Software Engineering
for Embedded Systems

Towards the Verification of Refactorings of Hybrid Simulink Models

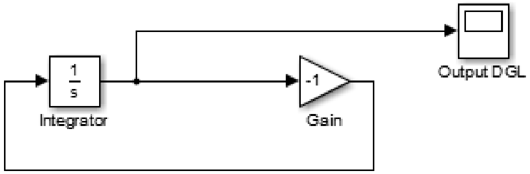
Sebastian Schlesinger, Paula Herber, Thomas Göthel, Sabine Glesner
Software Engineering for Embedded Systems
Technische Universität Berlin

Introduction

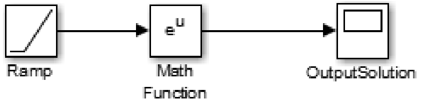
Goal

Automated verification of refactorings of hybrid Simulink models

Example



$$\frac{d}{dt}y(t) = -y(t)$$



$$y(t) = \exp(-t)$$

Criteria

- Automated verification
- Transformation correctness
- Support for hybrid models
- Industrial relevance

Motivation

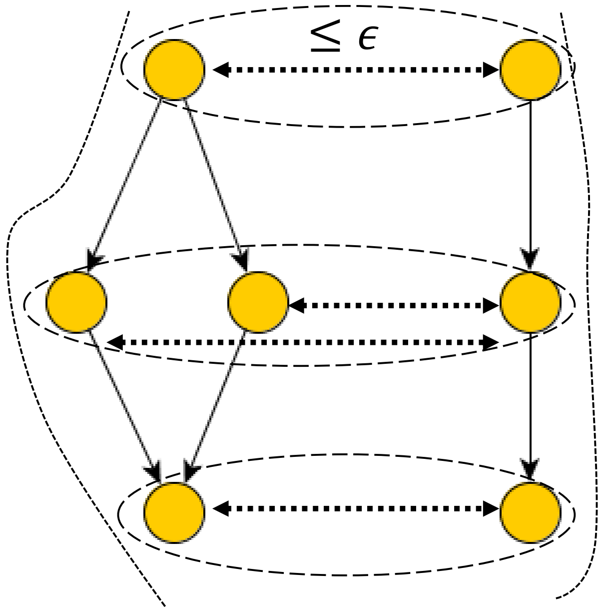
- Simulink de facto standard for Model Driven Engineering in Automotive, Aerospace etc.
- Verification esp. in safety-critical environments
- Refactorings improve structure, preserve behaviour

Related Work

	Simulink semantics	Simulink verification: Discrete Models	Simulink verification: Hybrid Models	Approximate Bisimulation	Simulink refactorings	Simulink reactorings verif.	Remarks
Mathworks documentation	✓						Informal semantics
Bouissou, Chapoutot, ACM SIGPLAN 2012	✓						Formal semantics
Herber, EMSOFT 2013		✓					Transf. to UCLID
Caspi, ACM TECS 2005		✓					Transf. to LUSTRE
Reicherdt, Glesner, ICSE 2014		✓					Transf. to BOOGIE
Agrawal, Simon, Karsai, 2004			✓				Transf. to hybrid automata
Girard, Pappas, European Journal of Control, 2011				✓			
Tran, Wilmes, Dziobek, ICSEA 2013					✓		
Stuermer, Mathworks Automotive 2007					✓		
Our approach aims at	✓	✓	✓	✓	✓	✓	under development

Background (Approximate) Bisimulation

- $LTS \quad T_i = (Q_i, Q_i^0, \rightarrow_i, \Pi, \langle . \rangle_i)$
- $B_\epsilon \subseteq Q_1 \times Q_2$
- B_ϵ *approximate bisimulation of precision* $\epsilon \Leftrightarrow \forall (q_1, q_2) \in B_\epsilon$:



1. $d(\langle q_1 \rangle_1, \langle q_2 \rangle_2) \leq \epsilon$
2. $\exists q_1' : q_1 \rightarrow q_1' \Rightarrow \exists q_2' : q_2 \rightarrow q_2' \wedge (q_1', q_2') \in B_\epsilon$ and vice versa

- $T_1 \sim T_2 \Leftrightarrow \forall q_1 \in Q_1^0 \exists q_2 \in Q_2^0 \exists B_\epsilon \subseteq Q_1 \times Q_2$ *approx. bisimulation relation*: $(q_1, q_2) \in B_\epsilon$

