Formal Timed Analysis of Mixed Music Scores

Léa Fanchon, Florent Jacquemard

Ircam & INRIA

ICMC 2013 – Perth
Writing Mixed Music
Antescofo (Anticipatory Score Following, Ircam 2007-2013)

http://repmus.ircam.fr/antescofo

![Diagram of Antescofo system]

- Mixed score
- Listening machine
- Reactive machine
- Audio environment (MAX/MSP)

- Continuous signal
- Discrete input events
- Discrete output actions
Antescofo (Anticipatory Score Following, Ircam 2007-2013)

http://repmus.ircam.fr/antescofo

interpretation trace = interleaving of input events and output actions with dates
"Ideal" Trace

http://repmus.ircam.fr/antescofo

continuous signal

ideal trace $\simeq$ the mixed score
Real Trace

http://repmus.ircam.fr/antescofo

![Diagram of audio and reactive machine processes](image)

Continuous signal: listening machine → discrete input events → discrete output actions

Discrete signal: reactive machine → audio env. (MAX/MSP)

(real trace ≠ ideal trace)
Example

Microscore

note $e_1$ 1.0
0.0 group $g_1$ \{(0.5 \text{ init})\}
0.25 group $g_3$ \{(0.5 \text{ msg}) (0.5 \text{ off})\}
note $e_2$ 1.0
0.0 group $g_2$ \{(0.5 \text{ on})\}
note $e_3$ 0.5

Ideal performance

\[
\begin{array}{c}
| & e_1 & 1.0 & e_2 & 1.0 & e_3 \\
\downarrow & & & & & \\
| & g_1 & 0.5 & \text{init} & & \\
\downarrow & & & & & \\
| & g_3 & 0.25 & \text{msg} & 0.5 & \text{off} \\
\end{array}
\]
Example

Microscore

note $e_1$ 1.0
0.0  group $g_1$  \{ (0.5 init) \}
0.25 group $g_3$  \{ (0.5 msg) (0.5 off) \}

note $e_2$ 1.0
0.0  group $g_2$  \{ (0.5 on) \}

note $e_3$ 0.5

Unexpected performance

\[
\begin{align*}
  e_1 & \xrightarrow{\delta_1=0.7} e_2 & \xrightarrow{\delta_2=0.9} e_3 \\
  g_1 & \xrightarrow{0.5} \text{init} & 0.5 & \text{on} \\
  g_3 & \xrightarrow{0.25} \text{msg} & 0.5 & \text{off}
\end{align*}
\]
Objective: Parameter Synthesis

Static Analysis of Mixed Music Scores

Compute a linear constraint $K$ on $\delta_1$ and $\delta_2$, ensuring that

- trace = ideal trace, or
- on after off in trace.

$K =$ indication of the robustness of mixed scores to the tempo variations during performances.

Computed using software verification techniques developed for realtime embedded software based on timed automata model.
Outline

1. Timed Automata: an abstract model for realtime systems
2. Compilation of mixed scores into timed automata
3. Parameter Synthesis
4. Perspectives
1. Timed Automata
Timed Automata

[Alur & Dill 1990]

+ describes logical ordering of events of the system
+ quantitative timing info (duration and time between events)

modeling realtime systems: transport, embedded systems, communication networks, manufacturing, circuits... for

→ simulation
→ verification
→ optimization
→ controller synthesis
Timed Automata Example

- $x$ is a clock
- transitions: location$_1$ \(\xrightarrow{\text{guard, symbol, \{clocks to reset\}}}\) location$_2$
- symbol belongs to a finite alphabet
- clock constraints $e := x\{<, \leq, \geq, >\}c \mid e \land e \mid true$, $c \in \mathbb{Z}$, for
  - transitions’ guards
  - locations’ invariants (condition to stay)
Timed Automata Example

trace:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x = 0$</td>
<td>$x = 0$</td>
<td>$x = 0$</td>
<td>$x = 0$</td>
<td>$x = 0$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t = 0$</td>
<td>$t = 0$</td>
<td>$t = 0$</td>
<td>$t = 0$</td>
<td>$t = 0$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

untimed trace: click click double
Timed Automata Networks

\[ A_1 \parallel \ldots \parallel A_n \text{ with disjoint locations and clocks} \]
\[ \text{non-disjoint alphabets } \Sigma_1, \ldots, \Sigma_n. \]
Timed Automata Networks

\( \mathcal{A}_1 \parallel \ldots \parallel \mathcal{A}_n \) with disjoint locations and clocks non-disjoint alphabets \( \Sigma_1, \ldots, \Sigma_n \).

