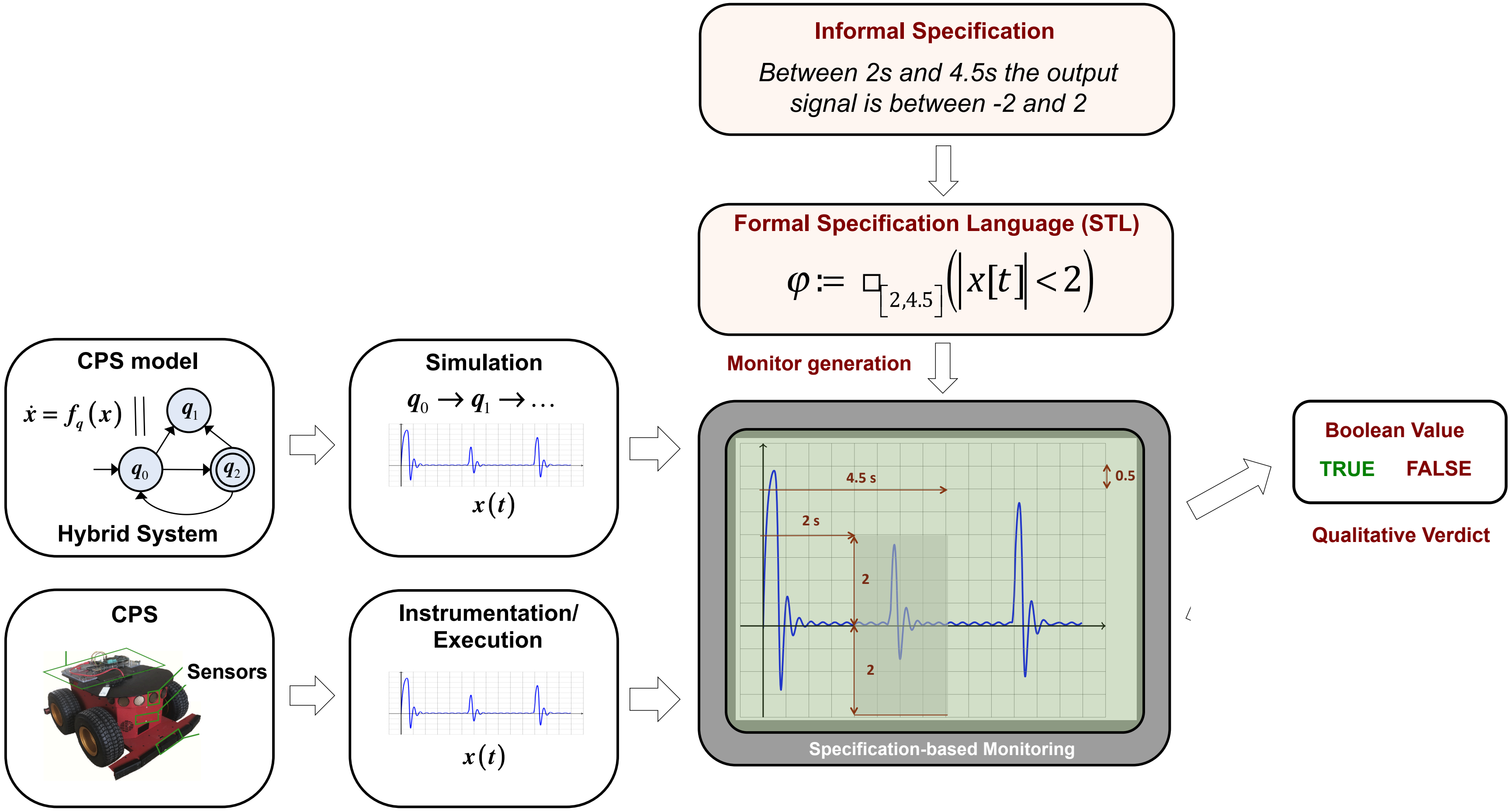
Robot Path Specification



- The robot battery should last between 4 hours and 6 hours
  $(\mathrm{Q(t)} \geq Q_{low})\ \mathbf{U}_{[240,360]}(\mathrm{Q(t)} < \mathrm{Q}_{\mathrm{low}})$
- For the first 10 hours, the robot is never in any room for more than 30 minutes

