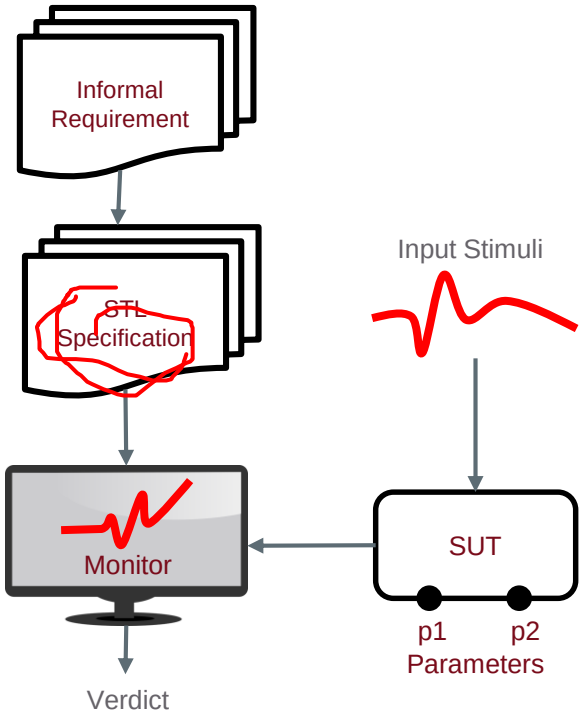
MIXED-TIME SIGNAL TEMPORAL LOGIC

FORMATS 2019

Thomas Ferrère – IST Austria
Oded Maler – VERIMAG
Dejan Nickovic – AIT Austrian Institute of Technology

INTRODUCTION

- Cyber-Physical Systems (CPS)
  - Heterogeneous components
  - SW, Sensors, Actuations, uC, etc.
- CPS are often safety critical
  - → model-based development (MBD)
  - → verification and testing
- Specification-based testing for CPS
  - Signal Temporal Logic (STL)
    - Declarative properties of CPS
  - STL monitoring as basic technology



HETEROGENEITY OF CPS

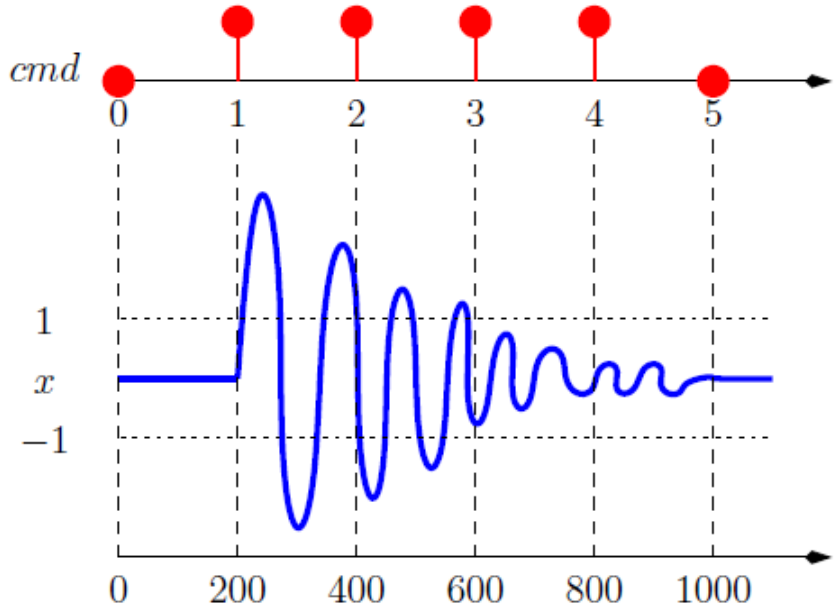
- Heterogeneous components in CPS
- MBD with heterogeneous models of computation
  - Ptolemy
  - MathWorks tools
    - Simulink, SimScape, SimEvents, etc.
  - Scade
  - Verilog AMS, VHDL AMS
- What about verification and testing?

- Specification-based testing for CPS
  - STL: only dense interpretation of time
  - Sensors, actuators, analog components
    - Dense time
  - Digital controllers
    - Discrete (clocked) time
- How to specify and evaluate system-level properties with different time domains?

