Modelling and Verification of Hybrid Systems

Dimitar P. Guelev, Zhan Naijun and Zhou Chaochen

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
Institute of Software, Chinese Academy of Sciences

August 24, 2012
Sections

• Background
• Hybrid CSP
• Differential Invariant and its Generation
• Hybrid Hoare Logic
• Verifying Programs through Symbolic Computation
• Discussion
Hybrid Systems

- Hybrid System involves Discrete Behaviour of Control Software and Continuous Behaviour of Controlled Physical Devices (E.g. Communication-based Control Systems, Embedded Systems, ...)

- Examples:
  High Speed Train Control Systems (ETCS, CTCS), Air Traffic Control Systems, Nuclear Reactor Control Systems, etc
Features:
- Mixture of Discrete and Continuous Behaviour
- Safety Criticality
- Interdisciplinary

Design and Verification of Safety Critical Hybrid Systems become a big Challenge for Computer Science, Control Theory and Mathematics!
Automaton-based Modelling and Verification of Hybrid Systems

- Hybrid automata [Alur et al, 1995], ...
  Finite State Automaton plus Differential Equations
- Model Checking Tool
- Advantage: Intuitive and Fully Automatic
- Disadvantage: ...
Logical Approach to Modelling and Verification of Hybrid Systems

- Hybrid Program and DADL (Differential and Algebraic Dynamic Logic [Platzer&Clarke 2008])
- Hybrid CSP
Hybrid CSP (1)

Notation

- Hybrid CSP (Subset)
  - CSP + Differential Equations + Interrupts

\[
P ::= \text{skip} \mid v := e \mid ch?x \mid ch!e \mid \langle F(\dot{s}, s) = 0 \& B \rangle \mid P ; Q \mid B \rightarrow P \mid \langle F(\dot{s}, s) = 0 \& B \rangle \geq_d P \mid \text{wait } d \mid \langle F(\dot{s}, s) = 0 \& B \rangle \geq \bigcirc_{i \in I}(io_i \rightarrow P_i) \mid \bigcirc_{i \in I}(io_i \rightarrow P_i) \]

\[
S ::= P \mid P^* \mid S \parallel S
\]

- Synchronous Communication
Hybrid CSP（2）

- \langle F(\dot{s}, s) = 0&B \rangle$ defines an evolution by a differential equation over $s$. $B$ is a first order formula of $s$, which defines a domain of $s$ in the sense that, if the evolution of $s$ is beyond $B$, the statement terminates. Otherwise it goes forward.

- $B \rightarrow P$ behaves like $P$ if $B$ is true. Otherwise it terminates.

- $\langle F(\dot{s}, s) = 0&B \rangle \geq_d P$ behaves like $\langle F(\dot{s}, s) = 0&B \rangle$ if it can terminate within $d$ time units. Otherwise, after $d$ (inclusive) time units, it will behave like $P$. 
Hybrid CSP（3）

- **wait** $d$ does nothing and terminates after $d$ time units.
- $\langle F(s, s) = 0&B \rangle \triangleright \Box_{i\in I}(io_i \to P_i)$ behaves like $\langle F(s, s) = 0&B \rangle$ until a communication in the following context appears. Then it behaves like $P_i$ after communication $io_i$ effects.
- $\Box_{i\in I}(io_i \to P_i)$ is the *external choice* of CSP.
- $P^*$ means the execution of $P$ can be repeated any finite times.
Hybrid CSP (4)

- $S_1 \parallel S_2$ behaves as if $S_1$ and $S_2$ are executed independently except that all communications along the common channels between $S_1$ and $S_2$ are to be synchronized. Let

$$(VC(S_1) \cap VC(S_2)) = \emptyset$$

$$(InChan(S_1) \cap InChan(S_2)) = \emptyset$$

$$(OutChan(S_1) \cap OutChan(S_2)) = \emptyset$$

where $VC(S)$ is the variables (including Continuous variables) of $S$, $InChan(S)$ and $OutChan(S)$ are Input and Output Channels of $S$. 
Hybrid CSP (5)

- Example: Plant Control (PLC)

\[ (\langle F(s, \dot{s}, u) = 0 \rangle \triangleright (\text{sensor}!s \rightarrow \text{actuator}?u))^* \parallel (\text{wait } d; \text{sensor}?v; \text{actuator}!\text{contl}(v))^* \]

