Cyber-Physical Systems

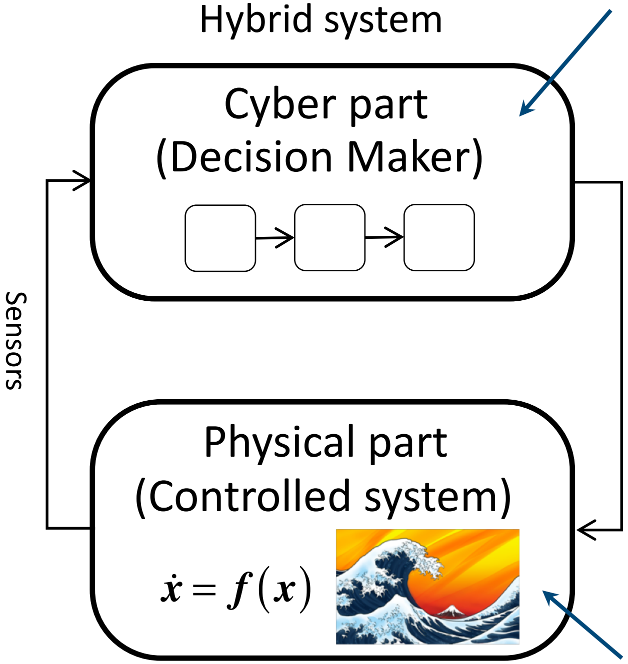
Laura Nenzi

Università degli Studi di Trieste
II Semestre 2018

Lecture 9: Signal Temporal Logic

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

Cyber-Physical Systems (CPS)



Hybrid system

Discrete

Cyber part
(Decision Maker)

Sensors

Actuators

Physical part
(Controlled system)

$\dot{x} = f(x)$

Continuous

Amazon drone

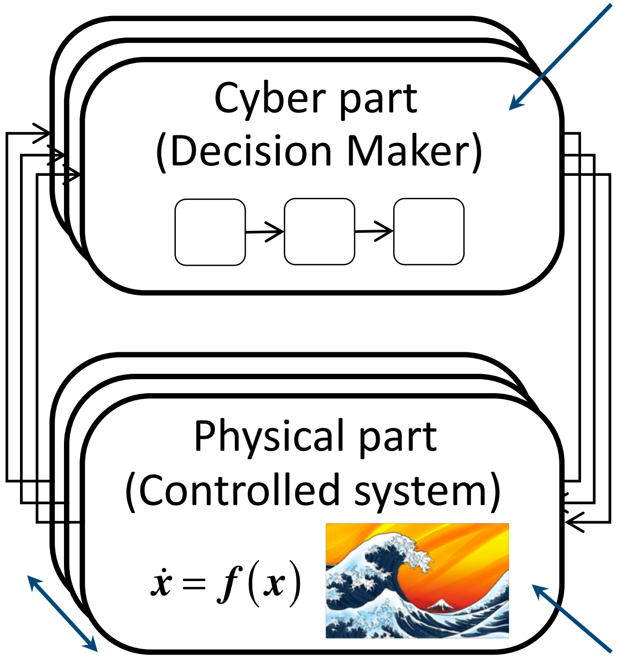
Kiva robots

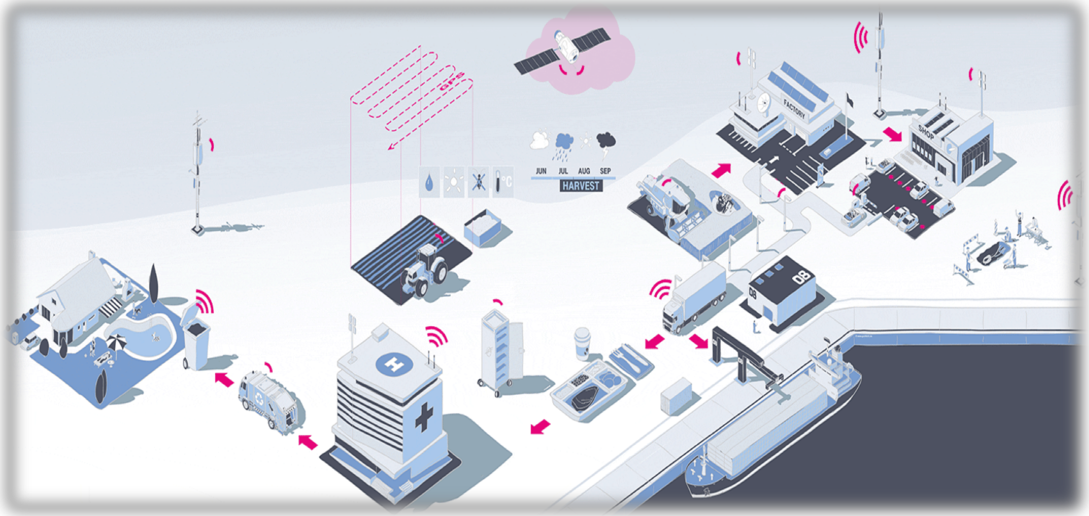
Google self-driving car

Insulin pump

Defibrillator

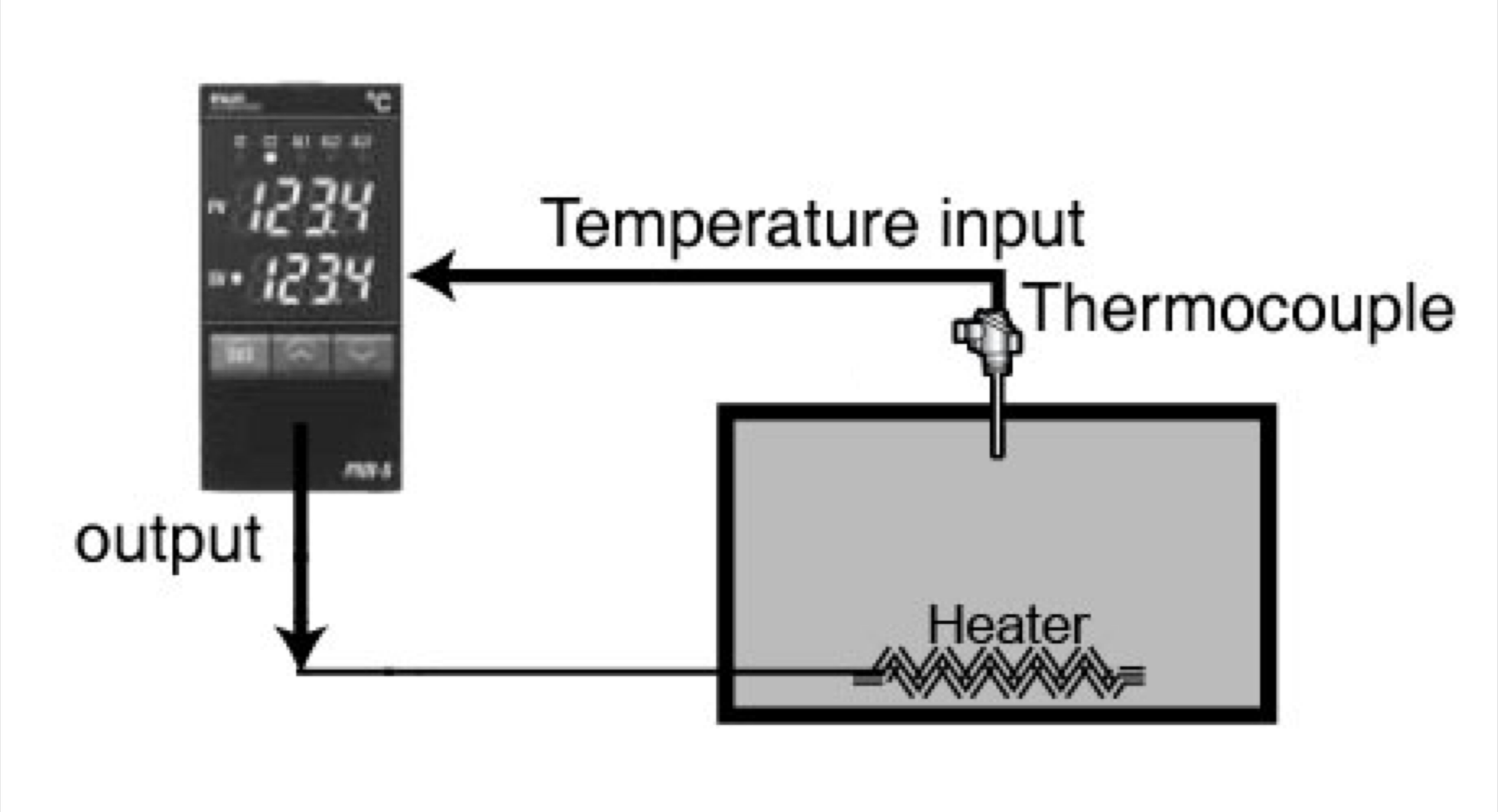
Cyber-Physical Systems (CPS)