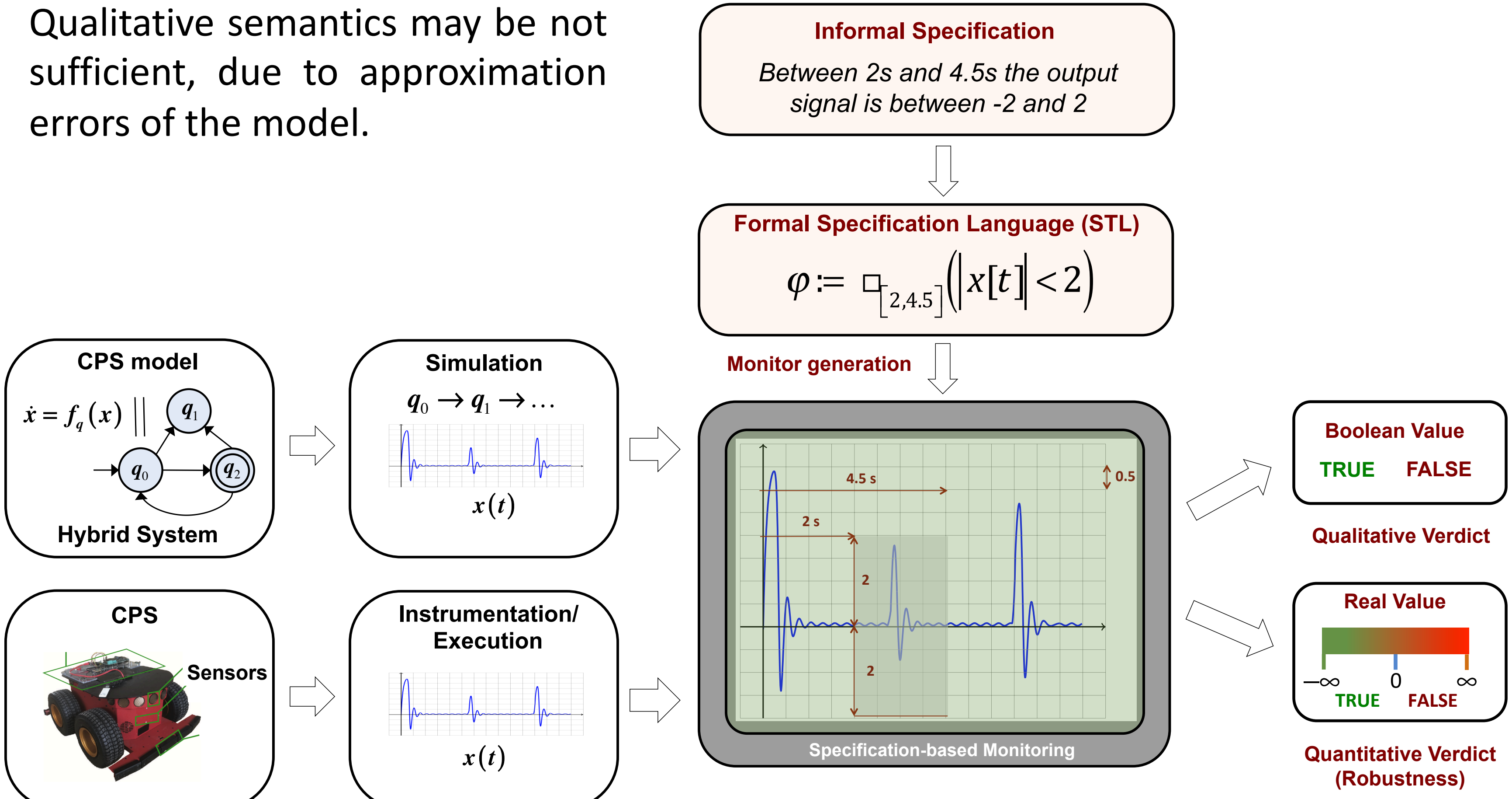
$$\mathbf{G}_{[0,600]}\left(\bigwedge_r \Big((p(t) \in B_r) \Rightarrow \mathbf{F}_{[0,30]}(p(t) \notin B_r)\Big)\right)$$

Specification-based Monitoring



Specification-based Monitoring

Qualitative semantics may be not sufficient, due to approximation errors of the model.