- Super-dense Computation (Assignment, Message Passing, ... do not consume time, or consume \textbf{Negligible} time)
Verification of Hybrid CSP Processes

- How to treat Differential Equations
- Extend Inductive Assertion Method
  - Maintain the Frame of Program Logic
  - Introduce **Differential Invariants**: First Order Assertions (Properties) of Continuous Variables satisfied by Solutions of Differential Equations for Given Initial Values.
Differential Invariant and its Generation（1）

- Related work:
  - Groebner basis [Manna et al 2004]
  - Barrier certificate [Pjajna&Jadbabdaie 2004, Platzer&Clarke 2008]
  - Tangent cone and Lie derivative [Ankur&Tiwari 2009]
  - Boundary method [Ankur,Gulwani&Tiwari 2009]
  - Ideal fixed point method [Sankaranarayanan 2010]
- Design a Complete Algorithm to generate Polynomial Differential Invariant for Polynomial Differential Equation [Sankaranarayanan 2010, Ankur&Tiwari 2009].
An Algorithm for Polynomial DI

- Given

\[ \dot{x_1} = f_1, \dot{x_2} = f_2, \ldots, \dot{x_n} = f_n \]

where \( f : (f_1, f_2, \ldots, f_n) \) is an array of polynomials.
An Algorithm for Polynomial DI (Cont.)

- To check whether $p \geq 0$ is a DI of the above differential equation, where $p$ is a polynomial of $x : (x_1, x_2, ..., x_n)$, apart from the initial state satisfying $p \geq 0$, we only need to check whether the solution $x(t) : (x_1(t), x_2(t), ..., x_n(t))$ will reach $p < 0$ from $p = 0$.

- That is to check, when $t = 0$ and

$$p(x(0)) = 0$$

whether there exists $\delta > 0$, such that $p(x(t)) \geq 0$ is true for $t \in (0, \delta)$. 
An Algorithm for Polynomial DI (Cont.)

- Given the differential equations, the Taylor Expansion of $p$ at $t = 0$ is

$$p(x(t)) = L_f^0 p(x(0)) + L_f^1 p(x(0)) t + L_f^2 p(x(0)) t^2 / 2! + ...$$

where $L_f^0 p(x(0)) = p(x(0)) = 0$, and $L_f^{i+1} p(x(0))$ is $p$'s $(i + 1)$ Lie derivative at $t = 0$. Namely

$$L_f^{i+1} p(x) = \sum_{j=1}^{n} \frac{\partial L_f^i p(x)}{\partial x_j} \cdot f_j(x)$$
An Algorithm for Polynomial DI (Cont.)

- By the Taylor expansion, if $L_f^0 p(x(0)) = 0$ and $L_f^1 p(x(0)) > 0$, then there is a $\delta > 0$, such that

  \[ \forall t. \delta > t > 0 \implies p(x(t)) > 0 \]

- If $L_f^0 p(x(0)) = 0$ and $L_f^1 p(x(0)) < 0$, then $p \geq 0$ is not a DI.

- If $L_f^0 p(x(0)) = 0$ and $L_f^1 p(x(0)) = 0$, then check $L_f^2 p(x(0))$.

- From Theorem for Ascending Chain of Ideals we can guarantee the termination of the procedure.
An Algorithm for Polynomial DI (Cont.)

- Theorem for Ascending Chain of Ideals: For any ascending chain of polynomial ideals

\[ I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots \]

there exists an \( N \geq 0 \) such that, for all \( i > N \), \( I_i = I_N \).
An Algorithm for Polynomial DI (Cont.)

- Ideal Generator: \((g_1, ..., g_n)\) is called a Generator for the following ideal

\[
\left\{ \sum_{i=1}^{n} g_i \cdot h_i \mid h_i \text{ any polynomial} \right\}
\]

- Let \(I_i\) be the ideal generated by \((L_f^0 p, ..., L_f^i p)\). Therefore \(I_i \subseteq I_{i+1}\), and

\[
I_0 \subseteq I_1 \subseteq I_2 \subseteq ...
\]

compose an ascending chain of ideals. It can be proved that if \(I_{N+1} = I_N\), then for any \(i > N\) we have \(I_i = I_N\).
An Algorithm for Polynomial DI (Cont.)

