Dependable Hybrid Systems Design: a Refinement Approach

Zheng Cheng Dominique Méry

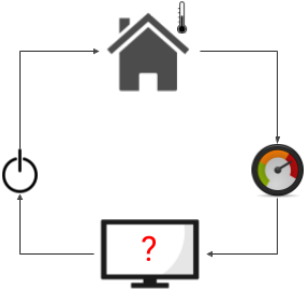
Nov, 2020

Where were we?

- Overview of hybrid system
- Review of calculus
- Review of Event-B
- Develop theories in Event-B

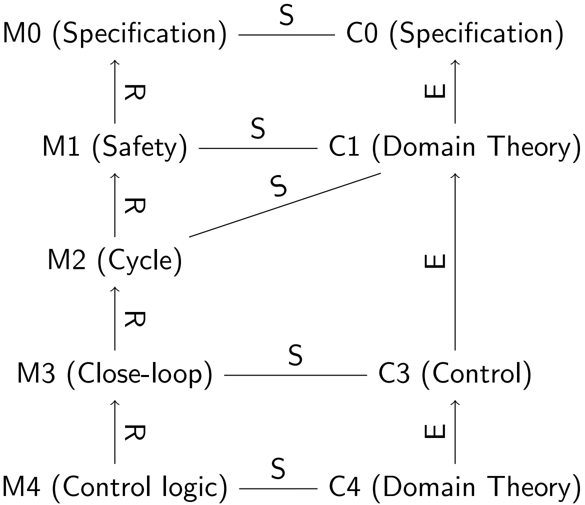
Outlines

Smart Heating System



- 2 modes: ON/OFF
- Simple dynamics: $\dot{T}$=1/-1
- Sample at δ s
- Switch mode costs t_{act} s ($t_{act} < \delta$)
- Safety: $T_{min} \leq$ T $\leq T_{max}$

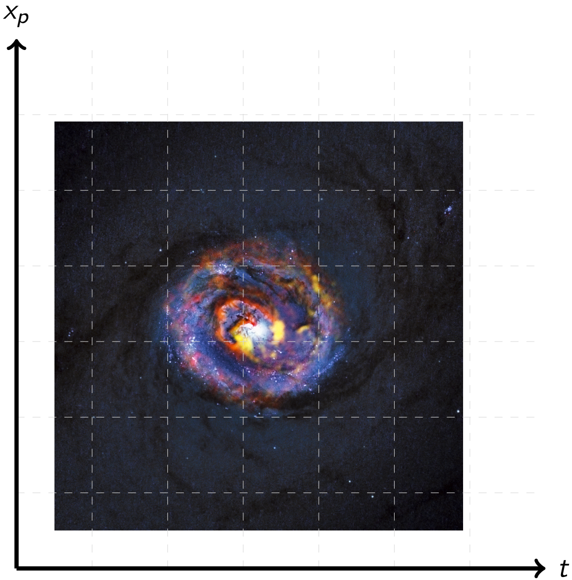
Refinement Strategy for Hybrid System Design



Lab Material

- `https://github.com/veriatl/LORIA_WEEK2`
- Import **theory-axiom-real** to Rodin, and deploy this theory
- Import **ex-heating-maintainer-event** to Rodin

Smart Heating System (Specification M0)

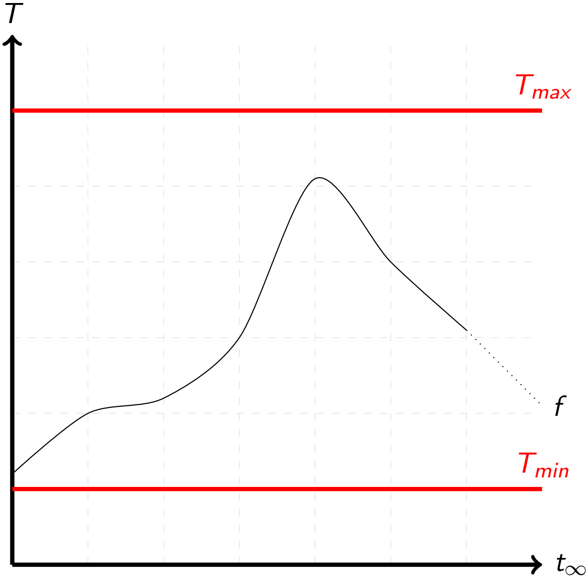


Smart Heating System (Specification M0)