Credits for this picture: http://kayarvizhy.com/

Temperature Control

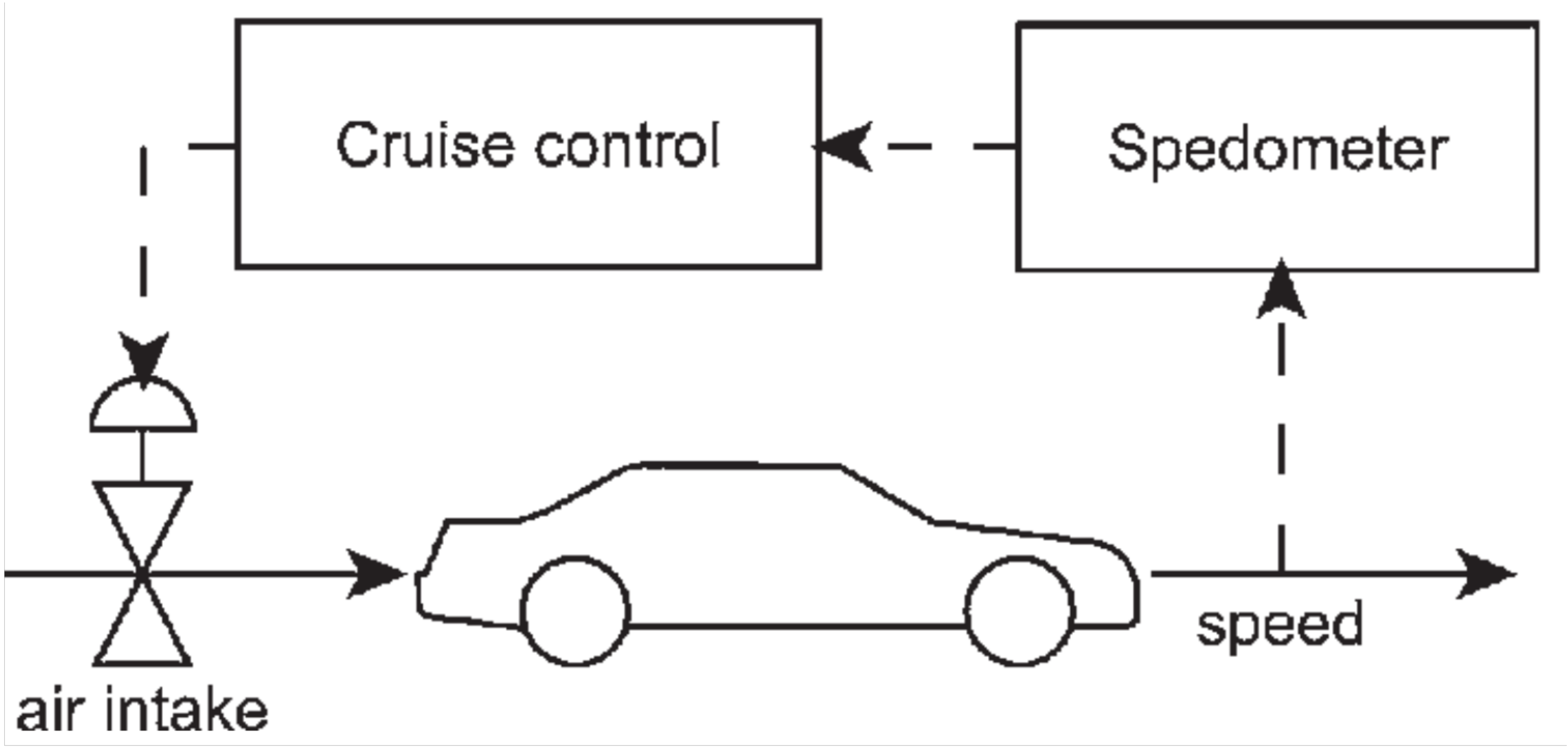


Artificial Pancreas

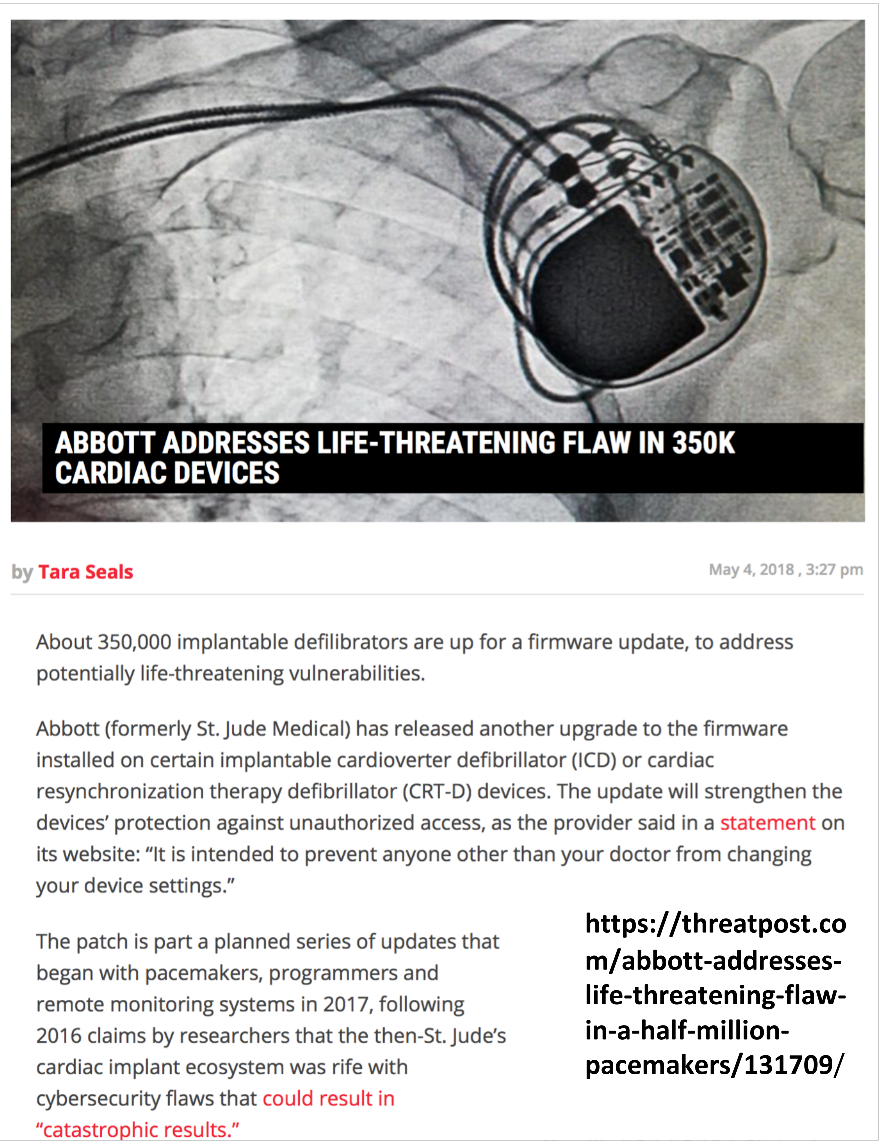
Type 1 diabetes occurs when the pancreas produces little or none of the insulin needed to regulate blood glucose

They rely on external administration of insulin to manage their blood glucose levels.

Automotive Car



Are we safe ?



ABBOTT ADDRESSES LIFE-THREATENING FLAW IN 350K CARDIAC DEVICES

by **Tara Seals**

May 4, 2018 , 3:27 pm

About 350,000 implantable defilibrators are up for a firmware update, to address potentially life-threatening vulnerabilities.

Abbott (formerly St. Jude Medical) has released another upgrade to the firmware installed on certain implantable cardioverter defibrillator (ICD) or cardiac resynchronization therapy defibrillator (CRT-D) devices. The update will strengthen the devices' protection against unauthorized access, as the provider said in a statement on its website: "It is intended to prevent anyone other than your doctor from changing your device settings."

The patch is part a planned series of updates that began with pacemakers, programmers and remote monitoring systems in 2017, following 2016 claims by researchers that the then-St. Jude's cardiac implant ecosystem was rife with cybersecurity flaws that could result in "catastrophic results."

https://threatpost.com/abbott-addresses-life-threatening-flaw-in-a-half-million-pacemakers/131709/

Vehicle safety notices – Prestige models among cars recalled in April

A number of Britain's biggest car makers issued vehicle safety recalls in the last month, covering issues from minor missing pieces of trim to engine and steering failure.

Audi, BMW, Lexus, Porsche and Hyundai were among manufacturers to issue mandatory recalls for their cars.

https://inews.co.uk/essentials/lifestyle/cars/car-news/vehicle-safety-recalls-notices-prestige-cars-recalled-april/

Some tragic accidents

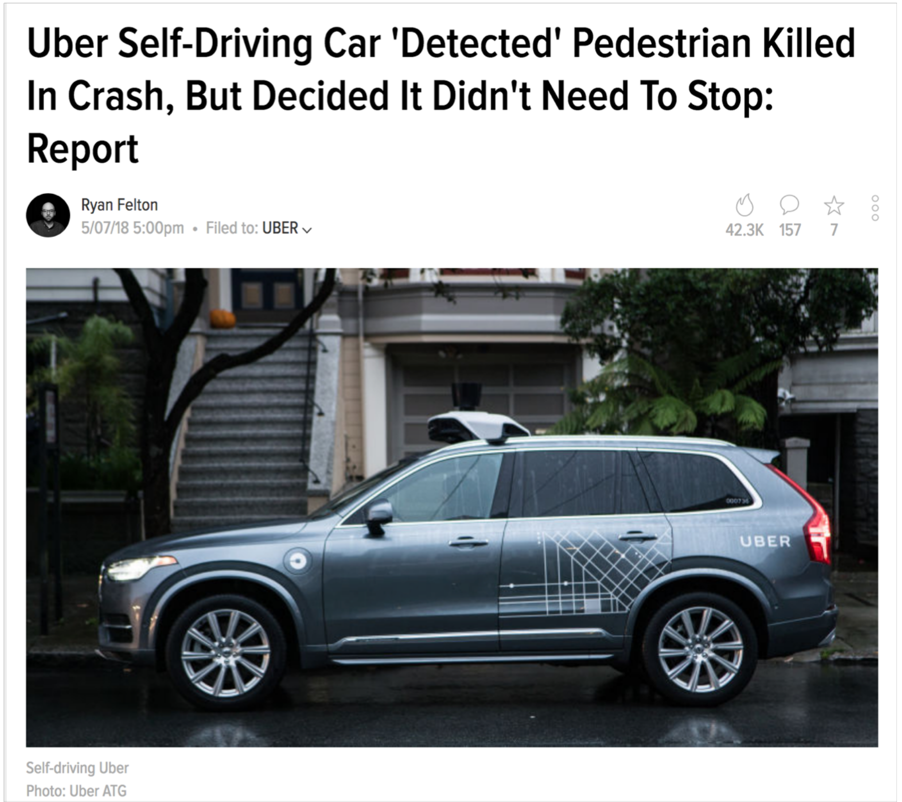
Tesla driver dies in first fatal crash while using autopilot mode

The autopilot sensors on the Model S failed to distinguish a white tractor-trailer crossing the highway against a bright sky

The first known death caused by a self-driving car was disclosed by Tesla Motors on Thursday, a development that is sure to cause consumers to second-guess the trust they put in the booming autonomous vehicle industry.

The 7 May accident occurred in Williston, Florida, after the driver, Joshua Brown, 40, of Ohio put his Model S into Tesla's autopilot mode, which is able to control the car during highway driving.

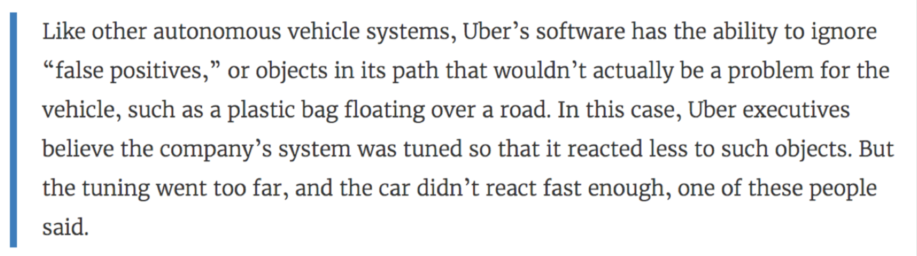
Against a bright spring sky, the car's sensors system failed to distinguish a large white 18-wheel truck and trailer crossing the highway, Tesla said. The car attempted to drive full speed under the trailer, "with the bottom of the trailer impacting the windshield of the Model S", Tesla said in a blogpost.



Uber Self-Driving Car 'Detected' Pedestrian Killed In Crash, But Decided It Didn't Need To Stop: Report

Ryan Felton
5/07/18 5:00pm • Filed to: UBER

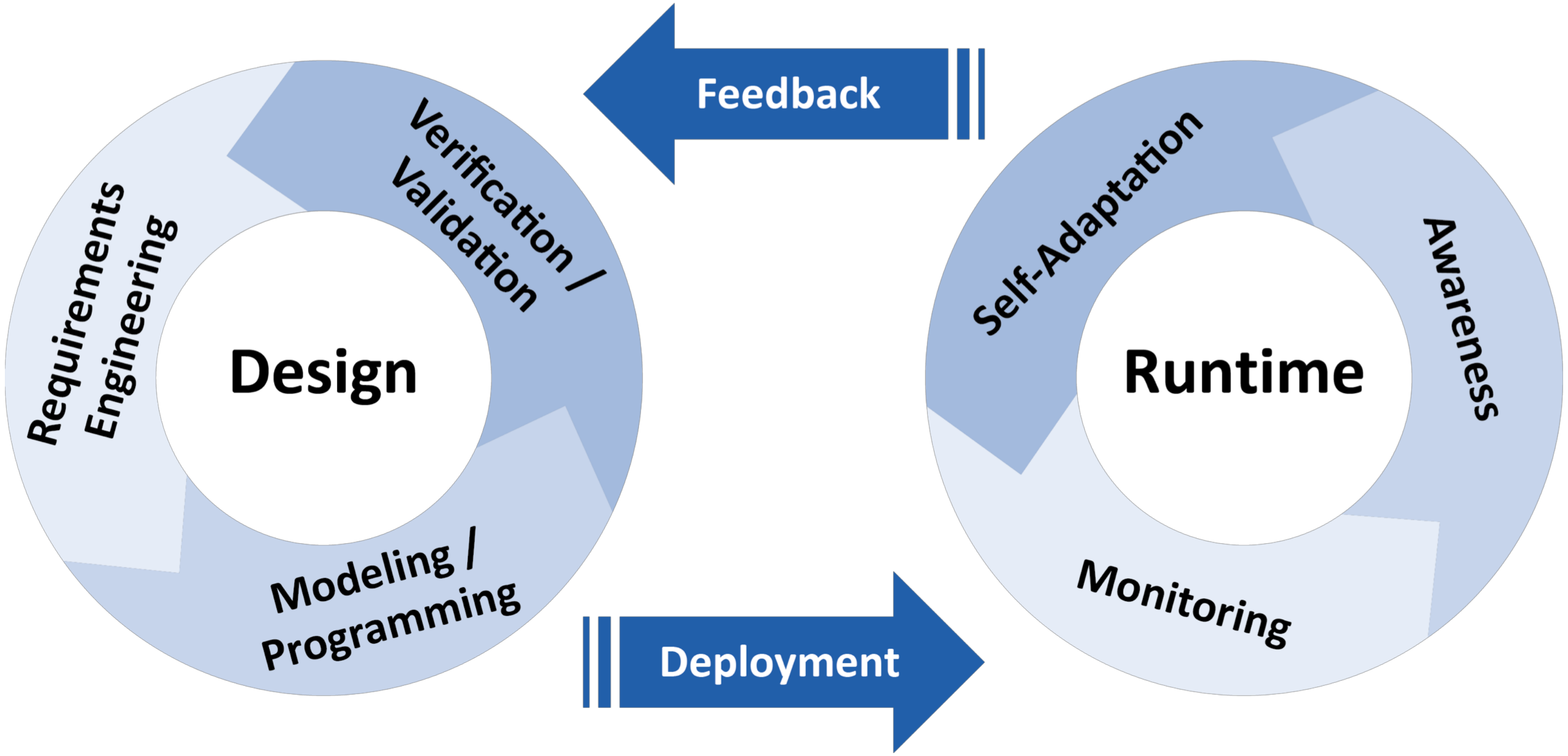
Self-driving Uber
Photo: Uber ATG



> Like other autonomous vehicle systems, Uber's software has the ability to ignore "false positives," or objects in its path that wouldn't actually be a problem for the vehicle, such as a plastic bag floating over a road. In this case, Uber executives believe the company's system was tuned so that it reacted less to such objects. But the tuning went too far, and the car didn't react fast enough, one of these people said.