- list the constraint

\[
\forall x. \, p(x) = 0 \Rightarrow L^1_f p(x) > 0
\]
\[
\lor \quad L^1_f p(x) = 0 \land L^2_f p(x) > 0
\]
\[
\ldots
\]
\[
\lor \quad L^1_f p(x) = L^2_f p(x) = \ldots = L^N_f p(x) = 0
\]

and apply the Tarski’s (Collins’) Algorithm.
An Algorithm for Polynomial DI (Cont.)
To check whether $p > 0$ is a DI of

$$\dot{x}_1 = f_1, \quad \dot{x}_2 = f_2, \ldots, \quad \dot{x}_n = f_n$$

- Equivalent to check whether $p = 0$ can be reached from $p > 0$. 
An Algorithm for Polynomial DI (Cont.)

- By Taylor expansion, if \( p(x(0)) = 0 \) and \( L^1_f p(x(0)) > 0 \), there is a \( \delta > 0 \)

\[
\forall t. \quad -\delta < t < 0 \Rightarrow p(x(t)) < 0
\]

That is, \( p = 0 \) can only be reached from \( p < 0 \). Hence, \( p > 0 \) is a DI.

- If \( p(x(0)) = 0 \) and \( L^1_f p(x(0)) < 0 \), then \( p > 0 \) can reach \( p = 0 \). \( p > 0 \) is not a DI.

- ...
Differential Invariant and its Generation (11)

Summary

- Guess a Template $p$,
- Compute the Constraints through Symbolic Computation Tool,
- and Obtain the Solutions for the parameters in $p$. 
Lyapunov Function Generation and Switching Controller Synthesis

- A similar technique can be used to Generate Lyapunov Function and Synthesize Optimal Controller for Hybrid Systems.
  
  1. Jiang Liu, Naijun Zhan and Hengjun Zhao: Automatically Discovering Relaxed Lyapunov Functions for Polynomial Dynamical Systems, Mathematics in Computer Science (to appear)
  
Hybrid Hoare Logic


Hoare Triple

- \{PreH\} P \{PostH\}
  where \textit{PreH} (and \textit{PostH}) is a Duration Calculus (with Infinite interval and Iteration (Kleene Star), LNCS 965, 1180, TCS 337) formula to record Pre-History (and Post-History) of \( P \).
Brief Introduction to Duration Calculus

- DC is based on (Continuous Time) Interval Temporal Logic.
- State $S : \text{Real} \rightarrow \{0, 1\}$ (Boolean Operators, e.g. $\neg S$ etc)
- Given an Interval $[b, e]$, $\int S = c$ if $\int_b^e S(t)dt = c$
- We have the followings

\[
\ell = \int 1 \\
[S] = (\int S = \ell) \land (0 < \ell < \infty) \\
[S]^\omega = (\ell = \infty) \land \Box((0 < \ell < \infty) \Rightarrow [S]) \\
[S]^< = [S] \lor (\ell = 0) \\
[S]^\dagger = [S]^< \lor [S]^\omega \\
(F \land (\ell = \infty)) \sim G \iff F \land (\ell = \infty)
\]
Hybrid Hoare Logic (3)

- Example: Plant Control (PLC)

\[
\langle F(s, \dot{s}, u) = 0 \rangle \triangleright (\text{sensor}!s \rightarrow \text{actuator}?u)^* \parallel \\
(\text{wait } d; \text{sensor}??v; \text{actuator}!\text{contl}(v))^*
\]

- Stability of PLC:

\[
\{[\text{Controlable}(s, u)] \} \text{ PLC } \{(\ell \leq T) \lor ((\ell = T) \overline{\exists}| s - s_{\text{target}} |< \varepsilon)\}\}
\]

- After introducing **Negligible** time (State \(N\))

\[
\{[\text{Controlable}(s, u) \land N]\} \text{ PLC } \\
\{(\int \neg N \leq T) \lor ((\int \neg N = T) \overline{\exists}| s - s_{\text{target}} |< \varepsilon)\}\}
Axioms and Rules
In order to establish a Compositional Calculus, we introduce for each channel $c$ two shared States $c!$ and $c?$ to mean the Ready states of Output and Input plus a shared Variable $c$ to hold the message to be passed.