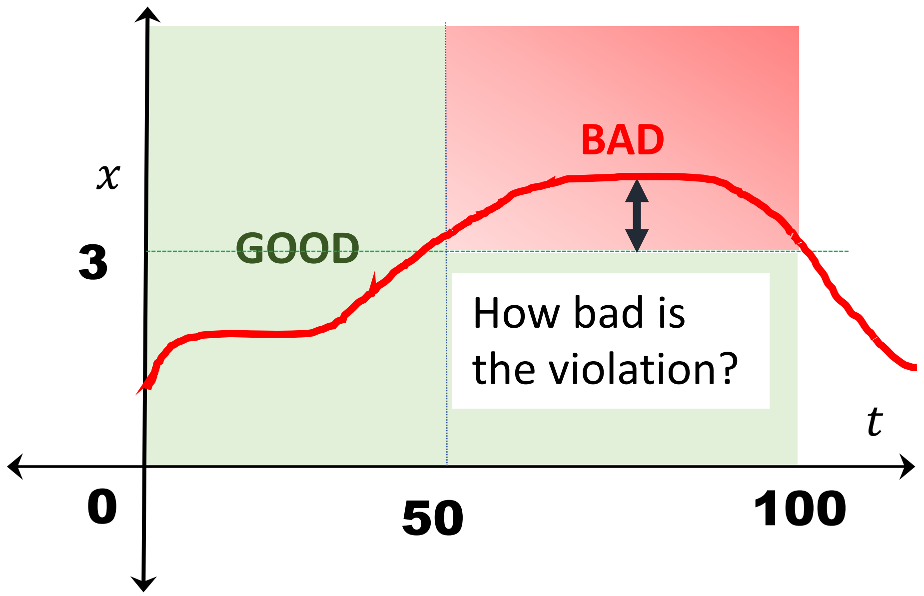


STL has quantitative semantics

- Quantitative semantics defined using the notion of a *Robust Satisfaction Value*, or *Robustness Value*
- Robustness ρ is a function that maps
  - a given trace $\mathbf{x}(t)$,
  - a formula φ,
  - and a time t