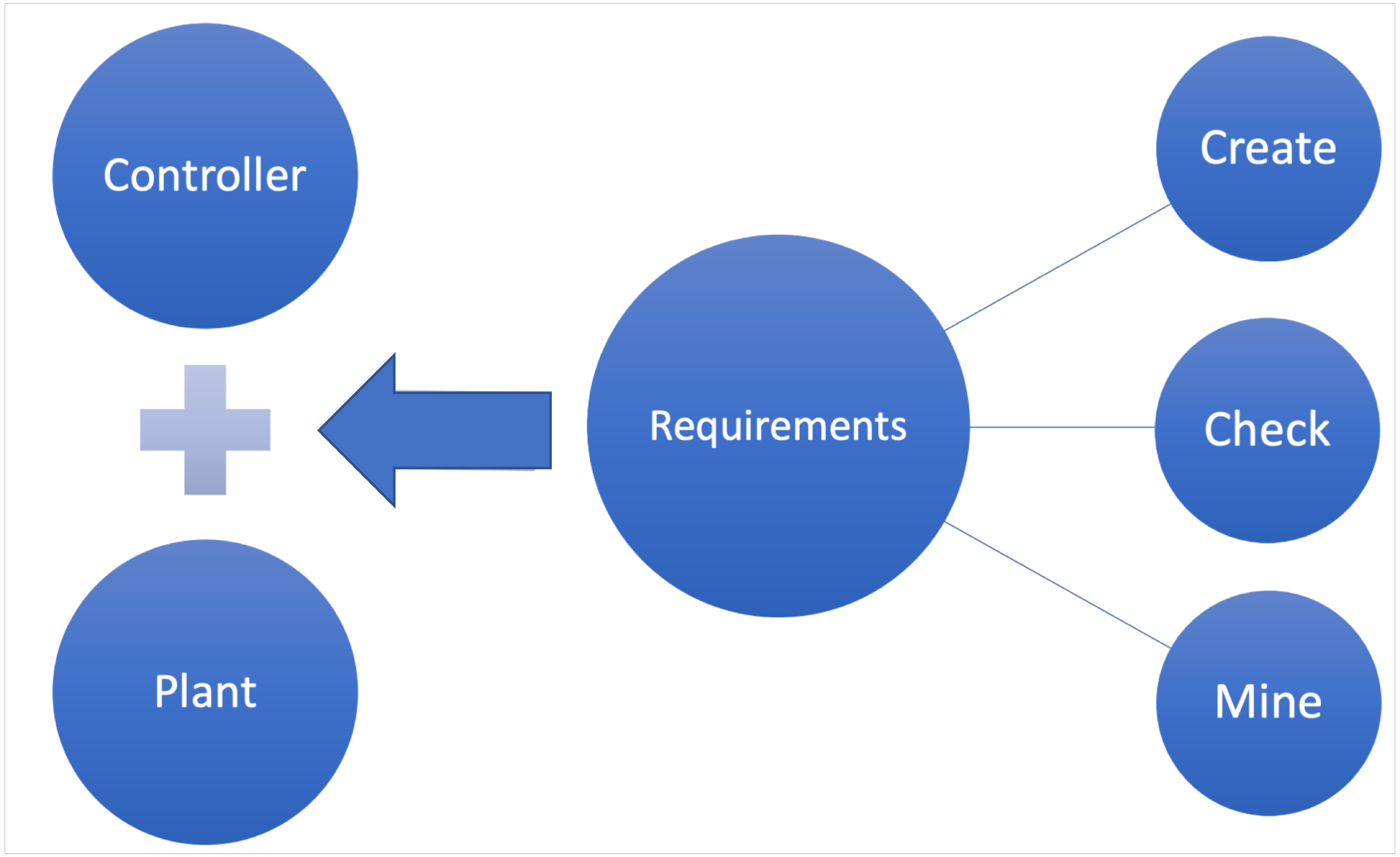
https://jalopnik.com/uber-self-driving-car-detected-pedestrian-killed-in-cra-1825834016

Model-based Design Approach

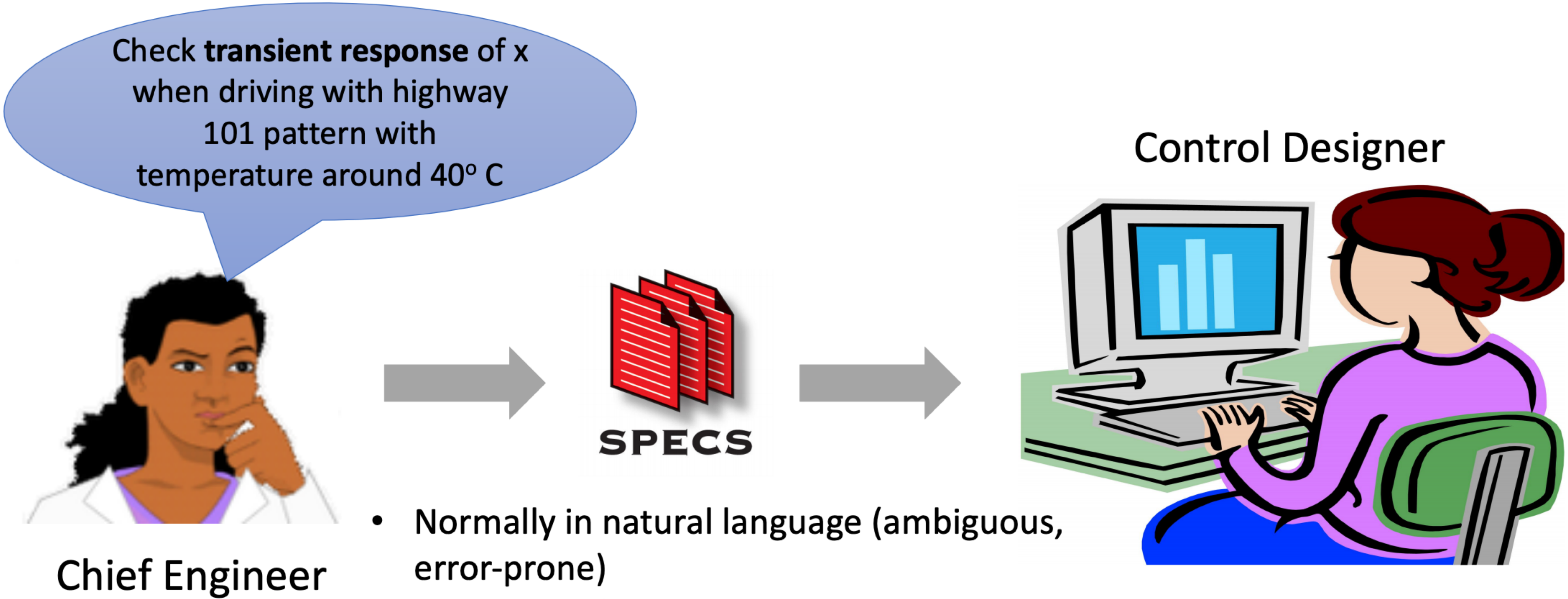


Requirements Driving Design

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

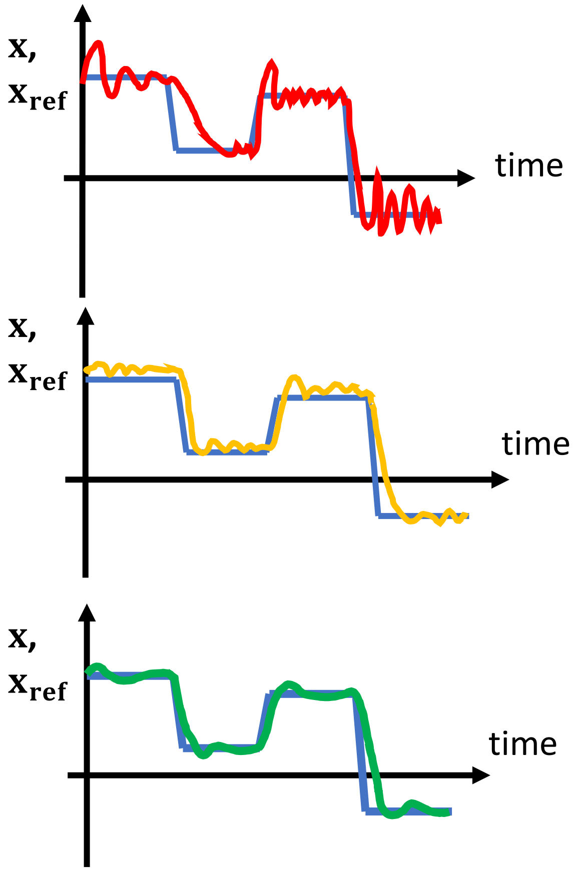


Typical day in a control designer's life



- 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




Uh Oh!

… should be okay

Looks good

Linear Temporal Logic (LTL)

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

Key ingredients:

t Typ

- Propositions
  E.g. p = T<75, q=T>60
- Boolean operators: ∧, ∨, ¬
  E.g. p ∧ q
- Temporal Operators: always (G or □), eventually (F or ◇), until (U), next (X or ○)
  E.g. **G**(p ∧ q)

Linear Temporal Logic (LTL)

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

$X\,a \wedge X\,X\,a \wedge X\,X\,X\,a$ with $a = T<75 \wedge T>60$

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)

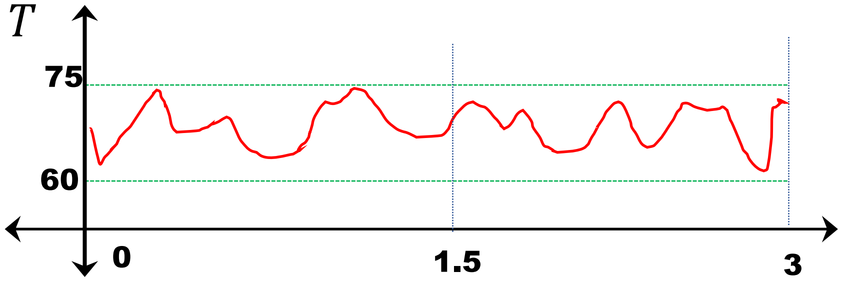
Invented by D. Nickovic and O. Maler from Verimag (2004)
It extended MITL by having **signal predicates over real values as atomic formulas:**

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

Expressing specifications in STL

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

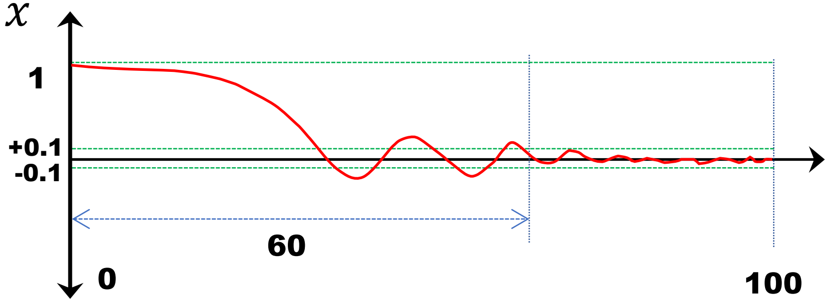
Always between time 0 and 3



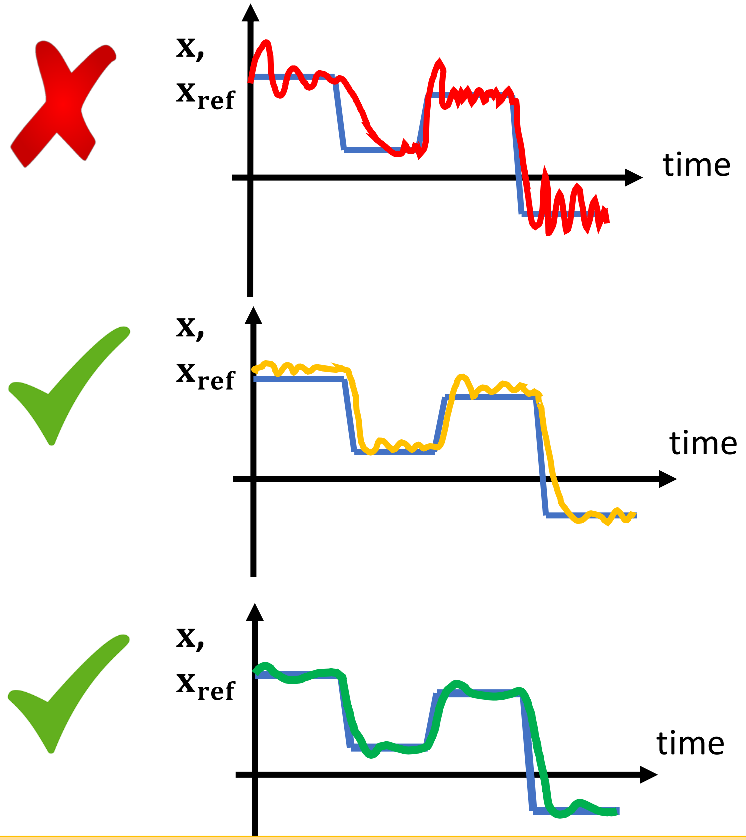
$\mathbf{Eventually}_{[0,60]}(\mathbf{Always}\ (|x(t)| < 0.1)\ )$