- **Monotonicity**
  If $\{\text{Pre}H\} \; P \; \{\text{Post}H\}$, $\text{Pre}H' \Rightarrow \text{Pre}H$
  and $\text{Post}H \Rightarrow \text{Post}H'$
  then
  $$\{\text{Pre}H'\} \; P \; \{\text{Post}H'\}$$

- **Skip**
  $$\{\text{Pre}H\} \; \text{skip} \; \{\text{Pre}H\}$$
Hybrid Hoare Logic (5)

Axioms and Rules

- **Assignment**
  
  If $\text{PreH} \Rightarrow (\ell < \infty) \land [\text{Pre} \, [e/x]]$, then

  $$
  \{\text{PreH}\} \ x := e \ {\{\text{PreH} \land \neg \text{Chan}(P) \land N\}}
  $$

  where we assume that $\text{Pre}$ does not contain $N$ nor Channel Variables,

  $\text{Chan}(P) = (\{c? \mid c \in \text{InChan}(P)\} \cup \{c! \mid c \in \text{OutChan}(P)\})$, and, by $\neg \text{Chan}(P)$, we mean this assignment statement is inside process $P$ and $\neg$ applies to each member of $\text{Chan}(P)$. The followings will follow the same assumption. This rule also shows that Assignment takes Negligible time.
Hybrid Hoare Logic (6)

Axioms and Rules

• **Sequential Composition**
  
  If \( \{PreH_i\} \ P_i \ \{PostH_i\} \ (i = 1, 2) \) and \( PostH_1 \Rightarrow PreH_2 \), then

  \[
  \{PreH_1\} \ P_1; P_2 \ {PostH_2}\]

• **Wait**

  If \( PreH \Rightarrow (\ell < \infty) \overline{[Pre]} \), then

  \[
  \{PreH\} \ \text{wait} \ d \ \{PreH \overline{([Pre \land \neg Chan(P)] \land (\int \neg N = d))}\}
  
  where \( d > 0 \)
Axioms and Rules

- **Boundary Interruption**
  Given a differential invariant $Inv$ of $\langle F(\dot{s}, s) = 0 & B \rangle$ with Initial States satisfying $Init$

  If $PreH \Rightarrow (\ell < \infty) \neg [Init \land Pre]$, then
  \[
  \{PreH\} \langle F(\dot{s}, s) = 0 & B \rangle
  \{PreH \neg [Inv \land Pre \land B \land \neg Chan(P)]^+\}
  \[Pre \land \text{Close}(Inv) \land \text{Close}(\neg B) \land N \land \neg Chan(P)\}
  
  where $Pre$ does not contain $s$, and $\text{Close}(G)$ stands for the closure of $G$. 

Timeout Interruption

\[ \langle F(\dot{s}, s) = 0 \& B \rangle \models_d Q \]

can be semantically defined as

\[ \langle F(\dot{s}, s) = 0, \dot{t} = 1 \& (B \land t < d) \rangle; ((t = d) \rightarrow Q) \]

with 0 as initial value of \( t \).

For the differential equation with \( t \), if we can generate a Differential Invariant which can deduce a range of \( t \), say \( Rg(t) \), then we can make sure that the duration of \( \int \neg N \) for \( [ Inv \land Pre \land B \land (t < d) \land \neg \text{Chan}(P) ] \) in the Boundary Interruption Rule must satisfy \( Rg(\int \neg N) \).
Axioms and Rules

- **Conditional**
  - If \((\text{Pre}H \Rightarrow (\ell < \infty) \lnot [B])\), then
    \[
    \{\text{Pre}H\} B \rightarrow P \{\text{Post}H\}
    \]
    provided \(\{\text{Pre}H\} P \{\text{Post}H\}\).
  - If \((\text{Pre}H \Rightarrow (\ell < \infty) \lnot [\neg B])\), then
    \[
    \{\text{Pre}H\} B \rightarrow P \{\text{Pre}H\}
    \]
Hybrid Hoare Logic (10)