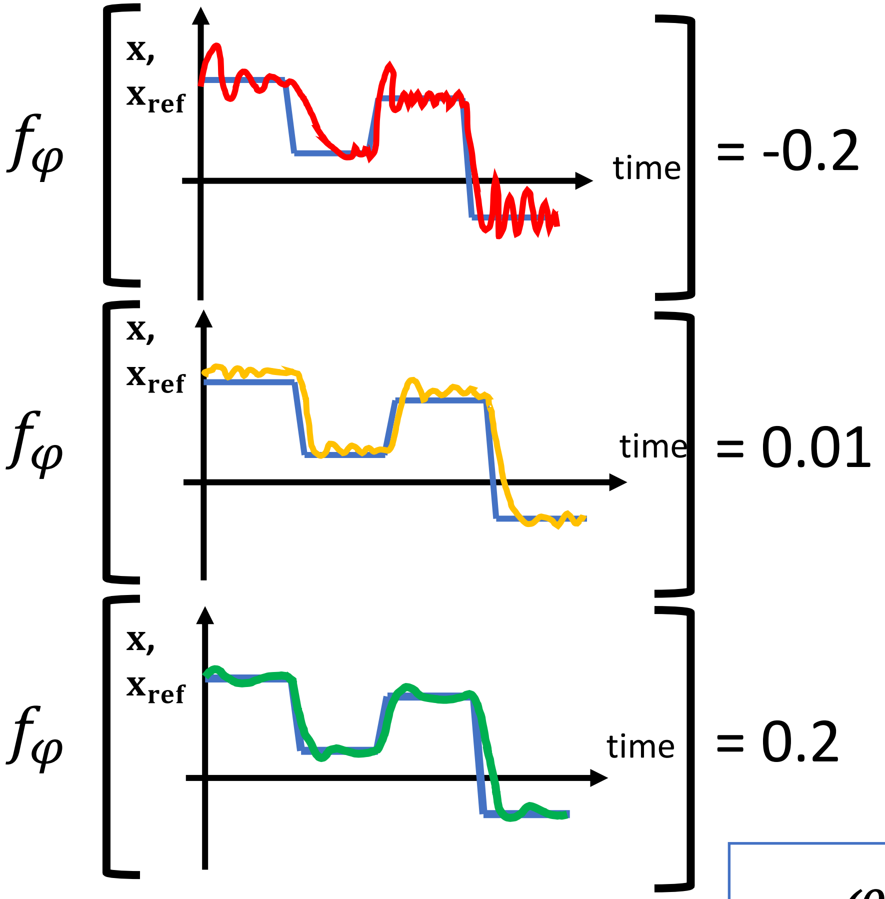
  to some real value
- We can interpret robustness as “distance to violation” of a given formula

Distance to violation/satisfaction



$$\mathbf{G}_{[50,100]}(x(t) < 3)$$

How do quantitative semantics help our engineer?



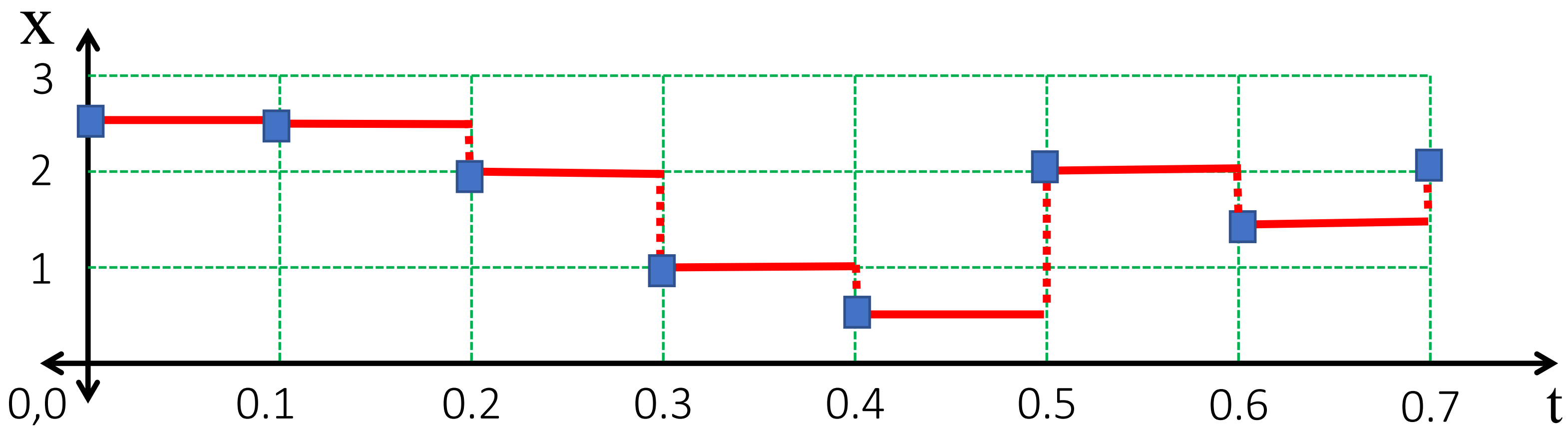
$$\varphi \equiv \text{Alw}_{[0,10]}\left(\text{step} \Rightarrow \text{Alw}_{[0,2]}\left(\left|x - x_{ref}\right| < 0.05\right)\right)$$