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

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

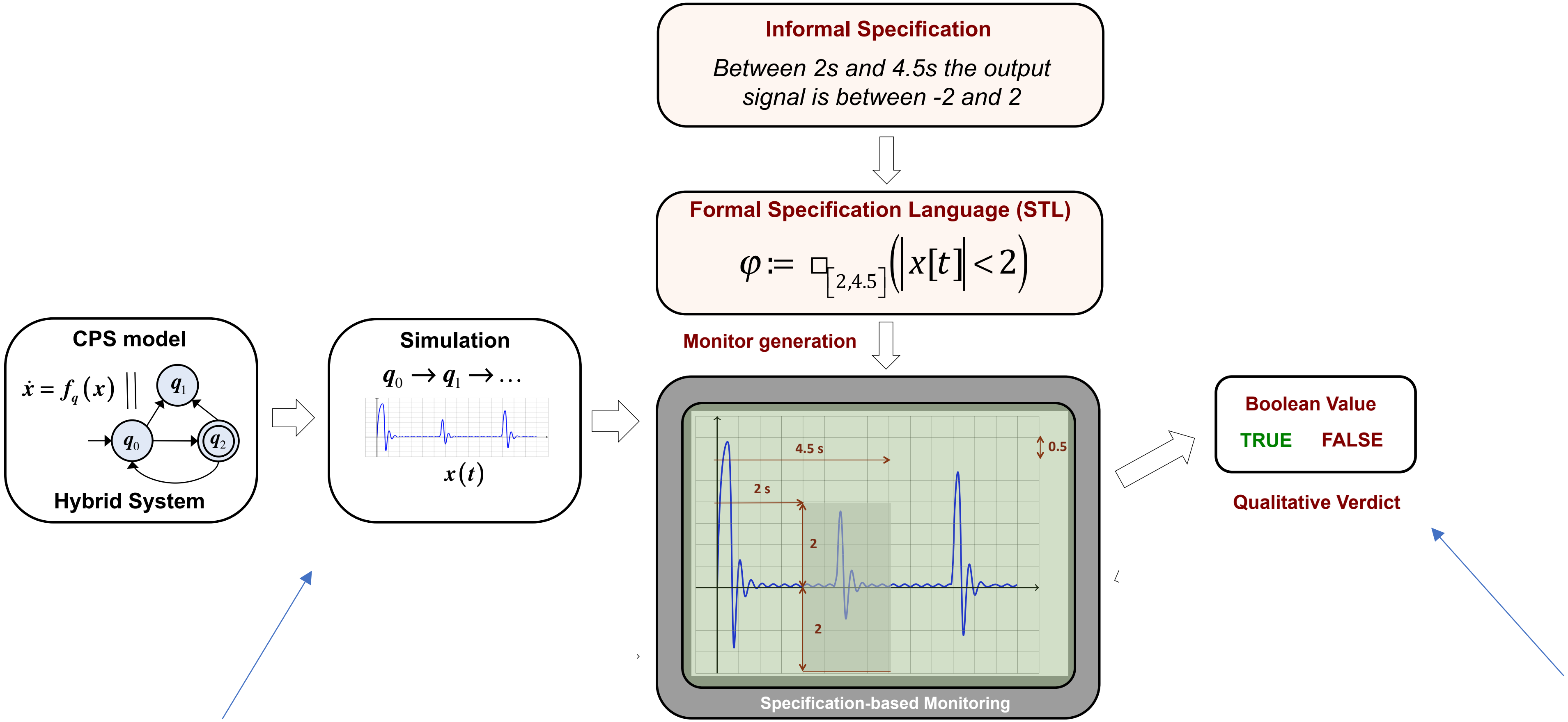


Can we express our engineer’s requirements?



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

Specification-based Monitoring



Informal Specification

Between 2s and 4.5s the output signal is between -2 and 2

Formal Specification Language (STL)

$$\varphi := \square_{[2,4.5]}\left(\left|x[t]\right| < 2\right)$$

Monitor generation

CPS model

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

Hybrid System

Simulation

$q_0 \rightarrow q_1 \rightarrow \ldots$

$x(t)$

Specification-based Monitoring

Boolean Value

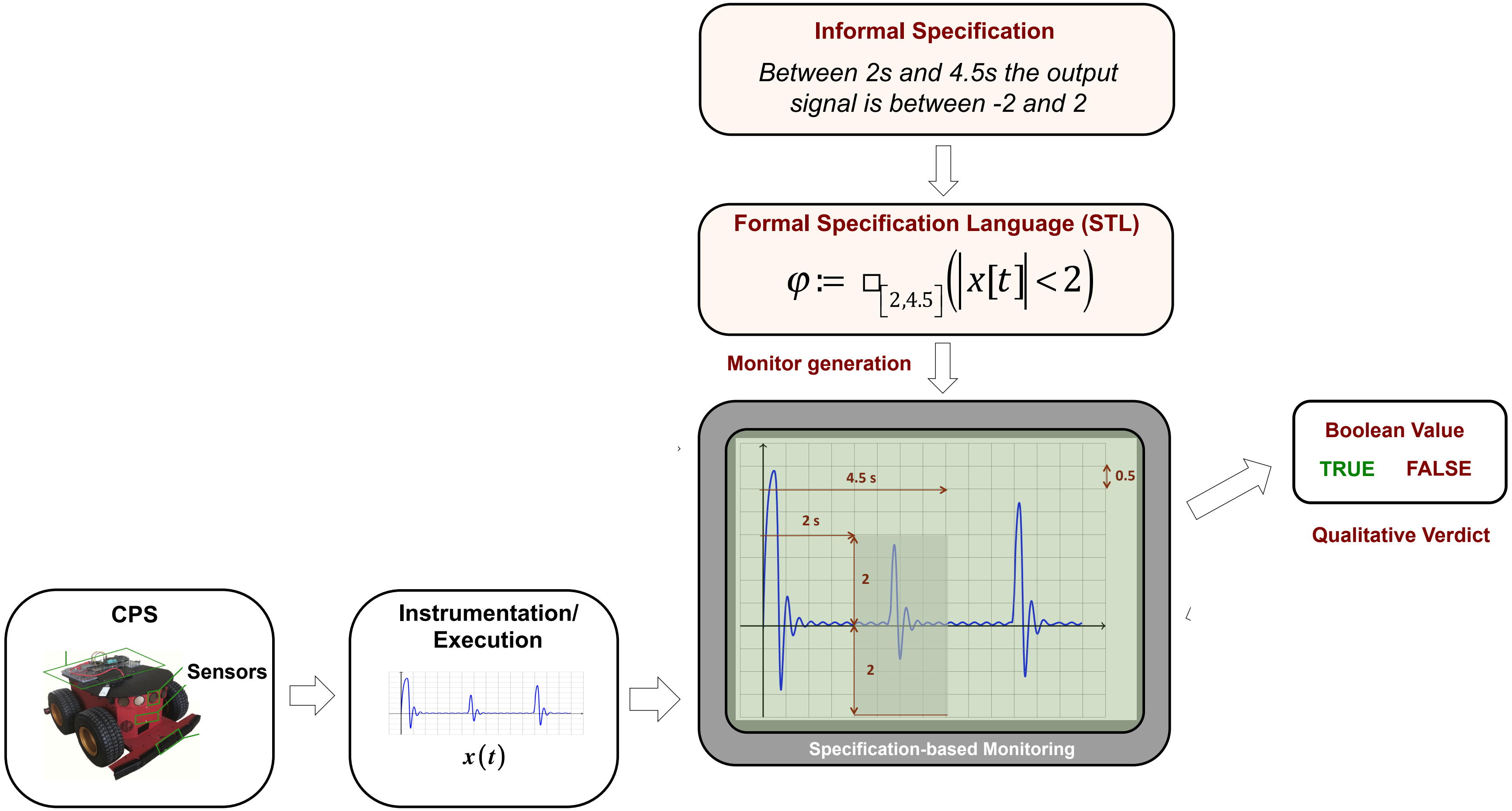
TRUE **FALSE**

Qualitative Verdict

Complex behaviours

Low dimensional vectors

Specification-based Monitoring



Informal Specification

Between 2s and 4.5s the output signal is between -2 and 2

Formal Specification Language (STL)

$$\varphi := \square_{[2,4.5]}\left(\left|x[t]\right| < 2\right)$$

Monitor generation

CPS

Sensors

Instrumentation/ Execution

$x(t)$

Specification-based Monitoring

Boolean Value

TRUE **FALSE**

Qualitative Verdict

STL Syntax

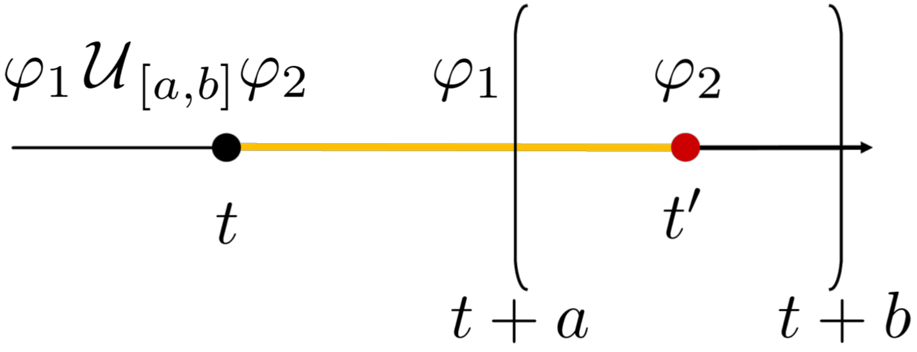
Syntax of STL			
$\varphi \ ::=$	$f(\mathbf{x}) \sim 0$	$\mid$	$f: \mathbb{D} \to \mathbb{R}$ is a function over the signal $\mathbf{x}$: $\mathbb{T} \to \mathbb{D}$, $\sim\, \in \{\leq, <, >, \geq, =, \neq\}$
	$\neg\varphi$	$\mid$	Negation
	$\varphi \wedge \varphi$	$\mid$	Conjunction
	$\mathbf{F}_{[a,b]}\varphi$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

Recursive Boolean Semantics of STL

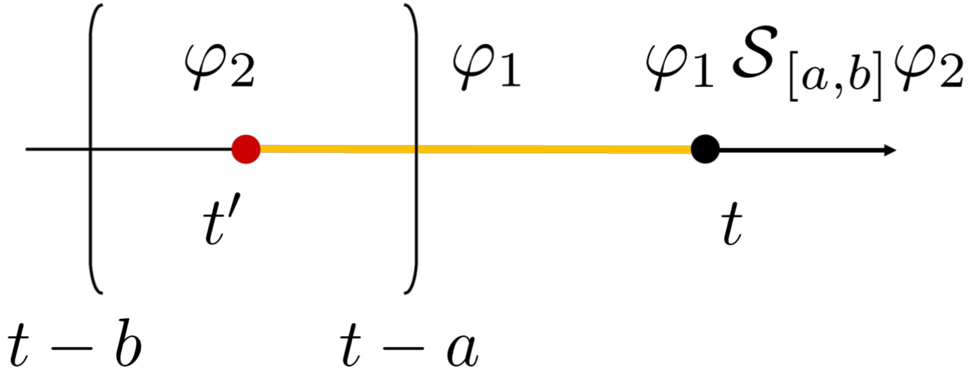
φ	$\beta(\varphi, \mathbf{x}, t)$
$f(\mathbf{x}) \sim 0$	$f(\mathbf{x}(t)) \sim 0,$ $\quad \sim\ \in \{\leq, <, >, \geq, =, \neq\}$
$\neg\varphi$	$\neg\beta(\varphi, \mathbf{x}, t)$
$\varphi_1 \wedge \varphi_2$	$\beta(\varphi_1, \mathbf{x}, \mathrm{t}) \wedge \beta(\varphi_2, \mathbf{x}, \mathrm{t})$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