Axioms and Rules

- **Output**

  If \( \text{PreH} \Rightarrow (\ell < \infty) \neg [\text{Pre}] \) and \( \text{Pre} \Rightarrow G(e) \),
  then \( \{\text{PreH}\} \ c!e \ \{\text{PreH} \neg [\text{Pre} \land c! \land \neg c? \land \neg (\text{Chan}(P) \setminus c!) \land G(c)] \dagger \neg [\text{Pre} \land c! \land c? \land G(c) \land N]\} \)

- **Input**

  If \( \text{PreH} \Rightarrow (\ell < \infty) \neg [\text{Pre}] \),
  then \( \{\text{PreH}\} \ c?x \ \{\text{PreH} \neg [\text{Pre} \land c? \land \neg c! \land \neg (\text{Chan}(P) \setminus c?)] \dagger \neg [\exists x.\text{Pre} \land c? \land c! \land (x = c) \land N]\} \)
Axioms and Rules

- **External Choice**
  We use $c_1?x_1 \rightarrow P_1 \quad \square \quad c_2?x_2 \rightarrow P_2$ to explain this Rule.
  - Let $PreH$ be the Pre-History, and $(PreH \Rightarrow (\ell < \infty) \neg[Pre])$.
  - Waiting Phase (2nd one):
    $[Pre \land_{i=1}^{2} (c_i? \land \neg c_i!) \land \neg(Chan(P) \setminus \{c_1?, c_2?\})]^{†}$
  - Synchronous Phase (3rd one): for $i = 1, 2$
    $[\exists x_i. Pre \land c_i! \land c_i? \land (x = c_i) \land N]$
  - If $\{PreH \land WaitPhase \land SynPhase_i\} P_i \{PostH_i\} (i = 1, 2)$, then we can conclude

$$\{PreH\} c_1?x_1 \rightarrow P_1 \quad \square \quad c_2?x_2 \rightarrow P_2 \{PostH_1 \lor PostH_2\}$$
Axioms and Rules

- **Communication Interruption**
  The Rule for $\langle F(\dot{s}, s) = 0 \& B \rangle \triangleright \Box_{i \in I}(i o_i \rightarrow P_i)$ is quite similar to the Combination of the Rules for $\langle F(\dot{s}, s) = 0 \& B \rangle$ and $\Box_{i \in I}(i o_i \rightarrow P_i)$
  - During Waiting Phase, each Communication of $I$ is Ready but its Partner is Not
  - During Synchronous Phase, **Close**($\neg B$) must be disjuncted by Readiness of at least One pair of Communications, and we can Randomly choose one of them to pass Message.
Axioms and Rules

- Repetition

\[
\text{If } \{\text{PreH}\} P \{\text{PreH} \land \text{InvH}\} \\
\text{and } \{\text{InvH}\} P \{\text{InvH} \land \text{InvH}\} \\
\text{then } \{\text{PreH}\} P^* \{\text{PreH} \land \text{InvH}^*\}
\]

where \(\text{InvH}\) is an Invariant History of \(P\).
Axioms and Rules

• Parallelism

If \{\lceil Pre_i \land N\rceil\} S_i \{PostH_i\} (i = 1, 2)
and PostH_i \Rightarrow (\ell < \infty) \lceil Post_i \rceil (i = 1, 2)
then \{\lceil \bigwedge_{i=1}^2 Pre_i \land N\rceil\} S_1 \parallel S_2 \{\bigwedge_{i=1}^2 (PostH_i \lceil Post_i \rceil^+)\}\\

From the above rules one can see that a Communication Deadlock process can conclude either \[ [c! \land \neg c? \land \ldots]^{\omega} \]
or a symmetric one.

• To see the full Calculus please refer to the paper (D. Guelev, et al).
Verifying Programs through Symbolic Computation

- An interdisciplinary effort
- Loop termination analysis (decidable cases, ranking function, etc.)
- Loop invariant generation
- Tool: DISCOVERER (Maple XIII) based on A complete (Symbolic) Discriminant System
Discussion

• 14 Scenarios of CTCS-3
• Tools
  7 pages of a CTCS Scenario
• Pictorial Modelling Languages
  Simulink, ...
• Simplify Proof Rules
• Represent our first attack to this area
• How to extend HCSP to Model Dynamical Systems
• Complexity of the Algorithm
  Polygonal Line, Mixture of Symbolic and Numeric
  Computation, Heuristic Methods (Machine Learning), ...
• ...
• ...
Thanks!