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 $\psi$
    - LTL with past
  - Continuous-time layer $\alpha$
    - 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
  $\psi ::= p \mid \neg \psi \mid \psi_1 \lor \psi_2 \mid X \psi \mid P \psi \mid \psi_1 U \psi_2 \mid \psi_1 S \psi_2 \mid @^{cd}(\alpha)$
  $\alpha ::= x \preceq c \mid \neg \alpha \mid \alpha_1 \lor \alpha_2 \mid \alpha_1 U_1 \alpha_2 \mid \alpha_1 S_1 \alpha_2 \mid @^{dc}(\psi)$

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

Time mapping operators

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

$$G((P \neg cmd \land cmd) \rightarrow @^{cd} (F_{[0,600]} G_{[0,300]} |x| \leq 1))$$
STL-MX FORMULA EQUIVALENCE

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

• Continuous-time formula equivalence
  • $\alpha \sim \alpha'$ iff for all signals $u, w$ and real time values $t$, $(u, w, t) \models^c \alpha \iff (u, w, t) \models^c \alpha'$
STL-MX PROPERTIES

- For all $\varphi$, $\varphi = @(cd)(\varphi)$

- There exists $\alpha$, s.t. $\alpha \neq @(dc)(\alpha)$
STL-MX PROPERTIES

- Time mapping operators commute over Boolean connectives

\[
\begin{align*}
@^{dc}(\neg \varphi) &= \neg @^{dc}(\varphi) \\
@^{cd}(\neg \alpha) &= \neg @^{cd}(\alpha) \\
@^{dc}(\varphi_1 \lor \varphi_2) &= @^{dc}(\varphi_1) \lor @^{dc}(\varphi_2) \\
@^{cd}(\alpha_1 \lor \alpha_2) &= @^{cd}(\alpha_1) \lor @^{cd}(\alpha_2)
\end{align*}
\]
EXPRESSIVITY OF STL-MX

- STL-MX \approx STL + \text{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 \( \sigma \)
  - \( \rightarrow \) Polynomial-time reduction

STL-MX to STL mapping

\[
\begin{align*}
\sigma(p) &= p \\
\sigma(X\varphi) &= \neg clk U_{(0,\infty)}(clk \land \sigma(\varphi)) \\
\sigma(\varphi_1 U \varphi_2) &= \sigma(\varphi_2) \lor (\sigma(\varphi_1) U_{(0,\infty)} \sigma(\varphi_2)) \\
\sigma \left( @^{cd} (\alpha) \right) &= \neg clk S(clk \land \sigma(\alpha))
\end{align*}
\]
MONITORING STL-MX

• Discrete-time part
  • \( \rightarrow \) LTL monitor – temporal testers

• Dense-time part
  • \( \rightarrow \) STL monitor – temporal testers

• Combining LTL + STL monitors
  • \( \rightarrow \) time mapping operators
  • \textbf{Monitor for} \( @^{cd} \)
  • \textbf{Monitor for} \( @^{dc} \)

Monitor for the bounded stabilization property
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: $\Delta - \Sigma$ MODULATOR

- $\Delta - \Sigma$ 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
  - $p_{out}(i) = \begin{cases} 
    1, & u_\Sigma(iT) \geq v_0 \\
    0, & \text{otherwise}
  \end{cases}$
- Pulse
  - $u_{pls}(t) = \begin{cases} 
    v_1, & p_{out} \left( \left\lfloor \frac{t}{T} \right\rfloor - 1 \right) = 0 \land p_{out} \left( \left\lfloor \frac{t}{T} \right\rfloor \right) = 1 \\
    v_0, & \text{otherwise}
  \end{cases}$
- 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 $\varphi_1$:

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

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 $\varphi_2$:

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

05/09/2019
CASE STUDY: SIMULATION AND EVALUATION

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

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

\[ \phi_1 \text{ satisfied} \]

\[ \phi_1 \text{ violated} \]
## CASE STUDY: EXECUTION TIMES

<table>
<thead>
<tr>
<th>Property</th>
<th>Sim #</th>
<th>$u_{\Sigma}$</th>
<th>$u_{in}$</th>
<th>$p_{out}$</th>
<th>time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_1$</td>
<td>1</td>
<td>20,470</td>
<td></td>
<td>727</td>
<td>143</td>
</tr>
<tr>
<td>$\varphi_1$</td>
<td>2</td>
<td>2,771</td>
<td></td>
<td>58</td>
<td>104</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>3</td>
<td>26,207</td>
<td></td>
<td>971</td>
<td>45</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>4</td>
<td>27,926</td>
<td></td>
<td>971</td>
<td>50</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>5</td>
<td>29,495</td>
<td></td>
<td>971</td>
<td>51</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>6</td>
<td>31,298</td>
<td></td>
<td>1,212</td>
<td>58</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>7</td>
<td>32,133</td>
<td></td>
<td>1,212</td>
<td>59</td>
</tr>
<tr>
<td>$\varphi_2$</td>
<td>8</td>
<td>33,005</td>
<td></td>
<td>1,212</td>
<td>61</td>
</tr>
</tbody>
</table>
CASE STUDY: STL-MX VS. STL

- STL-MX specification $\varphi_2$:
  \[ G(G_{0,12.8}(u_{in} > 1.05) \rightarrow F_{0,12.8}^{dc}(\neg p_{out} \land Xp_{out} \land X^2\neg p_{out} \land X^3 p_{out})) \]

- STL specification $\sigma(\varphi_2)$:
  \[ G(G_{0,12.8}(u_{in} > 1.05) \rightarrow F_{0,12.8} \left( \neg p_{out} \land \neg clkU(clk \land p_{out}) \land \neg clkUclk \land (\neg clkU(clk \land \neg p_{out})) \land \neg clkUclk \land (\neg clkU(clk \land (\neg clkU(clk \land p_{out})))) \right) \]
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
THANK YOU!

Lecturer, Date