Recursive Quantitative Semantics

φ	$\rho(\varphi, \mathbf{x}, t)$
$f(\mathbf{x}) > 0$, $f(\mathbf{x}) \geq 0$	$f(\mathbf{x}(t))$
$\neg\varphi$	$-\rho(\varphi, \mathbf{x}, t)$
$\varphi_1 \wedge \varphi_2$	$\min\big(\rho(\varphi_1, \mathbf{x}, \mathrm{t}) \wedge \rho(\varphi_2, \mathbf{x}, \mathrm{t})\big)$
$\mathbf{F}_{[a,b]}\varphi$	$\sup_{\tau \in [t+a, t+b]} \rho(\varphi, \mathbf{x}, \tau)$
$\mathbf{G}_{[a,b]}\varphi$	$\inf_{\tau \in [t+a, t+b]} \rho(\varphi, \mathbf{x}, \tau)$
$\varphi\ \mathbf{U}_{[a,b]}\ \psi$	$\sup_{\tau \in [t+a, t+b]} \left(\min\left(\rho(\psi, \mathbf{x}, \tau), \inf_{\tau' \in [t, \tau)} \rho(\varphi, \mathbf{x}, t)\right)\right)$

Robustness computation example

$\boldsymbol{\varphi \equiv \mathbf{G}_{[0,0.7]}\mathbf{F}_{[0,0.2]}(x(t) > 1.5)}$



$x(t) - 1.5$	**1**	**1**	**0.5**	**-0.5**	**-1**	**0.5**	**0**	**0.5**
$\mathbf{F}_{[0,0.2]}\ \mu$	**1**	**1**	**0.5**	**0.5**	**0.5**	**0.5**		
$\mathbf{G}_{[0,0.7]}\mathbf{F}_{[0,0.2]}\ \mu$	**0.5**							

$f(x(t)) > 0$ at time t	$f(x(t))$
$F_{[a,b]}\ \varphi$ at time t	Maximum over robustness of φ for $t' \in t \oplus [a, b]$
$G_{[a,b]}\varphi$ at time t	Minimum over robustness of φ for $t' \in t \oplus [a, b]$

Property of Robust Satisfaction Signal

- Sign indicates satisfaction status (soundness):

$$\rho(\varphi, \mathbf{x}, t) > 0 \;\Rightarrow\; \beta(\varphi, \mathbf{x}, t) = 1$$
$$\rho(\varphi, \mathbf{x}, t) < 0 \;\Rightarrow\; \beta(\varphi, \mathbf{x}, t) = 0$$

$$\rho(\varphi, x) = \rho(\varphi, x, 0)$$

- Absolute value indicates tolerance (correctness)

$$\left\|\mathbf{x} - \mathbf{x}'\right\|_{\infty} < \rho(\varphi, \mathbf{x}, t) \;\Rightarrow\; \beta(\varphi, \mathbf{x}, t) = \beta(\varphi, \mathbf{x}', t)$$

