Exploiting Structural Dynamism in FHM: Modelling of Ideal Diodes

EUROSIM 2010: Session on Physical Modelling and Control

Henrik Nilsson and George Giorgidze

School of Computer Science
University of Nottingham
The Assumption of Fixed Causality

- Current main-stream non-causal modelling and simulation languages, like Modelica, are *designed* and *implemented* assuming causality remains fixed during simulation:
The Assumption of Fixed Causality

- Current main-stream non-causal modelling and simulation languages, like Modelica, are **designed** and **implemented** assuming causality remains fixed during simulation:
  - Simplifies the language
The Assumption of Fixed Causality

- Current main-stream non-causal modelling and simulation languages, like Modelica, are designed and implemented assuming causality remains fixed during simulation:
  - Simplifies the language
  - Facilitates efficient implementation
The Assumption of Fixed Causality

- Current main-stream non-causal modelling and simulation languages, like Modelica, are designed and implemented assuming causality remains fixed during simulation:
  - Simplifies the language
  - Facilitates efficient implementation

- But this assumption is very limiting for hybrid modelling; even simple systems often cannot be simulated:
The Assumption of Fixed Causality

• Current main-stream non-causal modelling and simulation languages, like Modelica, are *designed* and *implemented* assuming causality remains fixed during simulation:
  - Simplifies the language
  - Facilitates efficient implementation

• But this assumption is very *limiting* for *hybrid modelling*; even simple systems often cannot be simulated:
  - Breaking pendulum
The Assumption of Fixed Causality

• Current main-stream non-causal modelling and simulation languages, like Modelica, are designed and implemented assuming causality remains fixed during simulation:
  - Simplifies the language
  - Facilitates efficient implementation

• But this assumption is very limiting for hybrid modelling; even simple systems often cannot be simulated:
  - Breaking pendulum
  - Ideal diodes in various configurations.
This Talk

• Introduction to *Functional Hybrid Modelling (FHM)*:
This Talk

- Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
This Talk

- Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented *assuming* an evolving system:
This Talk

- Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented *assuming* an evolving system:
    - in particular, causality allowed to change
This Talk

- Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented *assuming* an evolving system:
    - in particular, causality allowed to change
    - even more drastic changes possible.
This Talk

• Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented *assuming* an evolving system:
    • in particular, causality allowed to change
    • even more drastic changes possible.

• Application to modelling with ideal diodes:
This Talk

• Introduction to Functional Hybrid Modelling (FHM):
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented assuming an evolving system:
    • in particular, causality allowed to change
    • even more drastic changes possible.

• Application to modelling with ideal diodes:
  - Half-wave rectifier with in-line inductor
This Talk

• Introduction to *Functional Hybrid Modelling (FHM)*:
  - Novel approach to non-causal, hybrid modelling and simulation.
  - Designed and implemented *assuming* an evolving system:
    • in particular, causality allowed to change
    • even more drastic changes possible.

• Application to modelling with ideal diodes:
  - Half-wave rectifier with in-line inductor
  - Full-wave rectifier
FHM in a Nutshell
A *functional approach* to modelling and simulation of (physical) systems.
FHM in a Nutshell

- A *functional approach* to modelling and simulation of (physical) systems.
- Two-level design:
  - *equational level* for modelling components
  - *functional level* for spatial and temporal composition of components
FHM in a Nutshell

• A *functional approach* to modelling and simulation of (physical) systems.

• Two-level design:
  - *equational level* for modelling components
  - *functional level* for spatial and temporal composition of components

• *Non-causal modelling*: undirected equations.
FHM in a Nutshell

- A *functional approach* to modelling and simulation of (physical) systems.
- Two-level design:
  - *equational level* for modelling components
  - *functional level* for spatial and temporal composition of components
- *Non-causal modelling*: undirected equations.
- *First-class models* (= systems of equations) at the functional level.
FHM in a Nutshell

- A *functional approach* to modelling and simulation of (physical) systems.
- Two-level design:
  - *equational level* for modelling components
  - *functional level* for spatial and temporal composition of components
- *Non-causal modelling*: undirected equations.
- *First-class models* (= systems of equations) at the functional level.
- Equation systems allowed to *evolve* over time.
“Functional” as in *Pure Functional Programming*:

- **Declarative** programming paradigm
- Programs are pure functions: *no side effects*.
- Not just functions on “numbers”: arguments and results may be arbitrary types, including:
  - functions
  - *models* = systems of equations
- Both functions and models are thus *first-class entities*. 
Is FHM very different from current non-causal languages like Modelica?
Different?

Is FHM very different from current non-causal languages like Modelica?

Yes, in some ways, but:
Different?

Is FHM very different from current non-causal languages like Modelica?