Since and Until Operators

- Until



- Since

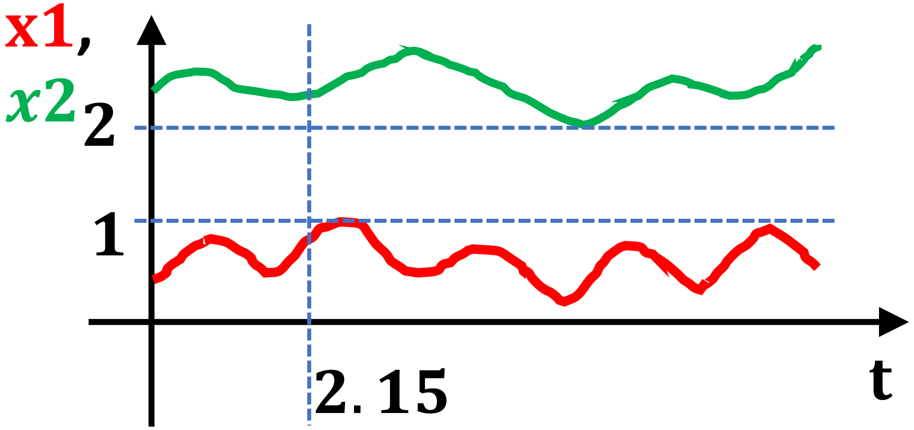


STL semantics

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

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

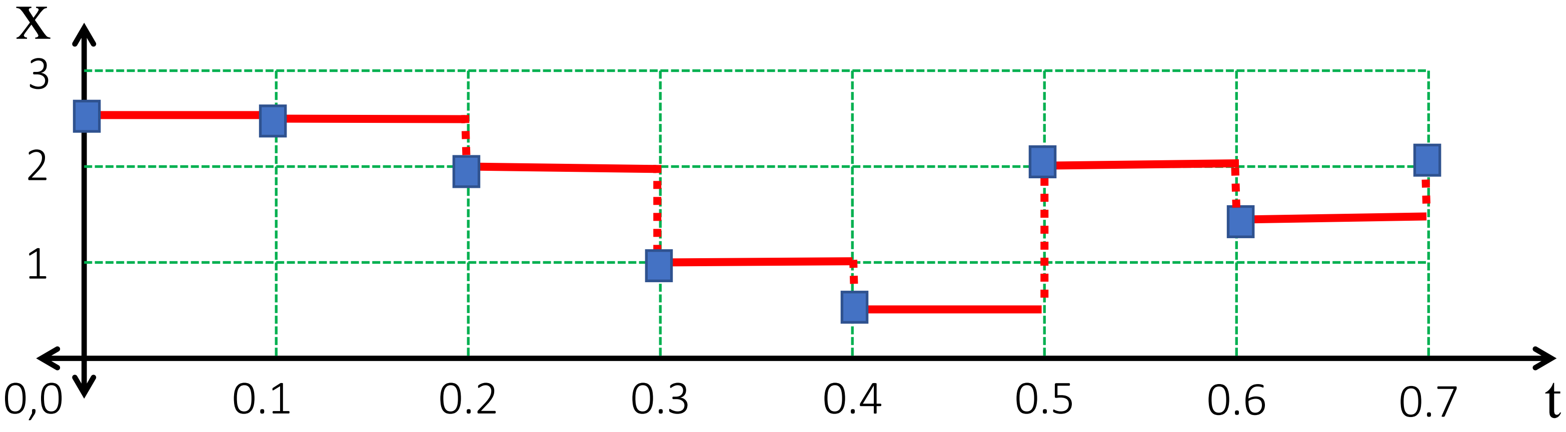
- If φ is the signal predicate $\mu = f(\mathbf{x}) > 0$, then $\beta(\varphi, \mathbf{x}, t) = \mathit{true}$ iff $f\big(\mathbf{x}(t)\big) > 0$

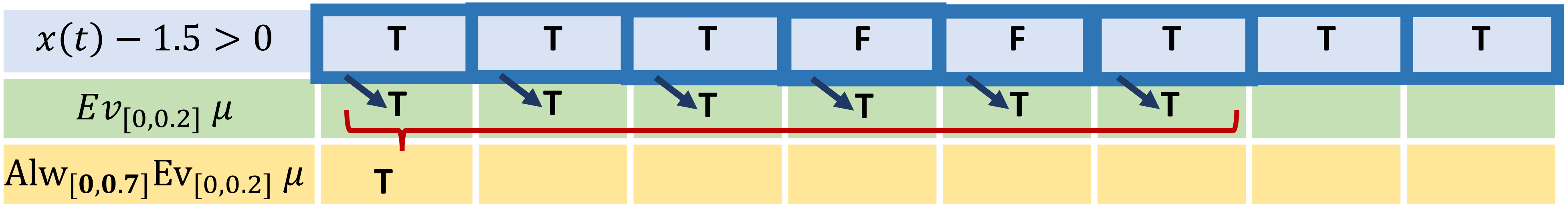


$$\mathbf{x} = (x1, x2)$$
$$\boldsymbol{f} = x2 - x1 - 1$$
$$\beta(f(\mathbf{x}) > 0, \mathbf{x}, 2.15)?$$

Recursive Boolean Semantics of STL

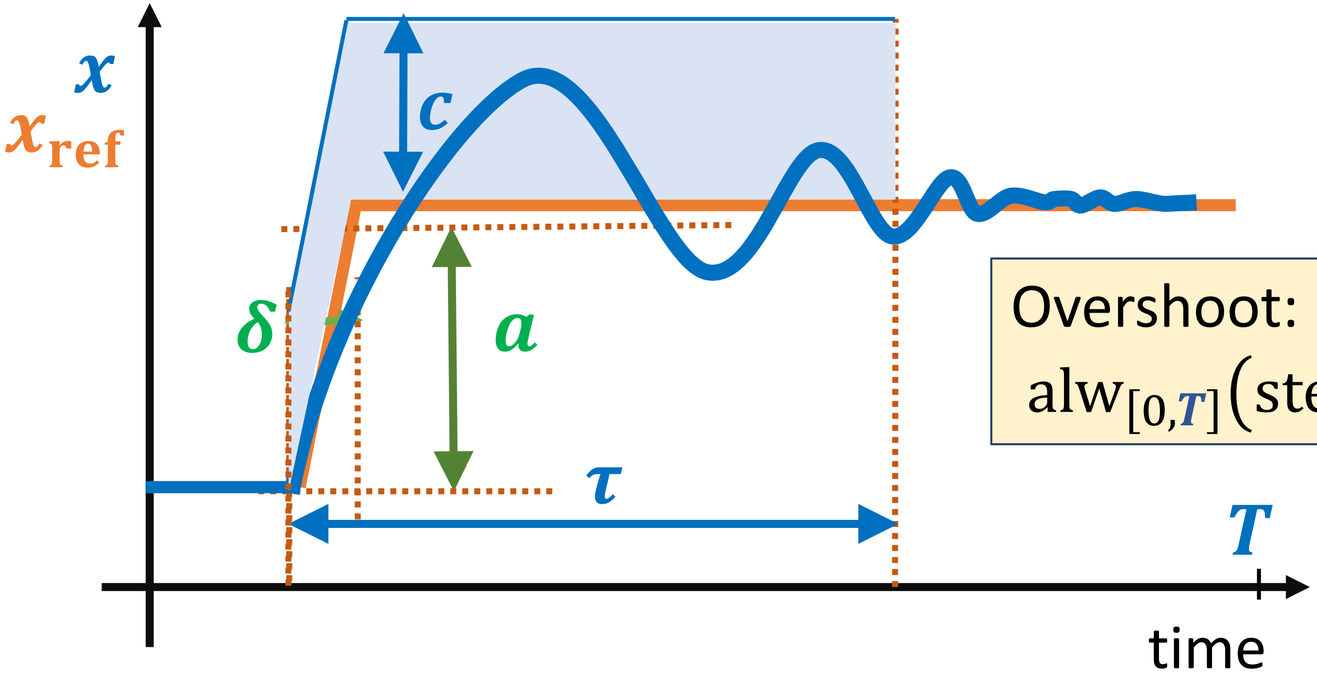
$\boldsymbol{\varphi} \equiv$
$\mathbf{G}_{[0,0.7]}\mathbf{F}_{[0,0.2]}(\boldsymbol{x}(\boldsymbol{t}) \geq \mathbf{1.5})$





$x(t) - 1.5 > 0$	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**
$Ev_{[0,0.2]}\ \mu$	**T**	**T**	**T**	**T**	**T**	**T**		
$\mathrm{Alw}_{\mathbf{[0,0.7]}}\mathrm{Ev}_{[0,0.2]}\ \mu$	**T**							

Example STL formulas: Overshoot



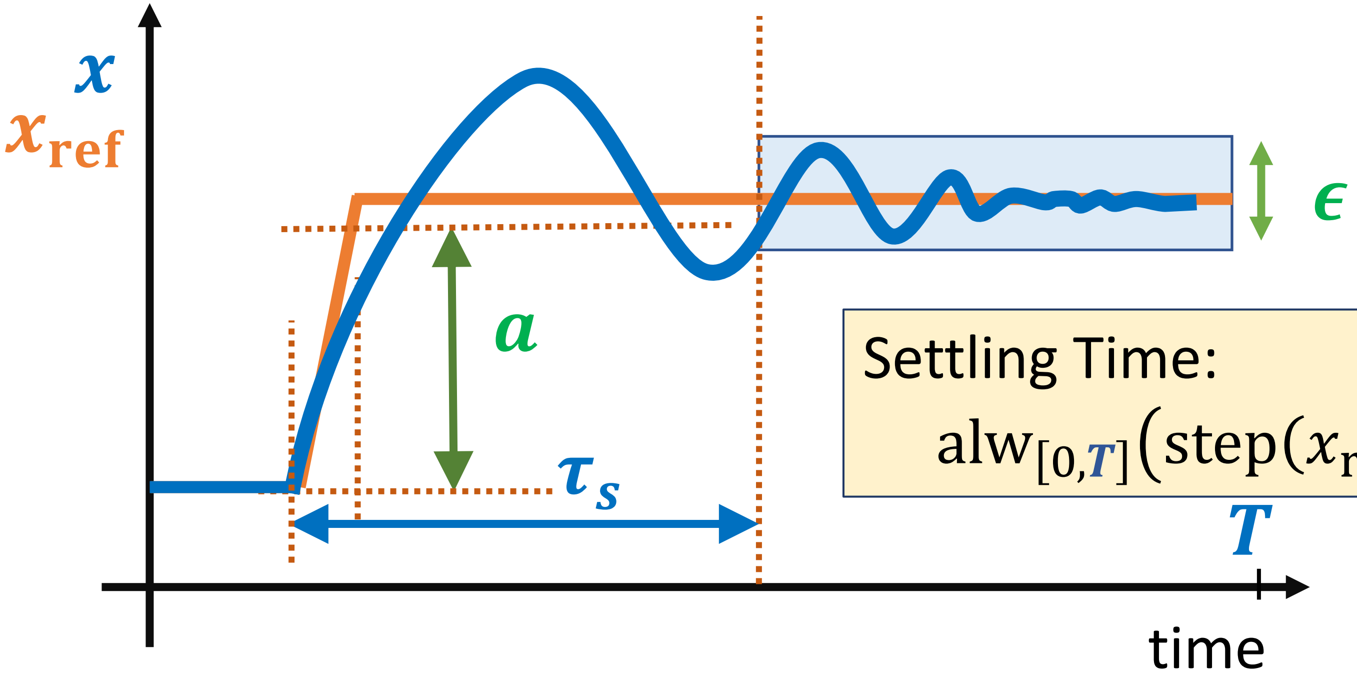
Step:

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

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:

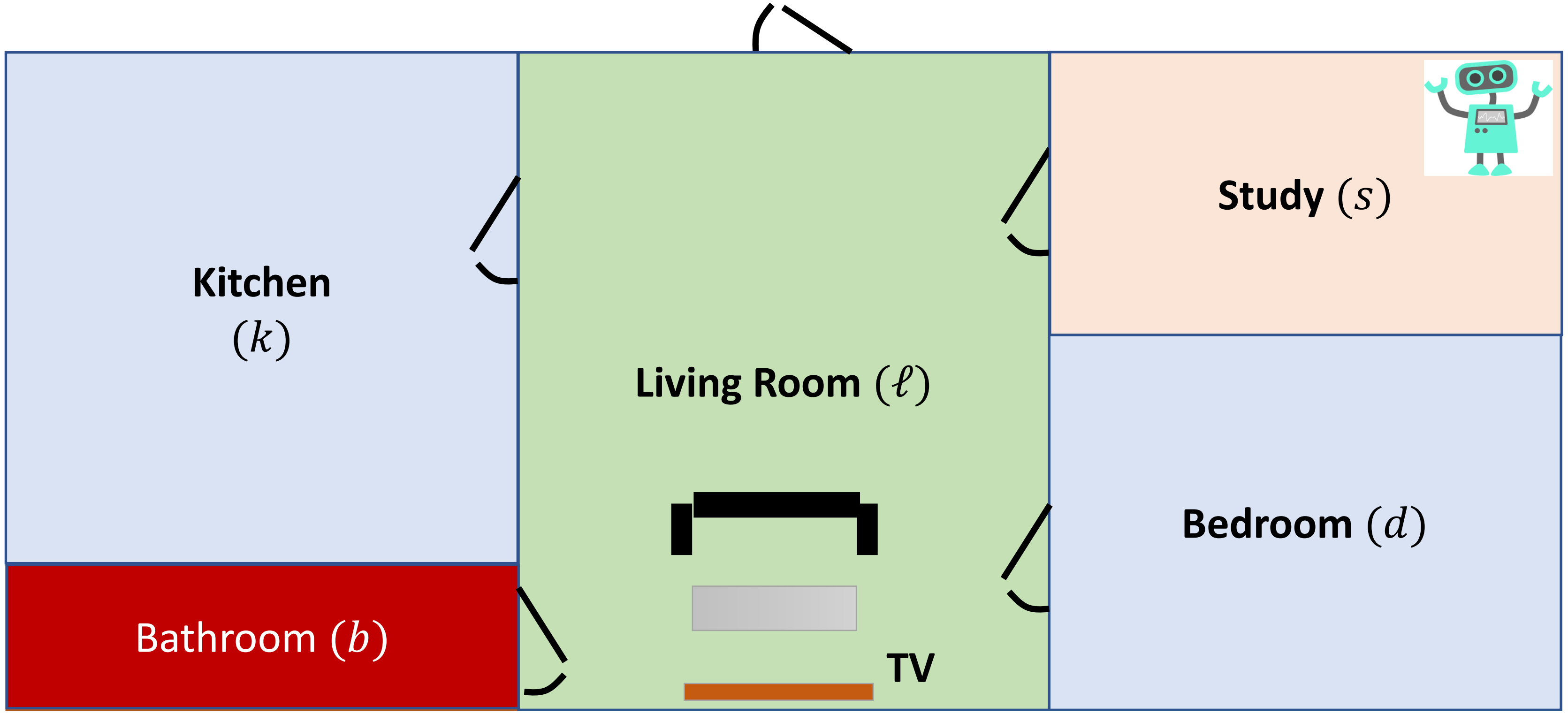
$$\text{step}(y, t) \coloneqq y(t+\delta) - y(t) > a$$