Checklist:

- Generic hybrid system state trajectory
- Generic safety property
- Big-step semantics

Smart Heating System (Safety M1)

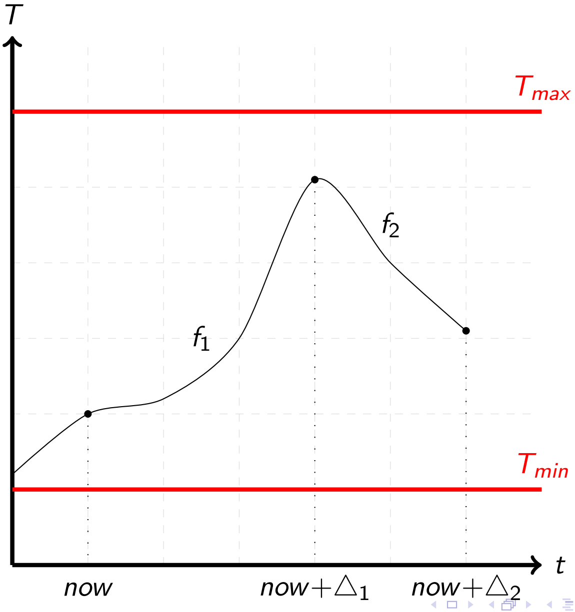


Smart Heating System (Safety M1)

Checklist:

- Concrete system state trajectory
- Concrete safety property
- Big-step semantics refined

Smart Heating System (Cycle M2)



Smart Heating System (Cycle M2)

Checklist:

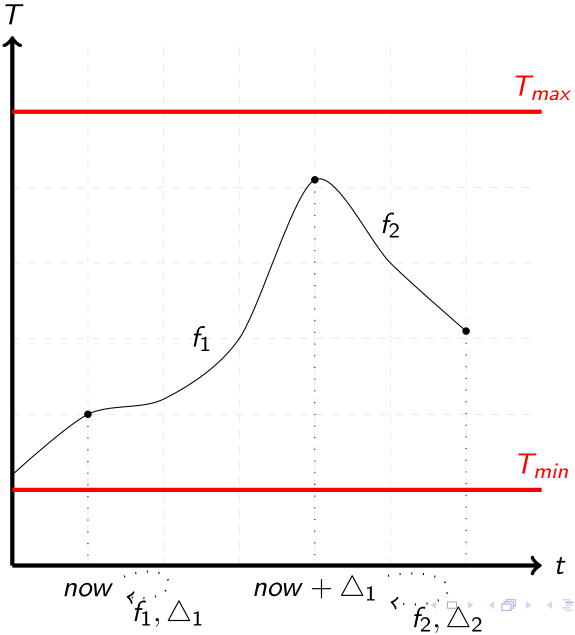
- Time pointer
- Refined system state trajectory
- Refined safety property
- Small-step semantics

Practice

In *M2_cycle*,

1. Encode invariant *safety*: up until *now*, the room temperature is within safe range.
2. Once task 1 is finished, a proof obligation named *Prophecy/safety/INV* will be generated automatically, try to prove this result.

Smart Heating System (Close-loop M3)



Smart Heating System (Close-loop M3)

Checklist:

- Variable for close-loop mode control
- Prediction (Controller)
- Progression (Plant)

Smart Heating System (Control Logic M4)

Event-triggered

Checklist:

- Event-triggered design(when certain events are detected what actions that system should take)
- Specification of time-triggered design

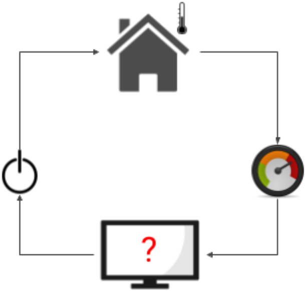
Smart Heating System (Control Logic M4)