MOTIVATING EXAMPLE

- Bounded stabilization property
  - Digital command cmd
  - Analog response x

- Whenever cmd is on its rising edge, the absolute value of x must become lower than 1 within 600 time units and remain continuously within that range for at least 300 time units
  - Sampling period $T = 200$ time units



MIXED-TIME SIGNAL TEMPORAL LOGIC (STL-MX)

- Two specification layers
  - Discrete-time layer φ
    - LTL with past
  - Continuous-time layer α
    - STL with past
- **Time mapping operators** to "switch" between layers
  - $@^{dc}$ - from discrete to continuous-time layer
  - $@^{cd}$ - from continuous to discrete-time layer

- Syntax

$\varphi \coloneqq p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid X\,\varphi \mid P\,\varphi \mid \varphi_1 U \varphi_2 \mid \varphi_1 S \varphi_2 \mid @^{cd}(\alpha)$

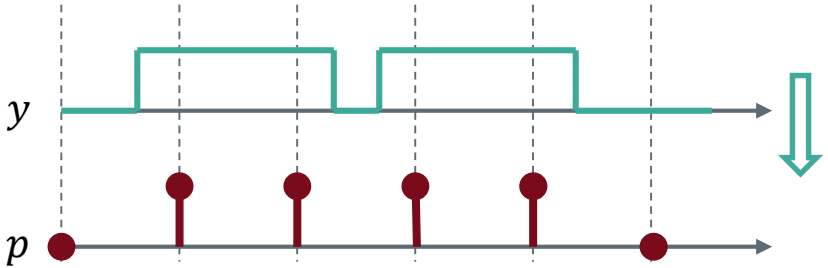
$\alpha \coloneqq x \preccurlyeq c \mid \neg a \mid \alpha_1 \vee \alpha_2 \mid \alpha_1 U_I \alpha_2 \mid \alpha_1 S_I \alpha_2 \mid @^{dc}(\varphi)$

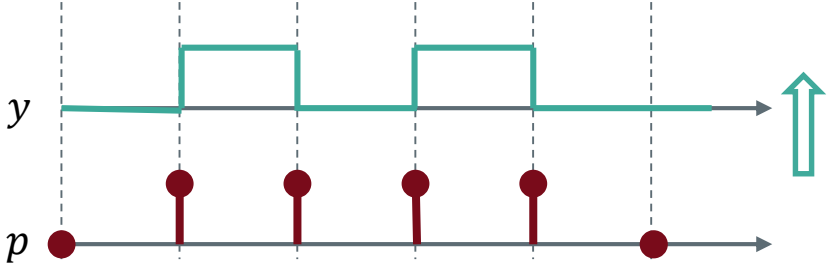
- X – next, P – previously, U – until, S – since
- Other combinatorial and temporal operators derived in standard way
  - $\wedge, \rightarrow, \leftrightarrow$
  - G – always, F – eventually
  - H – historically, O - once

STL-MX SEMANTICS

Time mapping operators

- $p = @^{cd}(y)$
- $y = @^{dc}(p)$



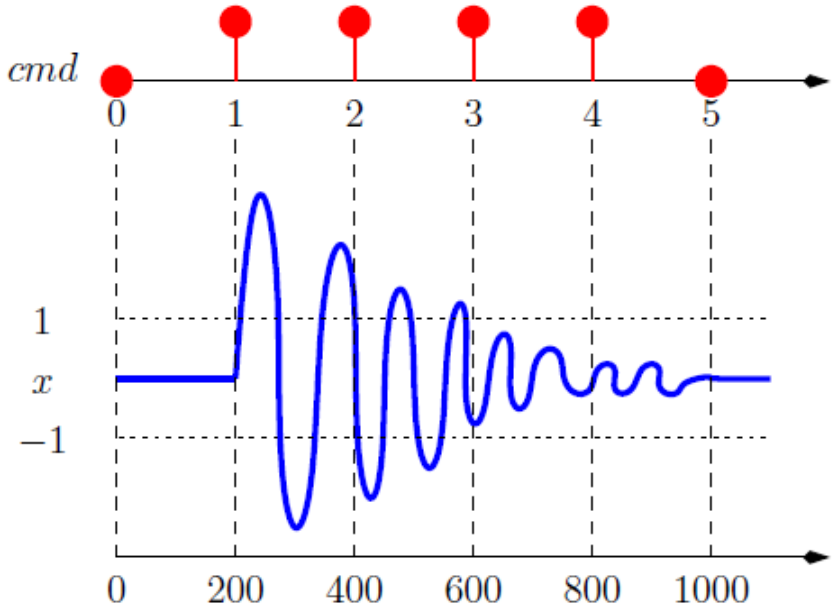


MOTIVATING EXAMPLE REVISITED

- Whenever cmd is on its rising edge, the absolute value of x must become lower than 1 within 600 time units and remain continuously within that range for at least 300 time units
  - Sampling period $T = 200$ time units

- STL-MX specification

$$G((P\neg cmd \wedge cmd) \rightarrow @^{cd}\left(F_{[0,600]}G_{[0,300]}|x| \leq 1\right))$$