Settling Time:

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

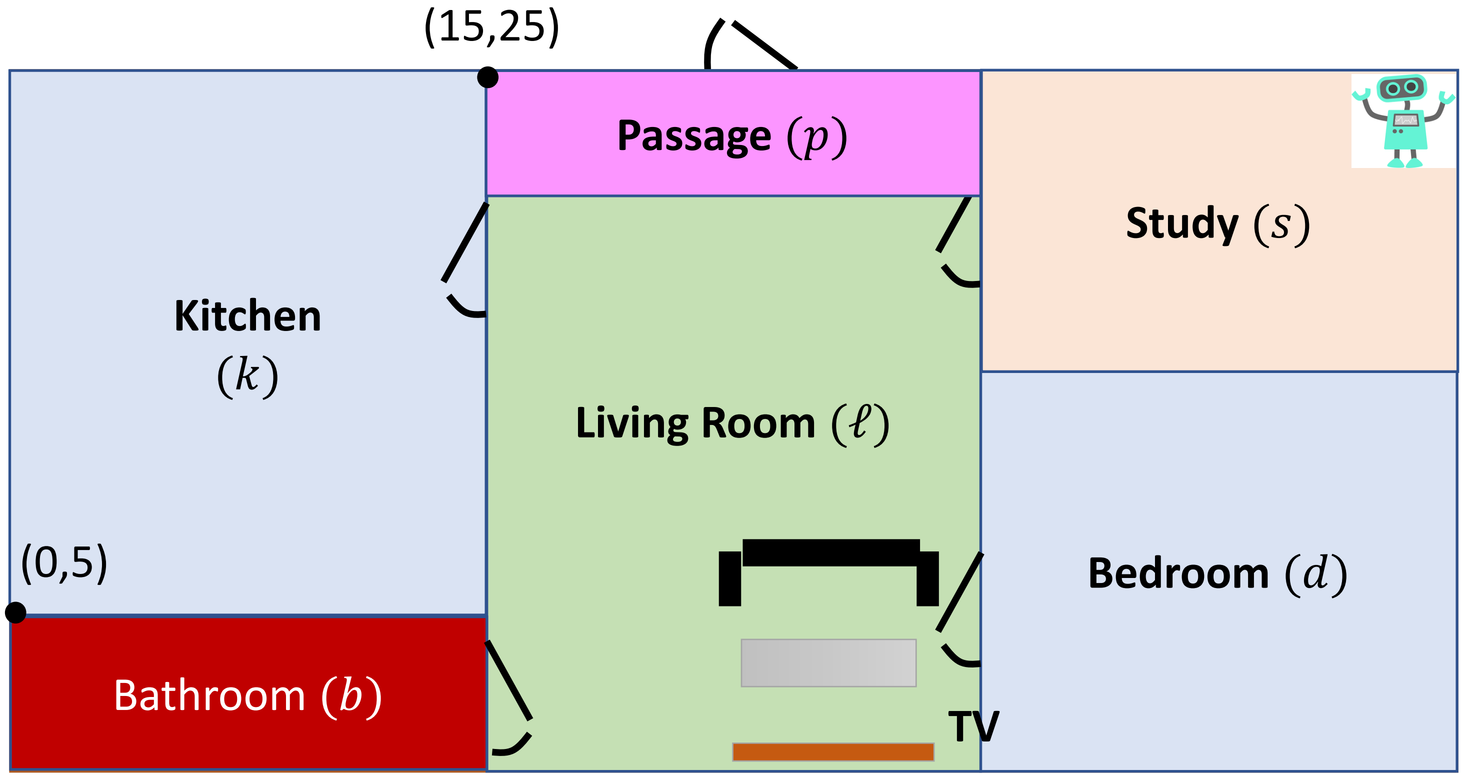
Example specifications in LTL

- 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.

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

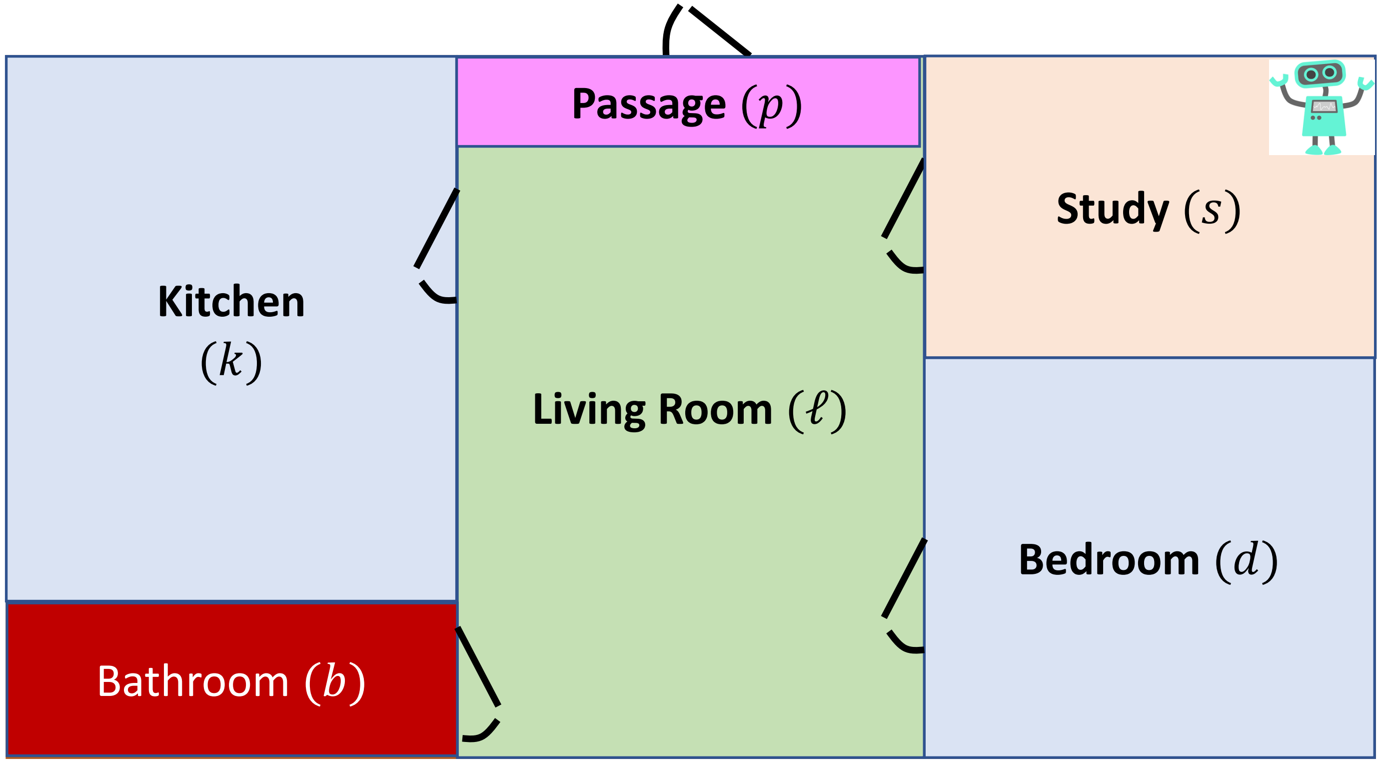
B_r: Box describing room r

$p(t)$: 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

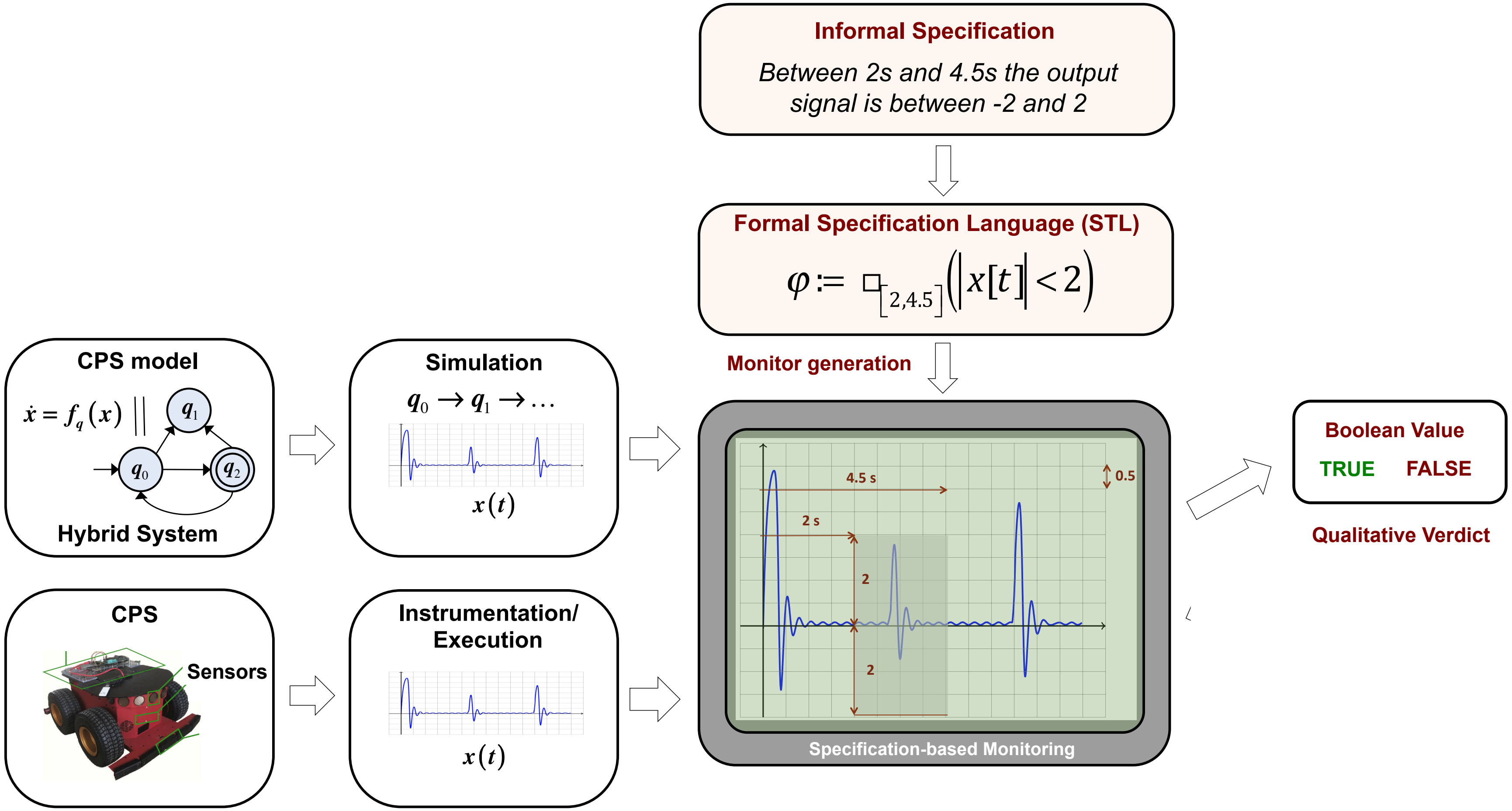


- The robot battery should last between 4 hours and 6 hours $(\mathrm{Q}(\mathrm{t}) \geq Q_{low})\,\mathbf{U}_{[240,360]}(\mathrm{Q}(\mathrm{t}) < \mathrm{Q}_{\mathrm{low}})$

- For the first 10 hours, the robot is never in any room for more than 30 minutes