Cartesian product: \( \langle \ell_1, \ldots, \ell_n \rangle \xrightarrow{g,a,R} \langle \ell'_1, \ldots, \ell'_n \rangle \) such that

- for all \( 1 \leq i \leq n \), if \( a \notin \Sigma_i \), then \( \ell'_i = \ell_i \)
- otherwise there exists \( g_i \) and \( R_i \) such that \( \ell_i \xrightarrow{g_i,a,R_i} \ell'_i \in \mathcal{A}_i \)

\[ g = \bigwedge_{a \in \Sigma_i} g_i \quad \text{and} \quad R = \bigcup_{a \in \Sigma_i} R_i \]
Timed Automata

infinite state model

state: \( \langle \ell, v \rangle \) where \( \ell \) location and \( v : \text{clocks} \rightarrow \mathbb{R}^+ \)

but... region abstraction: transformable into NFA

+ decision of reachability in EXPTIME, PSPACE-complete
+ symbolic algorithms (zones): Uppaal, HyTech
  - \( \subseteq \) of traces languages undecidable
  - no closure under complement
  - not determinizable
2. Compilation of Antescofo Scores into Timed Automata Networks
Compilation: Antescofo Scores → Timed Automata

note $e_1$ 1.0
0.0 group $g_1$ \{(0.5 init)\}
0.25 group $g_3$ \{(0.5 msg) (0.5 off)\}
note $e_2$ 1.0
0.0 group $g_2$ \{(0.5 on)\}
note $e_3$ 0.5

Timed Automata Network
Nested Groups

note $e_1$ 1.0
0.0 group $g_1$  
  
  $\{(0.5 \text{ init})$
  
  $(0.25 \text{ group } g_3 \ (0.5 \text{ msg}) (0.5 \text{ off})\}$

note $e_2$ 1.0
0.0 group $g_2$  
  
  $(0.5 \text{ on})$

note $e_3$ 0.5

idle $x = 0$

$e_1, x := 0$

$x \leq 1.0$

$e_2, x := 0$

$x \leq 1.0$

$e_3$

$\text{stop}$

idle $y_1 = 0$

$e_1, y_1 := 0$

$y_1 \leq 0.5$

$y_1 = 0.5, \text{ init}$

$y_1 \leq 0.25$

$y_1 = 0.25, g_3$

$\text{stop}$

idle $y_3 = 0$

$g_3, y_3 := 0$

$y_3 \leq 0.75$

$y_3 = 0.5, \text{ msg}, y_3 := 0$

$y_3 \leq 0.5$

$y_3 = 0.5, \text{ off}$

$\text{stop}$

idle $y_2 = 0$

$e_2, y_2 := 0$

$y_2 \leq 0.5$

$y_2 = 0.5, \text{ on}$

$\text{stop}$
Loops

0 loop \( \ell \) \( \{(1 \ a) \ (2 \ b)\} \) until \( \kappa \)

\[
\begin{align*}
\text{idle} 
\quad & e_1, y := 0 \\
\text{y \leq 1} 
\quad & y = 1, a, y := 0 \\
\text{y \leq 2} 
\quad & y = 2, b \\
\text{stop} 
\quad & \text{y < 1, } \kappa \\
\text{stop} 
\quad & \text{y < 2, } \kappa \\
\text{idle} 
\quad & \epsilon, y := 0
\end{align*}
\]
Group Attribute "tight"

Score

<table>
<thead>
<tr>
<th>Note</th>
<th>Score</th>
<th>Group</th>
<th>Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>1.0</td>
<td>$g_1$</td>
<td>(0.5 init)</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td>$g_3$</td>
<td>(0.5 msg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.5 off)</td>
</tr>
<tr>
<td>$e_2$</td>
<td>1.0</td>
<td>$g_2$</td>
<td>(0.5 on)</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_3$</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Almost equivalent score

<table>
<thead>
<tr>
<th>Note</th>
<th>Score</th>
<th>Group</th>
<th>Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>1.0</td>
<td>$g\text{[tight]}$</td>
<td>(0.5 init)</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td></td>
<td>(1.0 on)</td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td>$g_3$</td>
<td>(0.5 msg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.5 off)</td>
</tr>
<tr>
<td>$e_2$</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Timed Automaton for $g$

\[

e_1, \quad y := 0 \\
y \leq 0.5 \\
y = 0.5, \quad \text{init} \\
y < 0.5, y := 0 \\
e_2, y := 0 \\
y = 0.5, \quad \text{on} \\
\]
3. Parameter Synthesis
Parametric Timed Automata

[Alur, Henzinger, Vardi 1993]
parameters $\delta$ for (unknown) time bounds in guards and invariants.