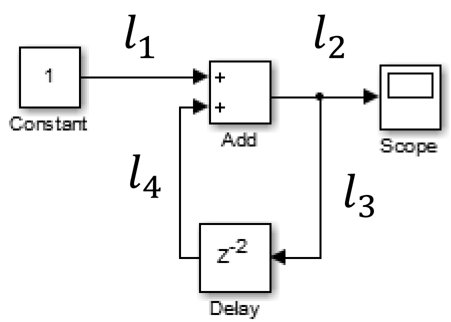
Overview over our approach

1. Abstract Representation (AR):
   Equation set describing how blocks modify signals
2. Proof of soundness of AR with operational semantics
3. Adaptation of approximate bisimulation as more suitable notion of equivalence than traditional bisimulation
4. Epsilon tubes for the precision of the approximate bisimulations

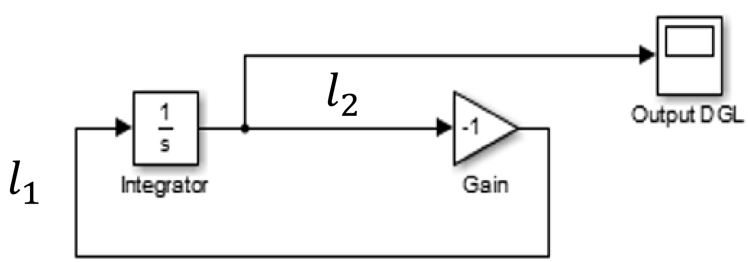
Abstract Representation

- Equation set that describes how the signal is modified at a block by relating input and output

⇒ Sound with semantics; enables abstraction



$$l_1(t) = 1, l_2(t) = l_1(t) + l_4(t),$$
$$l_3(t) = l_2(t), l_4(t + 2h) = l_3(t)$$



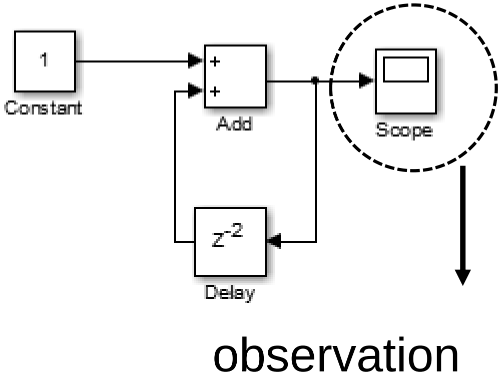
$$l_1(t) = -l_2(t), \frac{d}{dt} l_2(t) = l_1(t)$$

Equation holds for simulation step size $h \rightarrow 0$

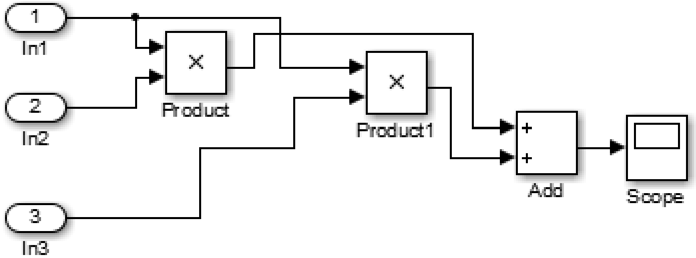
Appr. Bisimulation 1/2

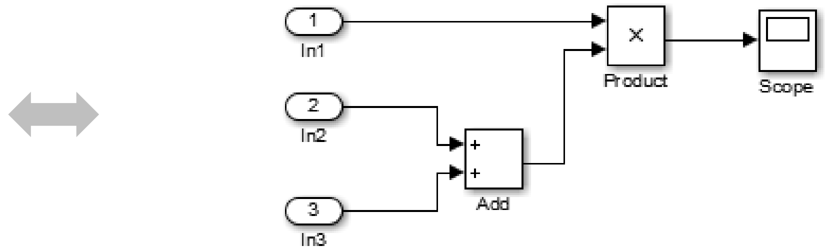
Adaptation of Approximate Bisimulation

- Simulink Model is graph $M = (B, V, I, O)$
- States $Q \subseteq \mathbb{R}^{V}$
- Observations $\Pi \subseteq \mathbb{R}^{\bigcup_{b \in O} var(b)}$
- Metric $d: \Pi \times \Pi \to \mathbb{R}$, $d(\langle\sigma_1\rangle, \langle\sigma_2\rangle) = \left|\left|\langle\sigma_1\rangle - \langle\sigma_2\rangle\right|\right|_\infty$