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

Specification-based Monitoring



Informal Specification

Between 2s and 4.5s the output signal is between -2 and 2

Formal Specification Language (STL)

$$\varphi := \square_{[2,4.5]}\left(\left|x[t]\right| < 2\right)$$

Monitor generation

CPS model

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

Hybrid System

Simulation

$q_0 \to q_1 \to \ldots$

$x(t)$

CPS

Sensors

Instrumentation/ Execution

$x(t)$

Specification-based Monitoring

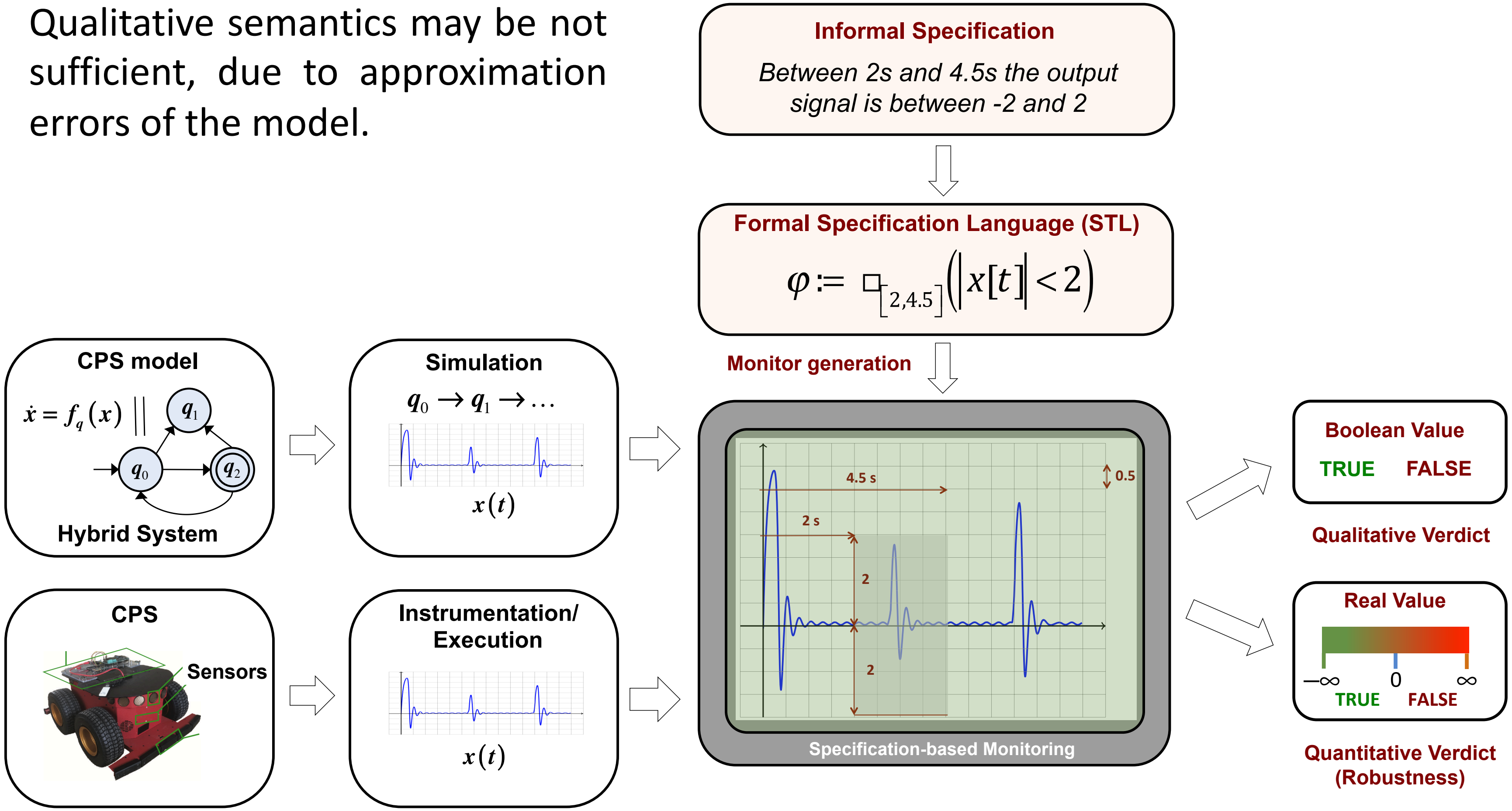
Boolean Value

TRUE **FALSE**

Qualitative Verdict

Specification-based Monitoring

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



Informal Specification

Between 2s and 4.5s the output signal is between -2 and 2

Formal Specification Language (STL)

$$\varphi := \square_{[2,4.5]}\left(\left|x[t]\right| < 2\right)$$

Monitor generation

CPS model

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

Hybrid System

Simulation

$q_0 \to q_1 \to \ldots$

$x(t)$

CPS

Sensors

Instrumentation/ Execution

$x(t)$

Specification-based Monitoring

Boolean Value

TRUE FALSE

Qualitative Verdict

Real Value

$-\infty$ 0 ∞

TRUE FALSE

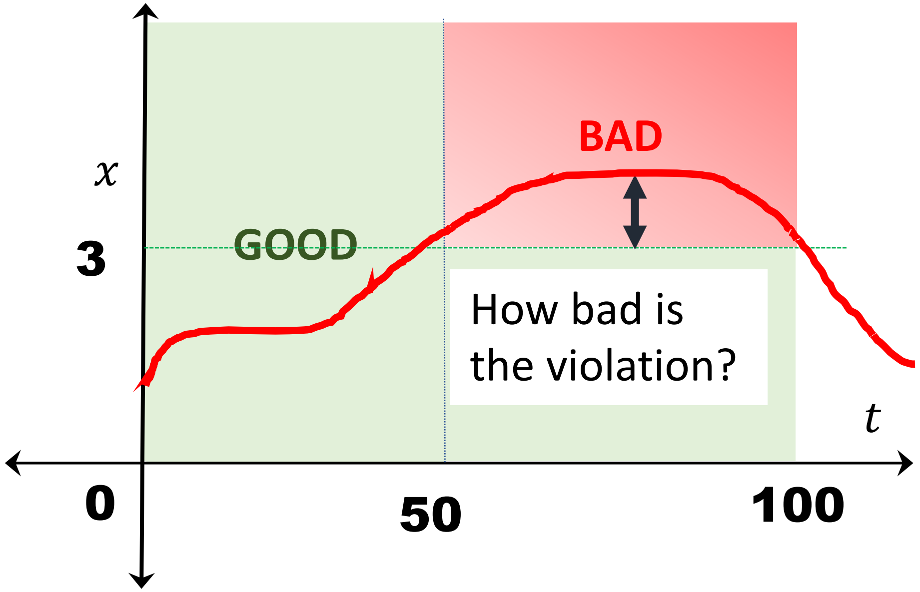
Quantitative Verdict (Robustness)

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

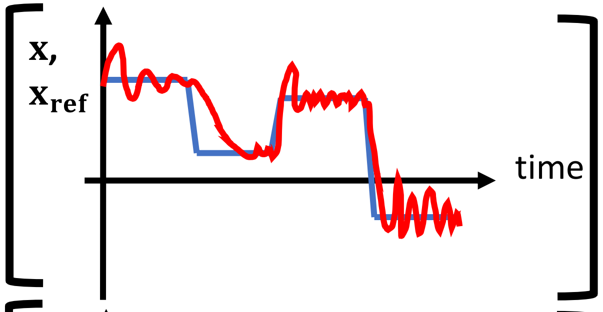
Distance to violation/satisfaction



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

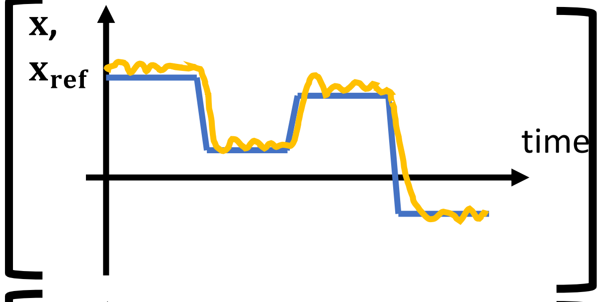
How do quantitative semantics help our engineer?

f_φ



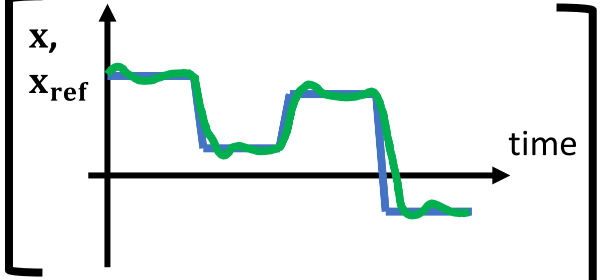
$= -0.2$

f_φ



$= 0.01$

f_φ



$= 0.2$

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

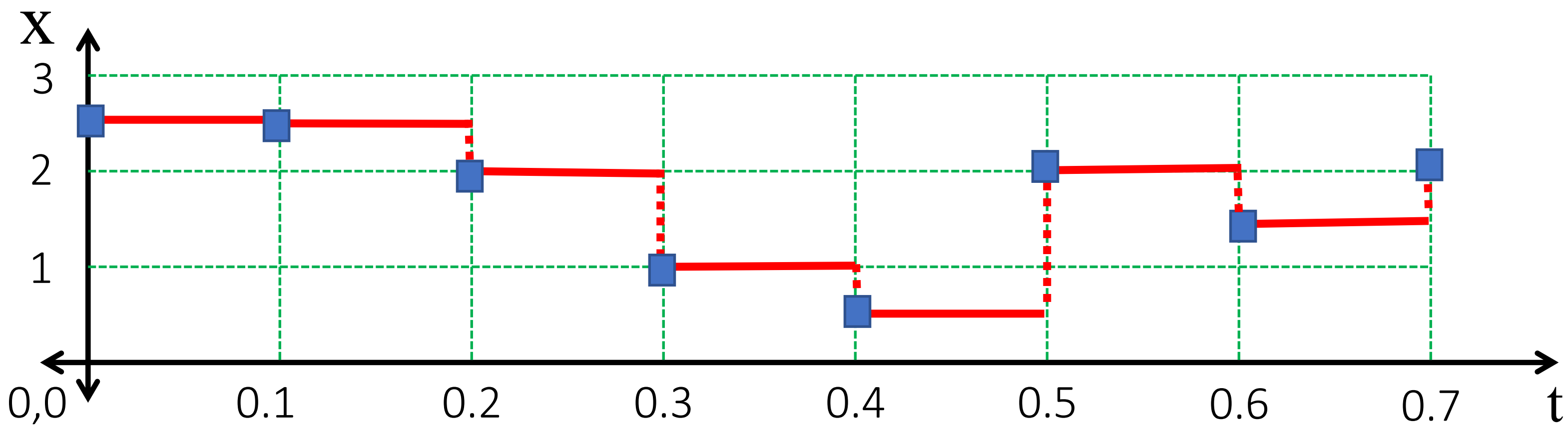
Recursive Quantitative Semantics

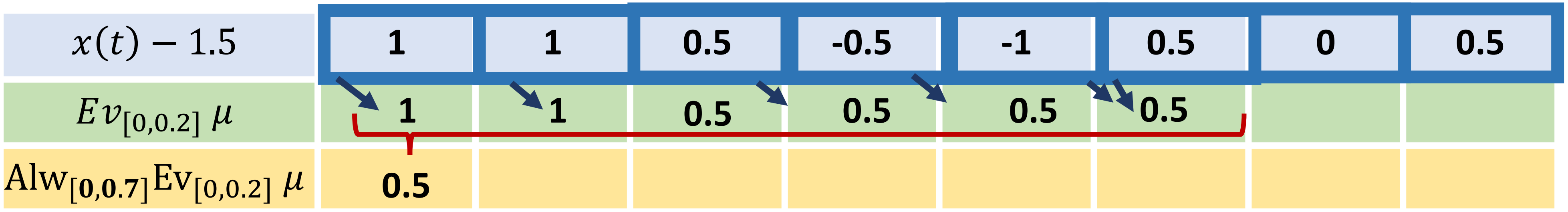
φ	$\rho(\varphi, \mathbf{x}, t)$
$f(\mathbf{x}) > 0\ ,\ f(\mathbf{x}) \geq 0$	$f(\mathbf{x}(t))$
$\neg\varphi$	$-\rho(\varphi, \mathbf{x}, t)$
$\varphi_1 \wedge \varphi_2$	$\min\big(\rho(\varphi_1, \mathbf{x}, \mathrm{t}) \wedge \rho(\varphi_2, \mathbf{x}, \mathrm{t})\big)$
$\mathbf{F}_{[a,b]}\varphi$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

Robustness computation example

$$\boldsymbol{\varphi} \equiv$$

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





$x(t) - 1.5$	**1**	**1**	**0.5**	**-0.5**	**-1**	**0.5**	**0**	**0.5**
$Ev_{[0,0.2]}\ \mu$	**1**	**1**	**0.5**	**0.5**	**0.5**	**0.5**		
$\mathrm{Alw}_{\mathbf{[0,0.7]}}\mathrm{Ev}_{[0,0.2]}\ \mu$	**0.5**							