Time-triggered

Checklist:

- Revisit the description of heating system
- Time-triggered design(the controller takes action only every once in a while)

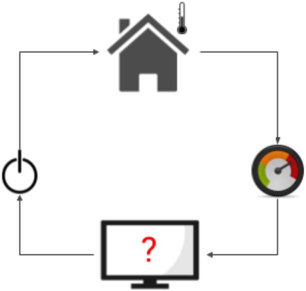
Smart Heating System (Revisit)



- 2 modes: ON/OFF

$\rightarrow$ the only actuation we can do

Smart Heating System (Revisit)



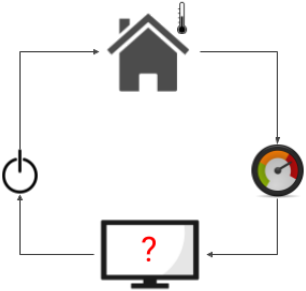
- 2 modes: ON/OFF

$\rightarrow$ the only actuation we can do

- Simple dynamics: $\dot{T}$=1/-1

$\rightarrow$ monotonicity

Smart Heating System (Revisit)



- 2 modes: ON/OFF

$\rightarrow$ the only actuation we can do

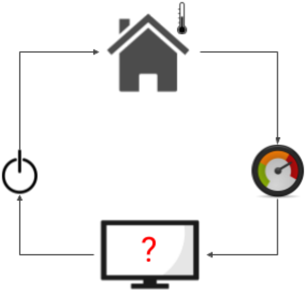
- Simple dynamics: $\dot{T}$=1/-1

$\rightarrow$ monotonicity

- Sample at δ s

$\rightarrow$ Decision at sampling time

Smart Heating System (Revisit)



- 2 modes: ON/OFF

$\rightarrow$ the only actuation we can do

- Simple dynamics: $\dot{T}$=1/-1

$\rightarrow$ monotonicity

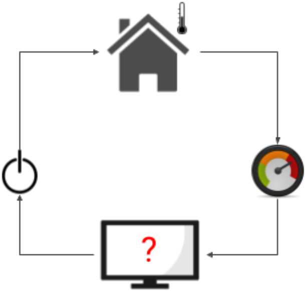
- Sample at δ s

$\rightarrow$ Decision at sampling time

- Switch mode costs t_{act} s ($t_{act} < \delta$)

$\rightarrow$ Cost of switch mode

Smart Heating System (Revisit)



- 2 modes: ON/OFF

→ the only actuation we can do

- Simple dynamics: $\dot{T}$=1/-1

→ monotonicity

- Sample at δ s

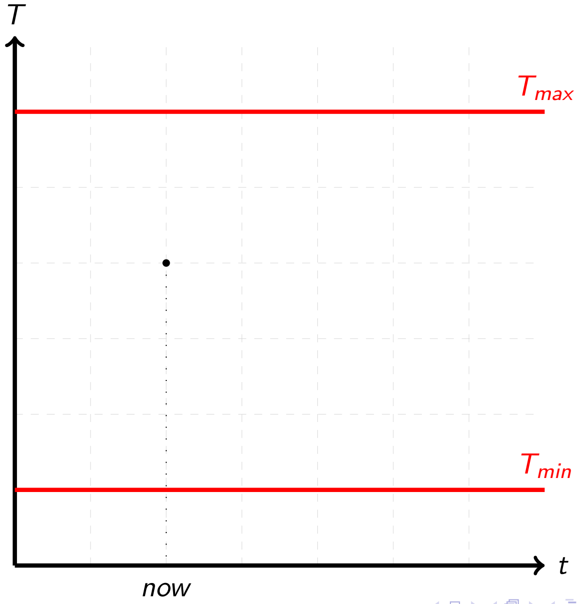
→ Decision at sampling time

- Switch mode costs t_{act} s ($t_{act} < \delta$)