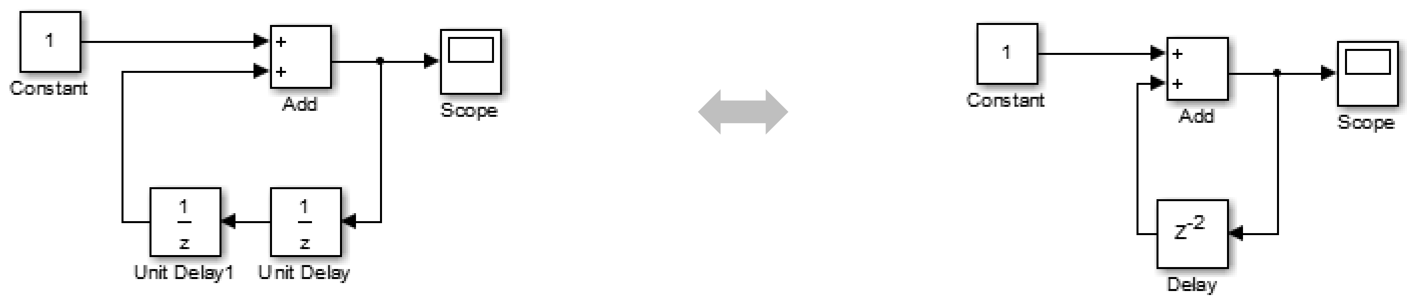
⇒ Unsampled Models: approx. bisimilar with $\epsilon = 0$



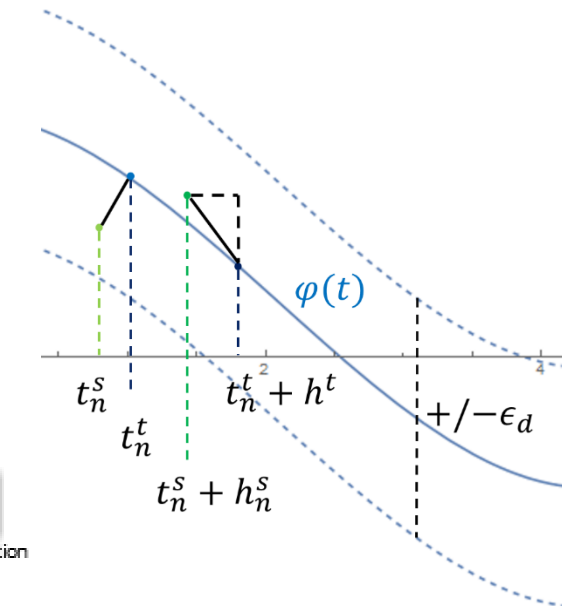


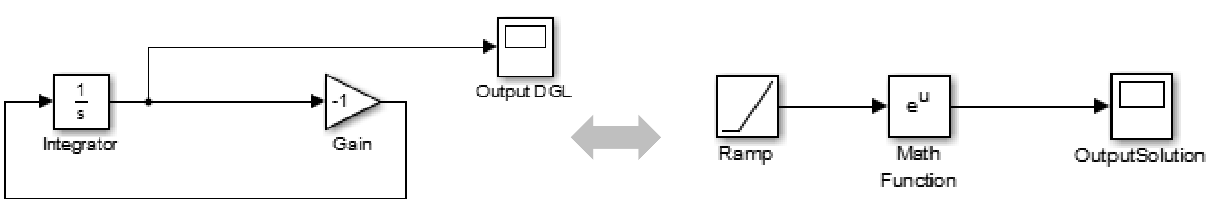
Appr. Bisimulation 2/2 Epsilon Tubes

⇒ Discrete Models: approx. bisimilar with $\epsilon = 0$



⇒ Continuous Models: approx. bisimilar with ϵ depending on second derivative of solution (for Euler technique)





Summary and Future Work

Summary

- Our goal: verification methodology of refactorings for hybrid Simulink models
- Ideas:

1. abstract representation, sound with operational semantics
2. adaptation of approximate bisimulation, allowing observations `close´ to each other

Future Work

- Automation
- Support for hybrid models containing both, discrete and continuous parts
- Enhancement of estimation of epsilon tubes