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

Property of Robust Satisfaction Signal

- Sign indicates satisfaction status (soundness):

$$\begin{aligned} \rho(\varphi, \mathbf{x}, t) > 0 &\Rightarrow \beta(\varphi, \mathbf{x}, t) = 1 \\ \rho(\varphi, \mathbf{x}, t) < 0 &\Rightarrow \beta(\varphi, \mathbf{x}, t) = 0 \end{aligned}$$

- Absolute value indicates tolerance (correctness)

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

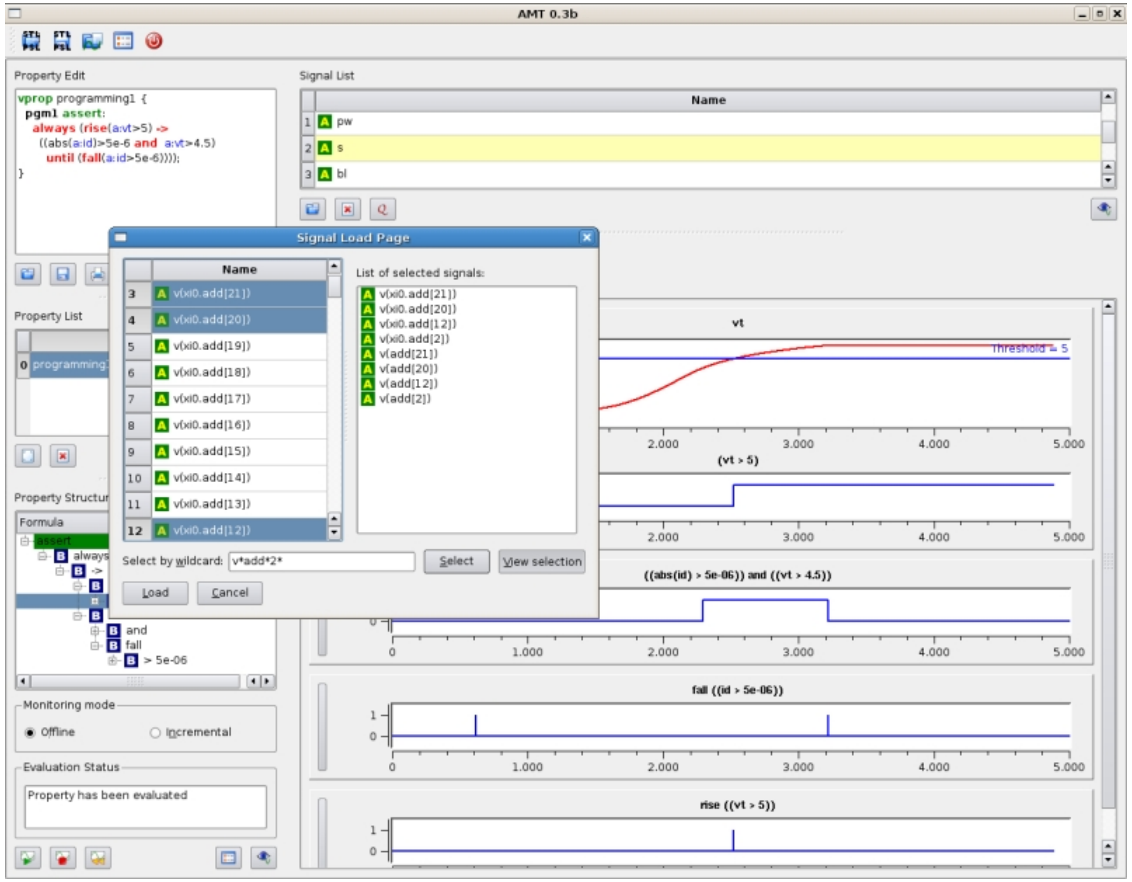
The many uses of STL

- 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

- STL with qualitative semantics
  - Correctness
- Offline monitoring
- Incremental monitoring



```
vprop programming1 {
 pgm1 assert:
  always (rise(a:vt>5) ->
   ((abs(a:id)>5e-6 and  a:vt>4.5)
    until (fall(a:id>5e-6))));
}
```

Signal List:

	Name
1	pw
2	s
3	bl

Property List: 0 programming1

Signal Load Page:

	Name
3	v(xi0.add[21])
4	v(xi0.add[20])
5	v(xi0.add[19])
6	v(xi0.add[18])
7	v(xi0.add[17])
8	v(xi0.add[16])
9	v(xi0.add[15])
10	v(xi0.add[14])
11	v(xi0.add[13])
12	v(xi0.add[12])

List of selected signals:
v(xi0.add[21])
v(xi0.add[20])
v(xi0.add[12])
v(xi0.add[2])
v(add[21])
v(add[20])
v(add[12])
v(add[2])

Select by wildcard: v*add*2*

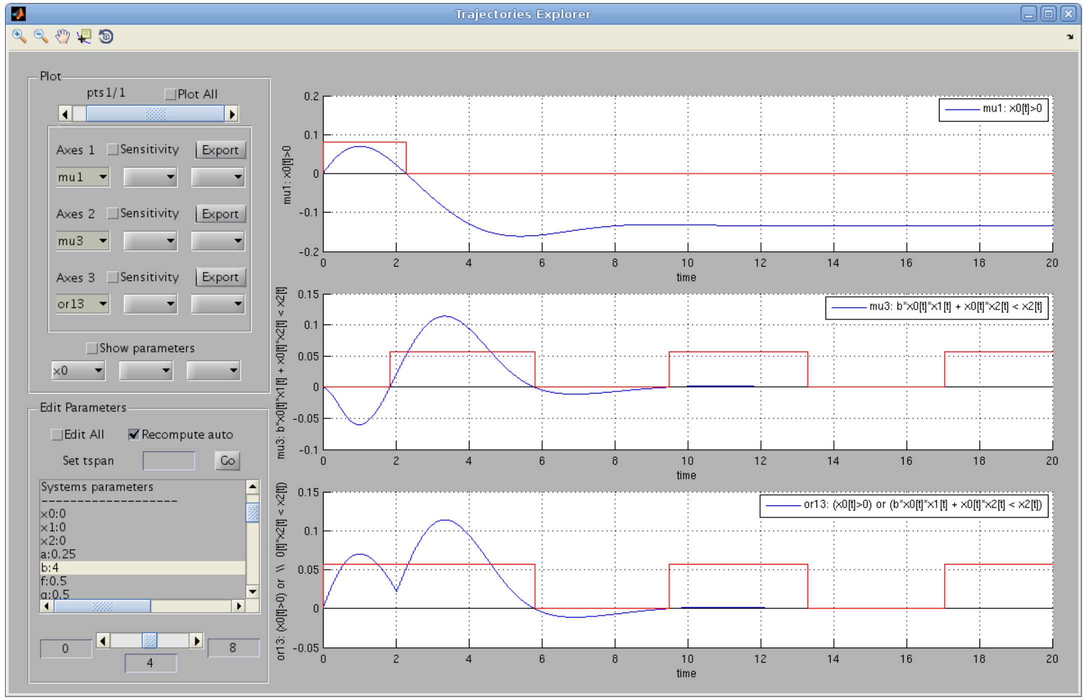
Monitoring mode: ● Offline ○ Incremental

Evaluation Status: Property has been evaluated

Breach

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

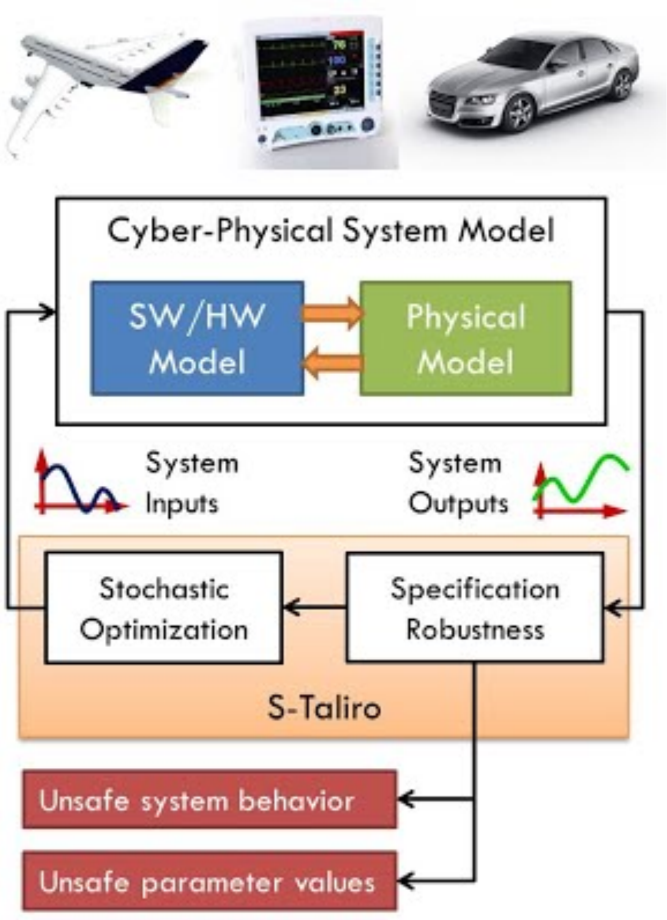
- STL with qualitative and quantitative semantics
  - Correctness
  - Robustness



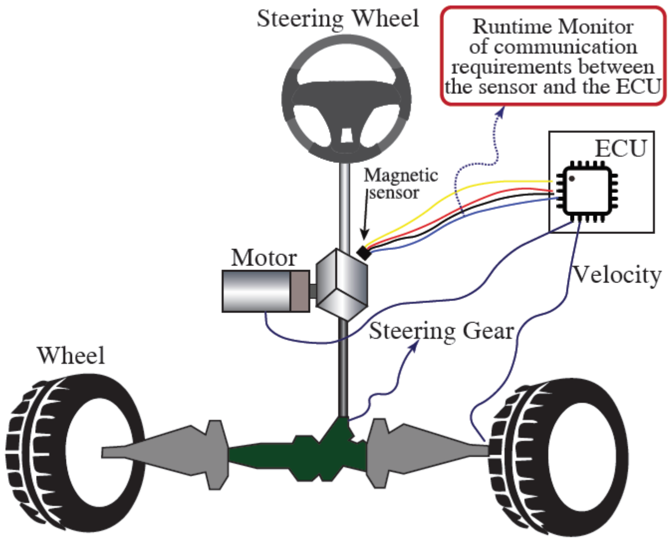
Systems parameters
----
x0:0
x1:0
x2:0
a:0.25
b:4
f:0.5
g:0.5

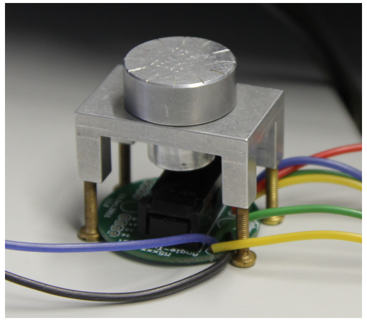
S-TaLiRo

- 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



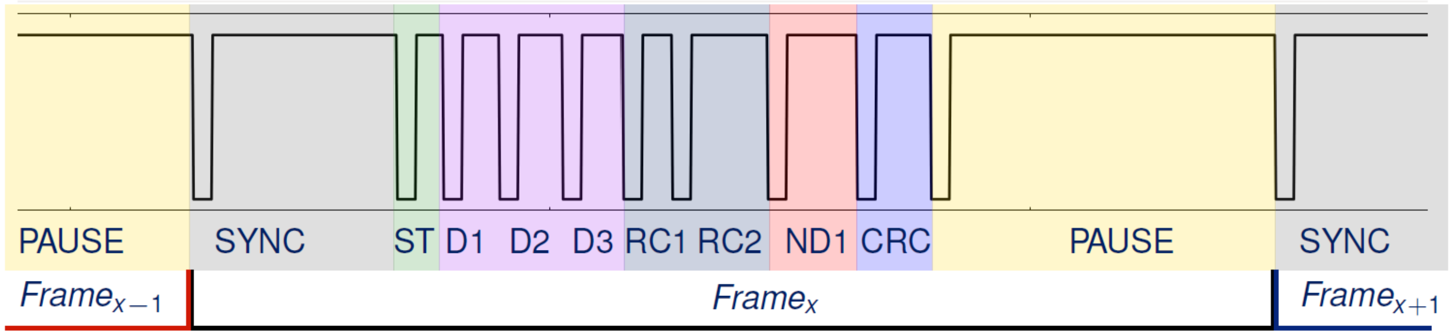
Hardware Monitoring of STL





Magnetic Angular Sensor

- Formalize SENT protocol requirements
  - STL & TRE
- Real-Time Correctness Monitors
  - With Recovery



K. Selyunin, S. Jakšić, T. Nguyen, C. Reidl, U. Hafner, E. Bartocci, D. Ničković, R. Grosu:
Runtime Monitoring with Recovery of the SENT Communication Protocol, CAV 2017

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.