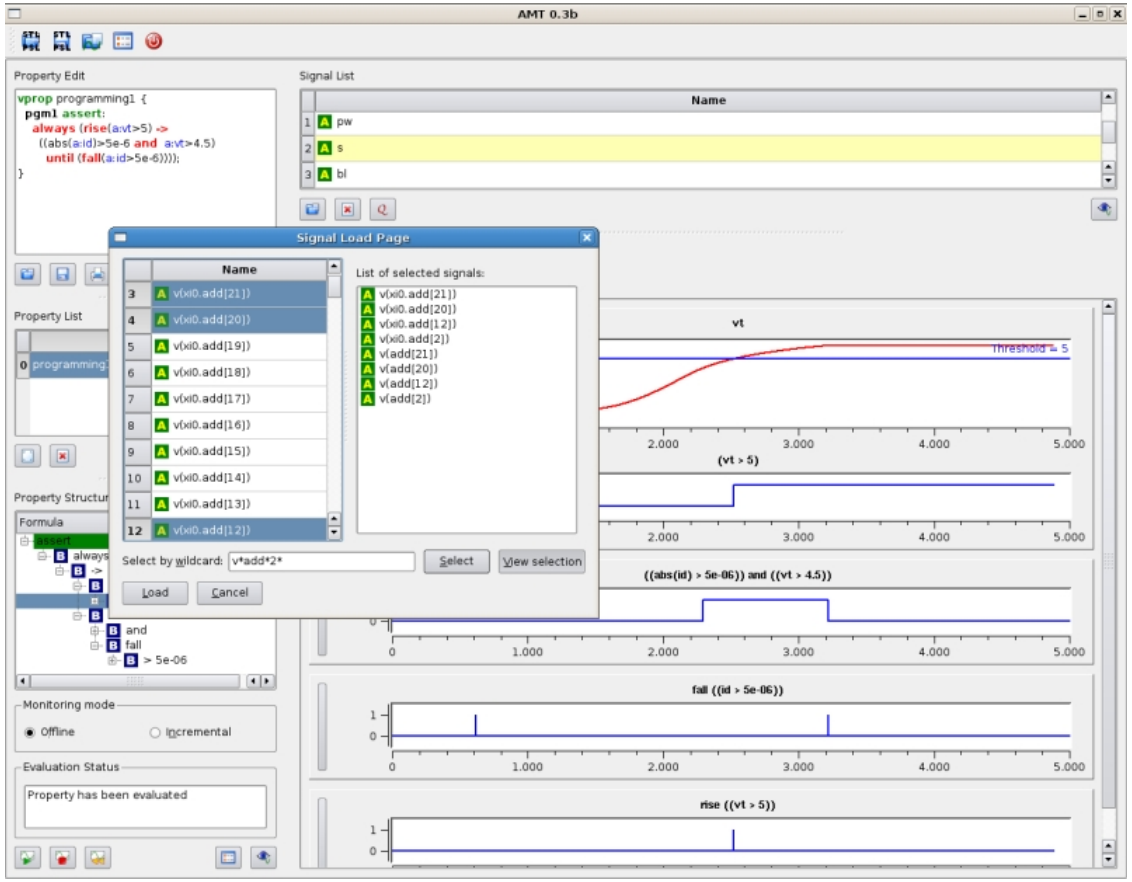
The many uses of STL

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

Analog Monitoring Tool (AMT)

http://www-verimag.imag.fr/DIST-TOOLS/TEMPO/AMT/content.html

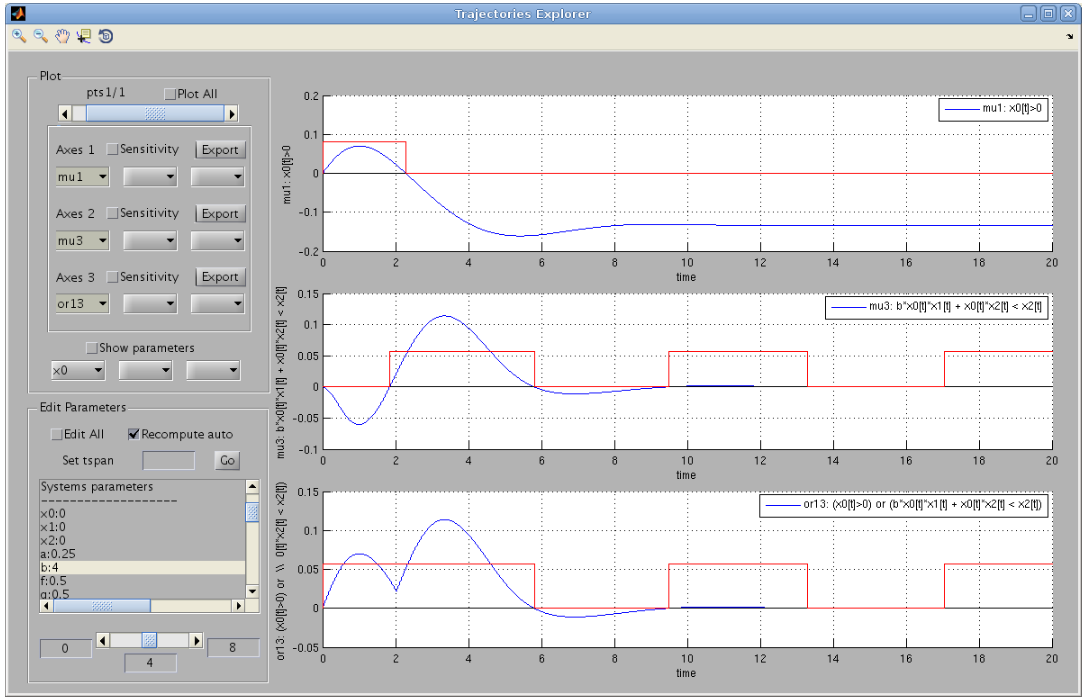
- Java toolbox
- STL with qualitative semantics
  - Correctness
- Offline monitoring