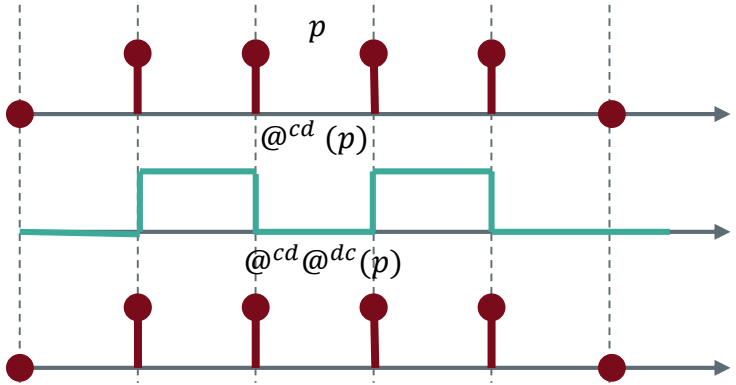
STL-MX FORMULA EQUIVALENCE

- Discrete-time formula equivalence
  - $\varphi \sim \varphi'$ iff for all signals u, w and time indices i, $(u, w, i) \vDash^d \varphi \leftrightarrow (u, w, i) \vDash^d \varphi'$

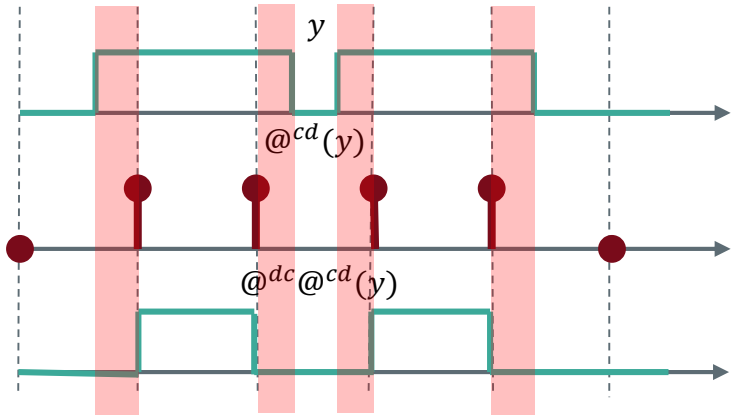
- Continuous-time formula equivalence
  - $\alpha \sim \alpha'$ iff for all signals u, w and real time values t, $(u, w, t) \vDash^c \alpha \leftrightarrow (u, w, t) \vDash^c \alpha'$

STL-MX PROPERTIES

- For all φ, $\varphi = @^{cd}@^{dc}(\varphi)$



- There exists α, s.t. $\alpha \neq @^{dc}@^{cd}(\alpha)$



STL-MX PROPERTIES

- Time mapping operators commute over Boolean connectives

$@^{dc}(\neg\varphi) = \neg@^{dc}(\varphi)$	$@^{dc}(\varphi_1 \vee \varphi_2) = @^{dc}(\varphi_1) \vee @^{dc}(\varphi_2)$
$@^{cd}(\neg\alpha) = \neg@^{cd}(\alpha)$	$@^{cd}(\alpha_1 \vee \alpha_2) = @^{cd}(\alpha_1) \vee @^{cd}(\alpha_2)$

EXPRESSIVITY OF STL-MX

- STL-MX ≈ STL + clock event clk

- Example: clock event clk with period T is continuous time signal
  - $true$ at multiples of T
  - $false$ otherwise

- **Every STL-MX formula can be mapped to STL**
  - Syntactic mapping σ
  - → Polynomial-time reduction