Yes, in some ways, but:

- FHM *implementation techniques* could be used in the implementations of existing non-causal languages to improve their support for systems with evolving structure.
Is FHM very different from current non-causal languages like Modelica?

Yes, in some ways, but:

- FHM **implementation techniques** could be used in the implementations of existing non-causal languages to improve their support for systems with evolving structure.
- FHM could be viewed as a **core language**:
  - semantics
  - compilation target
Prototype Hydra Implementation (1)

The current FHM instance is called *Hydra*:
Prototype Hydra Implementation (1)

The current FHM instance is called *Hydra*:

- Embedding in *Haskell*. 
The current FHM instance is called *Hydra*:

- Embedding in *Haskell*.
- Model *transformed* to form suitable for simulation, then *JIT compiled to native code* by an embedded compiler.
Prototype Hydra Implementation (1)

The current FHM instance is called *Hydra*:

- Embedding in *Haskell*.
- Model *transformed* to form suitable for simulation, then *JIT compiled to native code* by an embedded compiler.
- State-of-the art *numerical solvers from SUNDIALS* suite (from LLNL) used for simulation and event detection.
The current FHM instance is called \textit{Hydra}:

- Embedding in \textit{Haskell}.
- Model \textit{transformed} to form suitable for simulation, then \textit{JIT compiled to native code} by an embedded compiler.
- State-of-the art \textit{numerical solvers from SUNDIALS} suite (from LLNL) used for simulation and event detection.
- Transformation and compilation \textit{repeated} when system structure changes at events.
Prototype Hydra Implementation (2)

\[ f(\vec{y}, \frac{d\vec{y}}{dt}, t) = 0 \]

SR a

Embedded Compiler

\[ f_{llvm}(\vec{y}, \frac{d\vec{y}}{dt}, t) = 0 \]

LLVM JIT

Numerical Solver

Event Detector

Simulation Result

Event Handler

Event
Example: A Simple Circuit
Simple Circuit: Causal Model

\[ u_{R_2} = R_2 i_2 \]
\[ u_L = u_{in} - u_{R_2} \]
\[ i_2' = \frac{u_L}{L} \]
\[ u_{R_1} = u_{in} - u_C \]
\[ i_1 = \frac{u_{R_1}}{R_1} \]
\[ u_C' = \frac{i_1}{C} \]
\[ i = i_1 + i_2 \]
Non-causal resistor model:

\[ v_p - v_n = u \]
\[ i_p + i_n = 0 \]
\[ R i_p = u \]

Non-causal inductor model:

\[ v_p - v_n = u \]
\[ i_p + i_n = 0 \]
\[ L i'_p = u \]

Note the commonality: can be factored out as a separate \textit{two pin} component.
A non-causal model of the entire circuit is created by \textit{instantiating} the component models: copy the equations and rename the variables.

The instantiated components are then \textit{composed} by adding connection equations according to Kirchhoff’s laws, e.g.:

\begin{align*}
    v_{R_1,n} &= v_{C,p} \\
    i_{R_1,n} + i_{C,p} &= 0
\end{align*}
\[
twoPin ::= \text{SR}(\text{Pin}, \text{Pin}, \text{Voltage}) \\
twoPin = \text{sigrel} \ (p, n, u) \ 	ext{where} \\
p \cdot v - n \cdot v = u \\
p \cdot i + n \cdot i = 0
\]
Record describing an electrical connection with fields $v$ for voltage and $i$ for current.

\[
twoPin :: SR(Pin, Pin, Voltage)
twoPin = \text{sigrel} (p, n, u) \text{ where}
\begin{align*}
p.v - n.v &= u \\
p.i + n.i &= 0
\end{align*}
\]
Simple Circuit in FHM (1)

Record describing an electrical connection with fields $v$ for voltage and $i$ for current.

\[
twoPin :: SR(Pin, Pin, Voltage)\]

\[
twoPin = \text{sigrel} (p, n, u) \quad \text{where} \]
\[
p.v - n.v = u
\]
\[
p.i + n.i = 0
\]

(Partial) model represented by relation over 5 time-varying entities, i.e. signals.
Simple Circuit in FHM (1)

Record describing an electrical connection with fields $v$ for voltage and $i$ for current.

$$\text{twoPin} :: \text{SR}(\text{Pin}, \text{Pin}, \text{Voltage})$$

$$\text{twoPin} = \text{sigrel}(p, n, u) \text{ where}$$

$$p \cdot v - n \cdot v = u$$

$$p \cdot i + n \cdot i = 0$$

(Note: Somewhat idealised syntax compared with present implementation.)