Breach

https://github.com/decyphir/breach

- MATLAB toolbox for
  - Simulation
  - Monitoring of temporal properties
  - Reachability



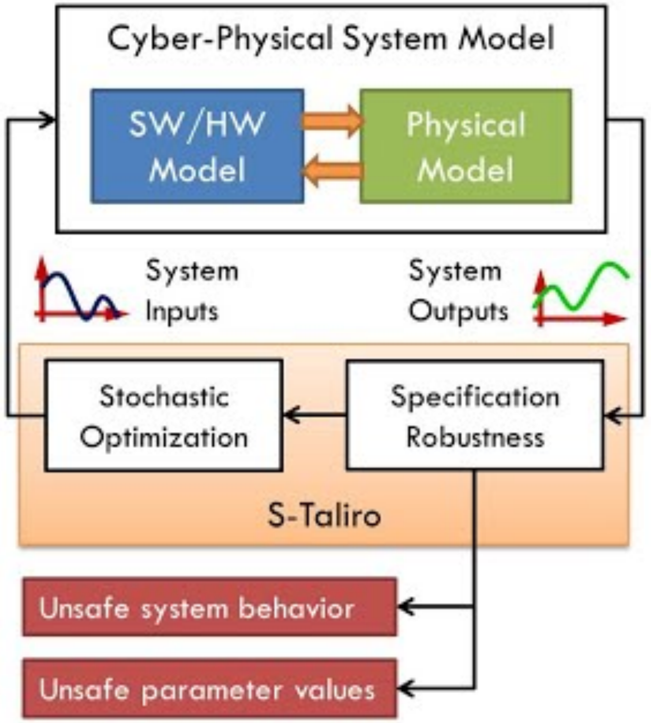
- STL with qualitative and quantitative semantics
  - Correctness
  - Robustness

- Offline and Online monitoring

S-TaLiRo

https://sites.google.com/a/asu.edu/s-taliro/s-taliro

- MATLAB toolbox for searching trajectories with minimal robustness
  - Randomized testing
    - Monte-Carlo simulation
    - Ant-colony optimization
    - Simulated annealing
    - Genetic algorithms
    - Cross enthopy
- MTL with quantitative semantics
  - Robustness
- Offline and Online monitoring



Moonlight

https://github.com/MoonLightSuite/MoonLight

- Java-toolbox + Matlab (and Python) interface for:
  - Monitoring of temporal properties
- STL + spatial operator with qualitative and quantitative semantics
  - Correctness
  - Robustness
- Offline monitoring

Bibliography

1. G. Fainekos, and G. J. Pappas. *Robustness of temporal logic specifications for continuous-time signals*. Theoretical Computer Science 2009.
2. Maler, Oded, and Dejan Nickovic. "Monitoring temporal properties of continuous signals." Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. Springer, Berlin, Heidelberg, 2004. 152-166.
3. Donzé, Alexandre, and Oded Maler. "Robust satisfaction of temporal logic over real-valued signals." International Conference on Formal Modeling and Analysis of Timed Systems. Springer, Berlin, Heidelberg, 2010.