STL-MX to STL mapping

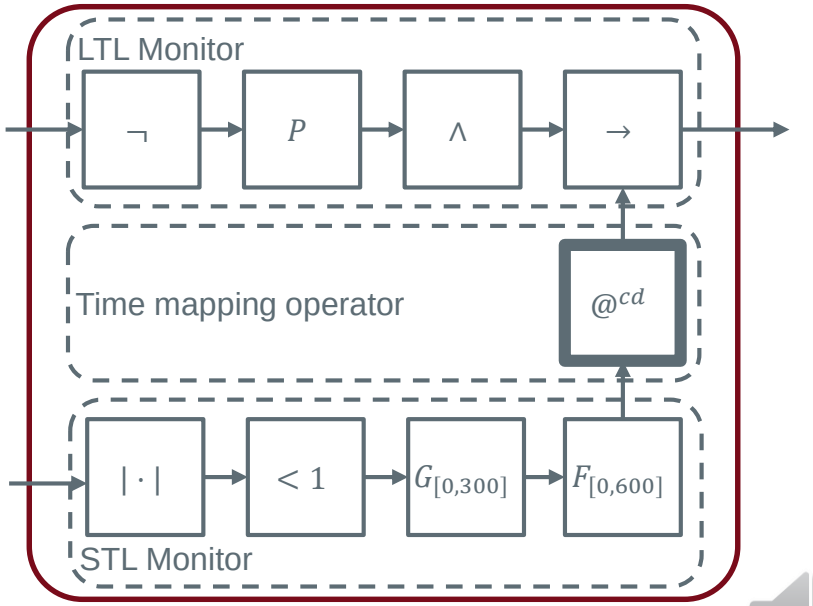
- $\sigma(p) = p$
- $\sigma(X\varphi) = \neg clk U_{(0,\infty)}\big(clk \wedge \sigma(\varphi)\big)$
- $\sigma(\varphi_1 U \varphi_2) = \sigma(\varphi_2) \vee (\sigma(\varphi_1) U_{(0,\infty)} \sigma(\varphi_2))$
- $\sigma\left(@^{cd}(\alpha)\right) = \neg clk\ S(clk \wedge \sigma(\alpha))$

MONITORING STL-MX

- Discrete-time part
  - → LTL monitor – temporal testers
- Dense-time part
  - → STL monitor – temporal testers

- Combining LTL + STL monitors
  - → time mapping operators
  - **Monitor for $@^{cd}$**
  - **Monitor for $@^{dc}$**

Monitor for the bounded stabilization property



MONITORING STL-MX

Monitor for $@^{cd}$

- **Input:** CT signal u, sampling period T
- **Output:** DT signal $w = @^{cd}(u)$

- $I(u) = I_1 \cdot I_2 \cdots I_n$ is a time partition consistent with u
- $k := 0$
- for every time interval I_j
  - while $kT \in I_j$
    - $w(k) = u(I_j)$
    - $k := k + 1$

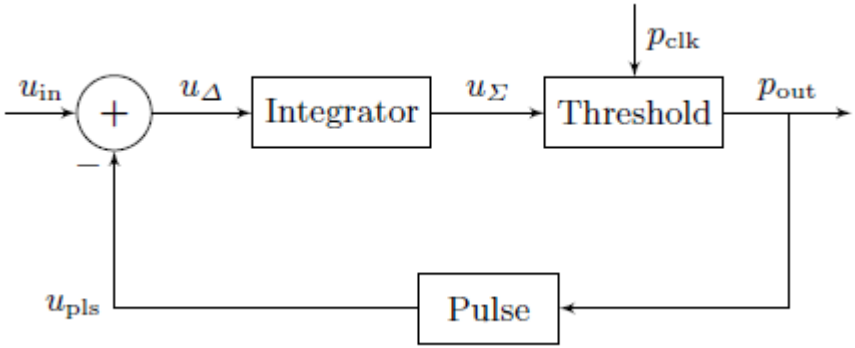
Monitor for $@^{dc}$

- **Input:** DT signal w, sampling period T
- **Output:** CT signal $u = @^{dc}(w)$

- for every time index k in w
  - $I_k = [kT, (k+1)T)$
    - $u(I_k) = w(k)$

CASE STUDY: Δ – Σ MODULATOR

- Δ – Σ modulator
- Subtractor
  - $u_\Delta(t) = u_{in}(t) - u_{pls}(t)$
- Integrator
  - $u_\Sigma(t) = A \cdot \int_0^t u_\Delta(t')dt'$
- Threshold
  - $p_{out}(i) = \begin{cases} 1, & u_\Sigma(iT) \geq v_0 \\ 0, & otherwise \end{cases}$
- Pulse
  - $u_{pls}(t) = \begin{cases} v_1, p_{out}\left(\left\lfloor \frac{t}{T} \right\rfloor - 1\right) = 0 \wedge p_{out}\left(\left\lfloor \frac{t}{T} \right\rfloor\right) = 1 \\ v_0, otherwise \end{cases}$
- Sampling period $T = 3.2\mu s$



CASE STUDY: PROPERTY SPECIFICATION

Property 1

- When we observe a rising edge in the output, the voltage out of the integrator has to return to a value below the threshold at the next clock tick

