Hybrid Systems: Theoretical Contributions
Part II

Edited and presented by
Tom Henzinger
More Research Samples

- Henzinger’s group:
  - Robust Hybrid Systems
  - Timed Games
  - Compositional Real-Time Systems
  - Interface Theories

- Sangiovanni’s group:
  - Petri Net Scheduling
A Quantitative Theory of Timed and Hybrid Systems

• Models are approximations
  - Sensor errors, estimations, uncertainties
  - Need theories that are robust w.r.t. small perturbations

• How close are two models?
  - Traditional: B may or may not match (or refine) A
  - Quantitative: B may match A “better” than B’ does

• Quantitative (bi)simulation relations:
  - H, Majumdar, Prabhu 2005
A Quantitative Theory of Timed and Hybrid Systems

1. Quantitative models:
   - What is the distance between two models?

2. Continuity of specifications:
   - Close models satisfy close specifications

3. Quantitative specifications:
   - View formulae as real-valued functions on states
A Quantitative Theory of Timed and Hybrid Systems

1. Distance between traces $d(t, t')$:
   - $\sup$ of timing mismatches

   \[
   a \quad a \\
   \begin{array}{ccc}
   1.1 & 1.9 & 3.2 \\
   \end{array}
   \quad t
   \]

   \[
   a \quad a \\
   \begin{array}{ccc}
   1.2 & 1.6 & 3.0 \\
   \end{array}
   \quad t'
   \]

   $d(t, t') = 0.3$

2. Trace distance between states $D(s, s')$:
   - $\sup_{t, t'} \inf_{L(s)} L(s') d(t, t')$
   - Game interpretation: adversary chooses trace from $s'$, and we try to match it as well as possible from $s$
A Quantitative Theory of Timed and Hybrid Systems

- **Traditional theory:**
  - $s$ refines $s'$ iff $L(s) \subseteq L(s')$
  - Efficient sufficient condition: $s$ simulated by $s'$

- **Quantitative theory:**
  - $D(s,s')$ not computable
  - Computable upper bound: $SD(s,s',0)$ where
    $SD(s,s',\delta) = \sup_\varepsilon \inf_{\varepsilon^*} \{ \max(\delta, SD(r,r',\delta + |\varepsilon - \varepsilon^*|)) : s \not\rightarrow_{\varepsilon} r, s' \not\rightarrow_{\varepsilon^*} r' \}$
A Quantitative Theory of Timed and Hybrid Systems

Continuity theorem for TCTL:

If $SD(s,s',0) \cdot \varepsilon$ and $s \rightarrow \phi$, then $s' \rightarrow \text{relax}(\phi, 2\varepsilon)$.

Example:

$\phi$: $9 \leq 5 \ p$
$\text{relax}(\phi, \varepsilon)$: $9 \leq 5 + \varepsilon \ p$

So, if we want a model to satisfy $9 \leq 5 \ p$ and the modeling error is estimated at most $\varepsilon$, then we should model check $9 \leq 5 - 2\varepsilon \ p$. 
A Quantitative Theory of Timed and Hybrid Systems

- **CTL:**
  
  \[
  [\emptyset \mathsf{p}](s) = \begin{cases} 
  0 & \text{if } \mathsf{p} \text{ can be avoided forever from } s \\
  1 & \text{otherwise} 
  \end{cases}
  \]

- **QCTL:**
  
  \[
  [\emptyset \mathsf{p}](s) = \beta^t \text{ where } t \text{ is the longest time that can be spent avoiding } \mathsf{p} \text{ from } s
  \]
  
  \[0 < \beta < 1 \quad \ldots \text{ discount factor}\]
A Quantitative Theory of Timed and Hybrid Systems

\[
[p](s) = \begin{cases} 
1 & \text{if } s \leq p \\
0 & \text{if } not s \leq p
\end{cases}
\]

\[
[\phi](s) = 1 - [\phi](s)
\]

\[
[\phi_1 \lor \phi_2](s) = \max([\phi_1](s), [\phi_2](s))
\]

\[
[\phi_1 \land \phi_2](s) = \min([\phi_1](s), [\phi_2](s))
\]

\[
[9] \phi](s) = \sup_{t \in L(s)} \sup_{\delta} \beta^\delta \phi [\phi](t @ \delta)
\]

\[
[8] \phi](s) = \inf_{t \in L(s)} \sup_{\delta} \beta^\delta \phi [\phi](t @ \delta)
\]

\[
[9] \phi](s) = \sup_{t \in L(s)} \inf_{\delta} \beta^\delta \phi [\phi](t @ \delta)
\]

\[
[8] \phi](s) = \inf_{t \in L(s)} \inf_{\delta} \beta^\delta \phi [\phi](t @ \delta)
\]

We have been able to show only the computability of a subset of QCTL over timed automata; the general model checking question remains open.

"Hybrid Systems Theory: II", T. Henzinger

ITR Review, Oct. 4, 2006

9
Quantitative Continuity Theorem:

Let $k$ be the number of nested temporal operators in $\phi$. If $\Delta(s,s',0) \cdot \varepsilon$, then $|\phi(s) - \phi(s')| \cdot (k+1) \cdot (1 - \beta^2 \varepsilon)$. This bounds the specification error in terms of the model error.
Quasi-Static Scheduling

- Petri nets have been successfully used in quasi-static scheduling of concurrent programs.

[ Liu, Sangiovanni-Vincentelli, Watanabe, Kondryatev ]
Motivational Example

- Petri nets generated from interesting applications are often unschedulable.

```
while(1){
    N=read(IN);
    write(X, N);
    for(i=0;i<N;i++){
        write(Y, D[i]);
    }
}
```

```
while(1){
    M=read(X);
    for(j=0;j<M;j++){
        E[j]=read(Y);
    }
}
```
Our Approach

• Question:
  Is a given Petri net schedulable?
  Is a given Petri net not schedulable?

• One Solution: Try to construct a schedule (very time consuming)

• Our approach: Employ necessary conditions for schedulability which are based on the Petri net structure and hence efficient to decide.
  - Checking cyclic dependence of transitions using linear programming
  - Checking a rank condition of the incidence matrix using linear algebra
Experiments

- Experiments from real applications show effectiveness and efficiency of our approach.
  - PVRG-JPEG encoder [Hung 93]
  - Motion-JPEG encoder [Lieverse 01]
  - Philips MPEG2 decoder [Wolf 99]
  - XviD MPEG4 encoder [Broekhof 04]

<table>
<thead>
<tr>
<th></th>
<th>#P</th>
<th>#T</th>
<th>#Arc</th>
<th>#FC S</th>
<th>Rank</th>
<th>CDC</th>
<th>Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEGenc1</td>
<td>26</td>
<td>27</td>
<td>64</td>
<td>6</td>
<td>&lt;0.01s</td>
<td>0.19s</td>
<td>&gt;24hr</td>
</tr>
<tr>
<td>MJPEGenc</td>
<td>117</td>
<td>124</td>
<td>330</td>
<td>25</td>
<td>&lt;0.01s</td>
<td>0.04s</td>
<td>&gt;24hr</td>
</tr>
<tr>
<td>MPEG2dec1</td>
<td>116</td>
<td>144</td>
<td>358</td>
<td>38</td>
<td>&lt;0.01s</td>
<td>0.25s</td>
<td>&gt;24hr</td>
</tr>
<tr>
<td>MPEG4dec</td>
<td>72</td>
<td>72</td>
<td>184</td>
<td>15</td>
<td>&lt;0.01s</td>
<td>0.16s</td>
<td>&gt;24hr</td>
</tr>
</tbody>
</table>