Valuation of Parameters $\delta_1, \delta_2 = \text{performance}$
Good Parameters Problem

[Frehse, Jha, Krogh 2008]
Find a set of parameter values within a rectangular domain $\subseteq \mathbb{R}^m$ for which the behavior of $\mathcal{A}$ is acceptable.

i.e. find a linear constraint $K$ on parameters such that every for valuation of parameters $\pi$ satisfying $K$, the behavior of $\mathcal{A}$ instantiated by $\pi$ ($\mathcal{A}[\pi]$) is acceptable.

3.1 Naïve Approach
3.2 Inverse Method
Symbolic Semantics of Parametric Timed Automata

\[ \mathcal{A}(K) : \text{PTA with a constraint } K \text{ over parameters.} \]

symbolic states \( \langle \ell, C \rangle \) where \( \ell \) location, \( C \) constraint over clocks and parameters.

initial state \( \langle \ell_0, C_0 \rangle \) with \( C_0 = K \land inv(\ell_0) \land \bigwedge_{i=1}^{p-1} x_{i+1} = x_i, \)

transitions \( \langle \ell, C \rangle \xrightarrow{a} \langle \ell', C' \rangle \) such that \( \exists \ell \xrightarrow{g,a,\text{Reset}} \ell' \) in \( \mathcal{A}(K) \) and

\[
C'(x') = \exists x, d \quad C(x) \land g(x + d) \\
\land \bigwedge_{x_i \in \text{Reset}} x'_i = 0 \land \bigwedge_{x_i \notin \text{Reset}} x'_i = x_i + d \land inv(\ell')(x')
\]

where \( x = x_1, \ldots, x_p \), and \( x' = x'_1, \ldots, x'_p \).
3.2 Naive Approach

Definition of acceptable behavior by a control automaton $C$

Naive Algorithm

Depth first exploration of the symbolic state space of $\mathcal{A} \parallel C$.
Prune branches with unsatisfiable constraints.

constraint $K =$
disjunction of the constraints of final symbolic states reached.
Depth First Symbolic State Exploration (example 1)

untimed trace: constraint:

1
Depth First Symbolic State Exploration (example 1)

untimed trace: $e_1$

constraint:

2
Depth First Symbolic State Exploration (example 1)

**Untimed Trace:** $e_1, e_2$

**Constraint:** $\exists y_1, y_3. y_1 = \delta_1 \leq 0.5 \land y_3 = \delta_1 \leq 0.75$
Depth First Symbolic State Exploration (example 1)

untimed trace: \( e_1, e_2 \)
constraint: \( \delta_1 \leq 0.5 \)
Depth First Symbolic State Exploration (example 1)

untimed trace: $e_1, e_2, \text{init}$

constraint: $\delta_1 \leq 0.5 \land \exists x . x = 0.5 - \delta_1 \leq \delta_2$
Depth First Symbolic State Exploration (example 1)

untimed trace: $e_1, e_2, \text{init}$

constraint: $\delta_1 \leq 0.5 \land 0.5 - \delta_2 \leq \delta_1$
Depth First Symbolic State Exploration (example 1)

untimed trace: $e_1, e_2, \text{init}, \text{msg}$

constraint: $\delta_1 \leq 0.5 \land 0.5 - \delta_2 \leq \delta_1 \land$

$\exists x, y_2 \cdot x = 0.75 - \delta_1 \leq \delta_2 \land y_2 = 0.75 - \delta_1 \leq 0.5$
-depth first symbolic state exploration (example 1)

untimed trace: \( e_1, e_2, \text{init}, \text{msg} \)

constraint: \( \delta_1 \leq 0.5 \land 0.75 - \delta_2 \leq \delta_1 \land 0.25 \leq \delta_1 \)
Depth First Symbolic State Exploration (example 1)

Untimed trace: $e_1, e_2, \text{init}, \text{msg}, \text{off}$

Constraint:

\[
\delta_1 \leq 0.5 \land 0.75 - \delta_2 \leq \delta_1 \land 0.25 \leq \delta_1 \land \\
\exists x, y_2 . x = 1.25 - \delta_1 \leq \delta_2 \land y_2 = 1.25 - \delta_1 \leq 0.5
\]
Depth First Symbolic State Exploration (example 1)

untimed trace: \( e_1, e_2, \text{init}, \text{msg}, \text{off} \)

constraint: \( \delta_1 \leq 0.5 \land 0.75 - \delta_2 \leq \delta_1 \land 0.25 \leq \delta_1 \land 1.25 - \delta_2 \leq \delta_1 \land 0.75 \leq \delta_1 \) unsatisfiable!
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace:

constraint:

1
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$

constraint:

2
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$, init

constraint: $\exists x. x = 0.5 \leq \delta_1$
untimed (ideal) trace: $e_1$, init
constraint: $0.5 \leq \delta_1$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$, init, msg
constraint: $0.5 \leq \delta_1 \land \exists x . x = 0.75 \leq \delta_1$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1, \text{init, msg}$

constraint: $0.75 \leq \delta_1$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: \( e_1, \text{init}, \text{msg}, e_2 \)
constraint: \( 0.75 \leq \delta_1 \land \exists y_3 . y_3 = \delta_1 - 0.75 \leq 0.5 \)
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$, init, msg, $e_2$

constraint: $0.75 \leq \delta_1 \land \delta_1 \leq 1.25$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: \( e_1, \text{init}, \text{msg}, e_2, \text{off} \)

constraint: \( 0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25 \wedge \\
\exists x, y_2 . x = 1.25 - \delta_1 \leq \delta_2 \wedge y_2 = 1.25 - \delta_1 \leq 0.5 \)
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1, \text{init}, \text{msg}, e_2, \text{off}$

constraint: $0.75 \leq \delta_1 \land \delta_1 \leq 1.25 \land 1.25 - \delta_2 \leq \delta_1$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$, init, msg, $e_2$, off, on

constraint: $0.75 \leq \delta_1 \land \delta_1 \leq 1.25 \land 1.25 - \delta_2 \leq \delta_1 \land \exists x \ . \ x = 0.5 \leq \delta_2$
Depth First Symbolic State Exploration (example 2)

untimed (ideal) trace: $e_1$, init, msg, $e_2$, off, on

constraint: $0.75 \leq \delta_1 \land \delta_1 \leq 1.25 \land 0.5 \leq \delta_2$
3.2 Inverse Method

[André phd 2010, Soulat phd 2012, Fribourg]
Given a reference valuation $\pi_0$ of the parameters, synthesize a constraint $K_0$ on the parameters guaranteeing the same untimed traces as with $\pi_0$.

start with $K := true$
computes iteratively a set $S$ of reachable states of $A(K)$
when $\langle \ell, C \rangle \in S$ and $\pi_0 \not\models \exists x \ C(x)$
then $K := K \land \neg J$ for chosen $\pi_0$-incompatible conjunct $J$.

tool IMITATOR: fixpoint after 6 iterations on example
returns $K = 0.75 < \delta_1 \leq 1.0 \land 1.25 < \delta_1 + \delta_2$. 
Conclusion

▶ compilation of Antescofo scores into a formal model.
▶ static analysis to characterize the range of performances for which the reactive engine of Antescofo will behave as expected.
▶ outcome of analysis:
  linear constraints on performance delays
    ▶ warn performers about dangerous parts in the score
    ▶ assistance for adjusting delays and structuring groups
    ▶ user friendly presentation on score?

not seen / ongoing / perspectives
▶ Dealing with error-handling instructions (missed events)
▶ Non-linear scores
▶ Game theoretic approach for a quantitative estimation of robustness

Thank you!
Perspectives: Verification

- delays = linear expressions on parameters (variables in scores)
- complexity wrt restrictions
- reachability analysis or timed model checking over the unparametric TA model (verification of properties of idealized performance)
- timed games: controller synthesis, dynamic scheduling open scores (straightforward extension)
Parametric Timed Automata for Circuits

[Alur, Henzinger, Vardi 1993] parameters $\delta$ for (unknown) time bounds in guards and invariants

We consider a bi-bounded inertial model for gates (see [BS95, MP95]), where any change of the input may lead to a change of the output (after some delay). For a gate with $n$ input signals, the delay associated with the change of the output signal of the gate depends on the configuration (low or high) of the $n$ input signals, thus leading to $2^n$ different delays. Modeling those $n$ possibilities would dramatically increase the complexity of the model. As a consequence, we usually use intervals of delays. Often, we consider only one interval for a gate, i.e., the minimum and the maximum of the possible delays associated with the $2^n$ input configurations. This option is usually a good compromise between precision of the result, and size of the model.

Another classical option is to consider two intervals, one corresponding to the rise of the output signal, the other corresponding to the fall of the output signal. This representation is interesting because delays corresponding to rise and fall can be very different. Although the size of the model is bigger than in the model with one interval of delays, this usually leads to a more precise state space containing less false executions, i.e., executions involving constraints on clocks and parameters which are actually not satisfiable.

Example 2.40. Consider the “NOT” gate depicted in Figure 2.11. This “NOT” gate has one input signal $a$, and one output signal $b$.

[André 2010]

Recall that, in stable mode, the output is equal to the inversion of the input.