- STL-MX specification φ_1:

$$G((P\neg p_{out} \wedge p_{out}) \rightarrow X@^{cd}(u_{\Sigma} < v_0)$$

Property 2

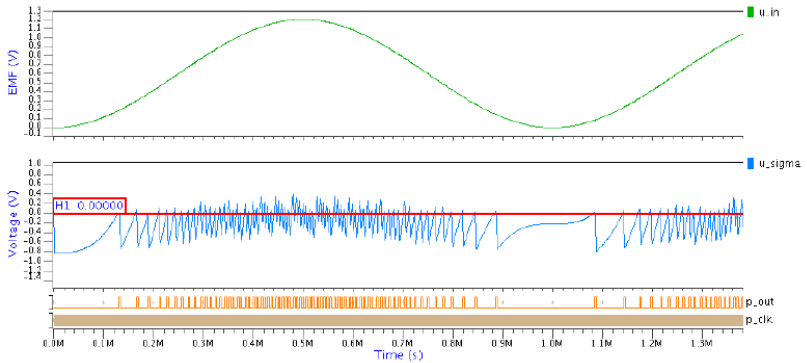
- When the input voltage is above $1.05V$ for $12.8\mu s$ the output must have a sequence of two consecutive spikes starting over that time frame

- STL-MX specification φ_2:

$$G(G_{[0,12.8]}(u_{in} > 1.05) \rightarrow F_{[0,12.8]}@^{dc}(\neg p_{out} \wedge Xp_{out} \wedge X^2\neg p_{out} \wedge X^3 p_{out}))$$

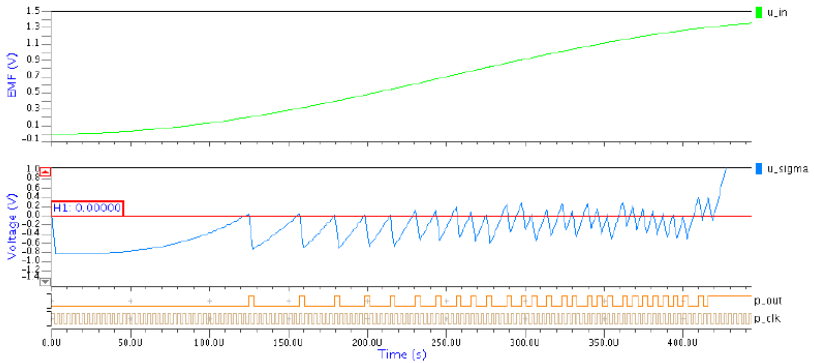
CASE STUDY: SIMULATION AND EVALUATION

$$u_{in}(t) = 0.6\,cos(1000 \cdot 2\pi \cdot t) + 0.6$$



φ_1 **satisfied**

$$u_{in}(t) = 0.7\,cos(1000 \cdot 2\pi \cdot t) + 0.7$$



φ_1 **violated**

CASE STUDY: EXECUTION TIMES

Property	Sim #	u_{Σ}	u_{in}	p_{out}	time(ms)
φ_1	1	20,470		727	143
φ_1	2	2,771		58	104
φ_2	3		26,207	971	45
φ_2	4		27,926	971	50
φ_2	5		29,495	971	51
φ_2	6		31,298	1,212	58
φ_2	7		32,133	1,212	59
φ_2	8		33,005	1,212	61

CASE STUDY: STL-MX VS. STL

- STL-MX specification φ_2:

$$G(G_{[0,12.8]}(u_{in} > 1.05) \rightarrow F_{[0,12.8]}@^{dc}(\neg p_{out} \wedge X p_{out} \wedge X^2 \neg p_{out} \wedge X^3 p_{out}))$$

- STL specification $\sigma(\varphi_2)$:

$$G(G_{[0,12.8]}(u_{in} > 1.05) \rightarrow F_{[0,12.8]} \begin{pmatrix} \neg p_{out} \wedge \\ \neg clkU(clk \wedge p_{out}) \wedge \\ \neg clkUclk \wedge (\neg clkU(clk \wedge \neg p_{out})) \wedge \\ \neg clkUclk \wedge (\neg clkU\left(clk \wedge \left(\neg clkU(clk \wedge p_{out})\right)\right)) \end{pmatrix})$$

FUTURE WORK

- Automatic insertion of @cd and @dc conversion operators based on type inference
  - Facilitate use of the specification language
- More sophisticated conversion operators
  - Instead of periodic sample and hold.
  - Truth value of discrete signal depends on integrating values at continuous time in some interval around it
  - Event-based conversion in asynchronous style
- Tighter interaction between the monitoring procedure and the simulators
- Equipping STL-mx with quantitative semantics

CONCLUSIONS

- STL-MX
  - Syntactic and semantic constructs
  - Co-existence of discrete and continuous-time specifications
  - Main application - runtime monitoring of CPS and mixed signal designs

- Step towards system-wide specification-based verification

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

THANK YOU!

Lecturer, Date