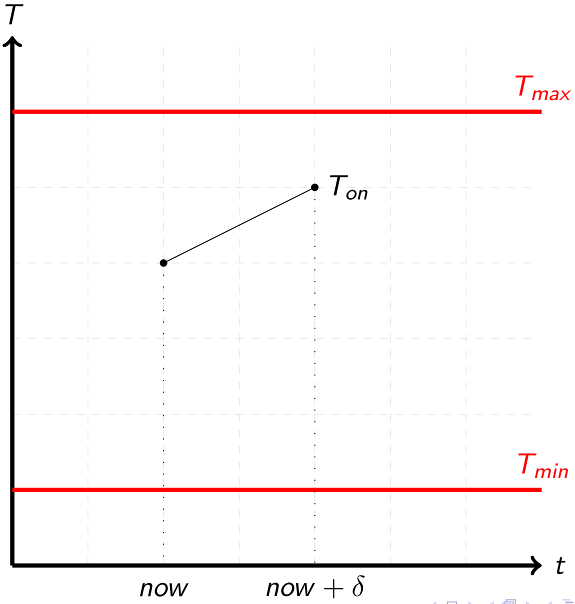
→ Cost of switch mode

- Safety: $T_{min} \leq$ T $\leq T_{max}$

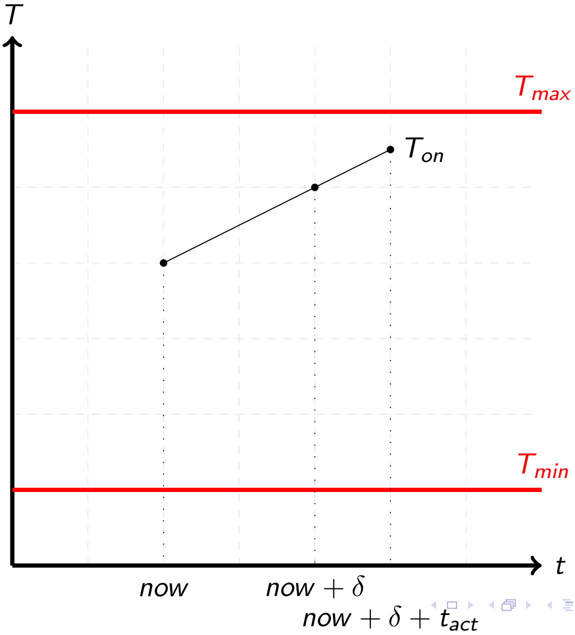
Case 1: ON mode, $T(now) \leq T_{max}$, Stay ON



Case 1: ON mode, $T(now + \delta) \leq T_{max}$, Stay ON



Case 1: ON mode, $T(now + \delta + t_{act}) \leq T_{max}$, Stay ON

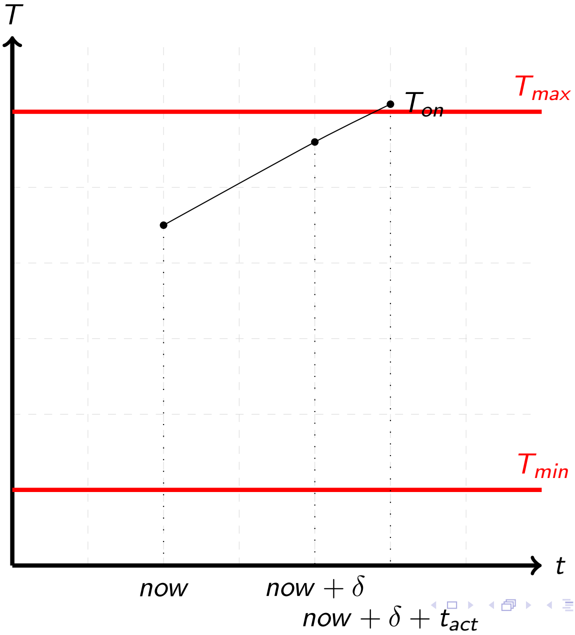


Practice

In *M4_2_control_logic_time_trigger*, *Ctrl_ON_safe* corresponds to case 1.

1. Explain *Ctrl_ON_safe* using natural language.
2. If you think that this control logic is sound, try to prove the proof obligation *Ctrl_ON_safe/grd4/THM*.
3. Otherwise, please modify *Ctrl_ON_safe* to match your expectation, and try to convince Rodin your proposal is sound.