(Partial) model represented by relation over 5 time-varying entities, i.e. signals.
resistor :: Resistance → SR(Pin, Pin)

resistor \( r = \text{sigrel} (p, n) \) where

\[
twoPin \diamond (p, n, u)
\]

\[
r \cdot p.i = u
\]
**Simple Circuit in FHM (2)**

**Parametrised** model represented by **function** mapping parameters to a model. Note: first class models!

\[
resistor :: \text{Resistance} \rightarrow \text{SR}(\text{Pin}, \text{Pin})
\]

\[
resistor \ r = \text{sigrel}(p, n) \ \text{where}
\]

\[
twoPin \odot (p, n, u)
\]

\[
r \cdot p.i = u
\]
**Simple Circuit in FHM (2)**

- **Parametrised** model represented by function mapping parameters to a model. Note: first class models!

\[
\text{resistor} :: \text{Resistance} \rightarrow \text{SR}(\text{Pin}, \text{Pin})
\]

\[
\text{resistor } r = \text{sigrel}(p, n) \quad \text{where}
\]

\[
\text{twoPin} \diamond (p, n, u)
\]

\[
r \cdot p.i = u
\]

**Signal relation application** allows modular construction of models from component models.
Inductors and capacitors are modelled similarly:

\[
\text{inductor} :: \text{Inductance} \rightarrow SR(Pin, Pin)
\]
\[
\text{inductor } l = \text{sigrel } (p, n) \text{ where }
\]
\[
twoPin \diamond (p, n, u)
\]
\[
l \cdot \text{der}(p.i) = u
\]

\[
\text{capacitor} :: \text{Capacitance} \rightarrow SR(Pin, Pin)
\]
\[
\text{capacitor } c = \text{sigrel } (p, n) \text{ where }
\]
\[
twoPin \diamond (p, n, u)
\]
\[
c \cdot \text{der}(u) = p.i
\]
simpleCircuit :: SR Current

simpleCircuit = sigrel i where

resistor(1000) \diamond (r1p, r1n)
resistor(2200) \diamond (r2p, r2n)
capacitor(0.00047) \diamond (cp, cn)
inductor(0.01) \diamond (lp, ln)
vSourceAC(12) \diamond (acp, acn)
ground \diamond gp

...
Simple Circuit in FHM (4)

\[ i = r_{1p}.i + r_{2p}.i \]
Notes on the Causal Model (1)

- Topology of causal model and modelled system do not agree:

![Causal Model Diagram]

vs.

![Causal Model Diagram]
• Topology of causal model and modelled system do not agree:

vs.

• A small change in the modelled system can lead to large changes in the model.
In non-causal modelling, user need not worry about causality, but the simulator may well exploit structural properties like causality for e.g. efficient simulation.
In non-causal modelling, user need not worry about causality, but the *simulator* may well exploit structural properties like causality for e.g. efficient simulation.

Once-off exploitation of *any* structural properties will preclude significant dynamic structural changes.
Example: Ideal Diodes (1)
Example: Ideal Diodes (2)
The in-line inductor means that an assumption of fixed causality will cause *simulation to fail* with a division by zero when the switch opens.
Example: Ideal Diodes (3)

\[
\text{icDiode :: } \text{SR} (\text{Pin}, \text{Pin}) \\
\text{icDiode = sigrel} (p, n) \text{ where} \\
\text{twoPin} \diamond (p, n, u) \\
\text{initially; when } p.v - n.v > 0 \Rightarrow \\
\hspace{1cm} u = 0 \\
\text{when } p.i < 0 \Rightarrow \\
\hspace{1cm} p.i = 0
\]

(Note: again, syntax somewhat idealised compared with present implementation.)
Example: Ideal Diodes (4)
Example: Ideal Diodes (5)
To simulate the full-wave rectifier:

- The diode model has to be extended to allow expressing the voltage over the diodes always pairwise equal:

\[
\begin{align*}
\text{icDiode} &::= \text{SR}(\text{Pin}, \text{Pin}, \text{Voltage}) \\
\text{icDiode} &= \text{sigrel}(p, n, u) \text{ where} \\
\text{twoPin} &\triangleq (p, n, u) \\
\text{initially; when} &\quad p.v - n.v > 0 \implies u = 0 \\
\text{when} &\quad p.i < 0 \implies p.i = 0
\end{align*}
\]
Example: Ideal Diodes (7)

- Redundant, semantically identical equations needs to be eliminated ("constant propagation" suffice in this case).
- End result is a fairly compositional model.
- No separate formalism, such as state charts, for controlling the switching.
- No need to worry about the here $2^4 = 16$ (and, in general, $2^n$) possible modes: each mode computed on demand.
- No domain-specific assumptions built into the language itself.
Conclusions

- Assuming unchanging structural properties like causality severely limits what hybrid models can be simulated.
- Avoiding this restriction allows a number of challenging systems to be modelled and simulated in a straightforward manner.
- Of course not the whole story; many challenging problems remain: e.g., state transfer between structural configurations, chattering . . .