Case 2: ON mode, $T(now + \delta + t_{act}) > T_{max}$, TO OFF



Practice

In $M4\_2\_control\_logic\_time\_trigger$, $Ctrl\_ON\_unsafe$ corresponds to case 2.

1. Draw the trajectory when mode switching
2. Give a mathematical expression for such trajectory
3. Referencing $Ctrl\_ON\_safe$, complete the encoding of $Ctrl\_ON\_unsafe$, and convince Rodin that the control logic in this case is sound(hint: prove $Ctrl\_ON\_unsafe/grd4/THM$).

Code Generation

```
if q = ON ∨ q = OFFON then
    if T_on(now + δ + t_act) ≤ T_max then
        q ← ON
    else
        q ← ONOFF
    end if
else if q = OFF ∨ q = ONOFF then
    if T_off(now + δ + t_act) ≥ T_min then
        m ← OFF
    else
        m ← OFFON
    end if
end if
```

Problems

1. Initial condition shifting might make the algorithm unnecessary complex

if $q = ON \vee q = OFFON$ **then**
    if $T_{on}(now + \delta + t_{act}) \leq T_{max}$ **then**
        q $\leftarrow$ ON
    else
        q $\leftarrow$ ONOFF ...
    end if
else if $q = OFF \vee q = ONOFF$ **then**
    if $T_{off}(now + \delta + t_{act}) \geq T_{min}$ **then**
        m $\leftarrow$ OFF
    else
        m $\leftarrow$ OFFON ...
    end if
end if

Problems

1. Initial condition shifting might make the algorithm unnecessary complex
2. Solution of differential equations might be non-unique(e.g. bessal function), or non-exists(e.g. most of real-life systems).

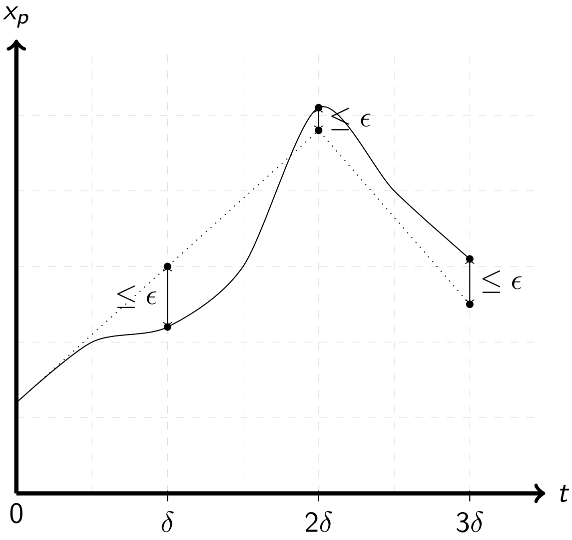
```
if q = ON ∨ q = OFFON then
    if T_on(now + δ + t_act) ≤ T_max then
        q ← ON
    else
        q ← ONOFF
    end if
else if q = OFF ∨ q = ONOFF then
    if T_off(now + δ + t_act) ≥ T_min then
        m ← OFF
    else
        m ← OFFON
    end if
end if
```

Challenge

Can we express control logic in terms of sensor reading plus evaluable terms?

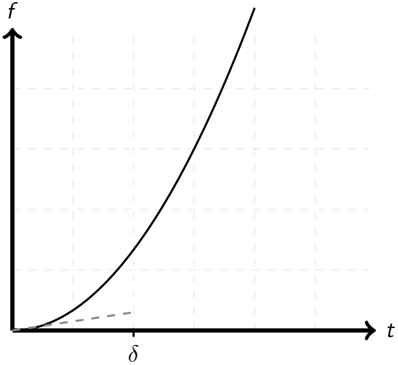
```
if q = ON ∨ q = OFFON then
    if f_1(T(now), constants) then
        q ← ON
    else
        q ← ONOFF
    end if
else if q = OFF ∨ q = ONOFF then
    if f_2(T(now), constants) then
        m ← OFF
    else
        m ← OFFON
    end if
end if
```

Proposal: Numerical Solutions + Coping with Errors



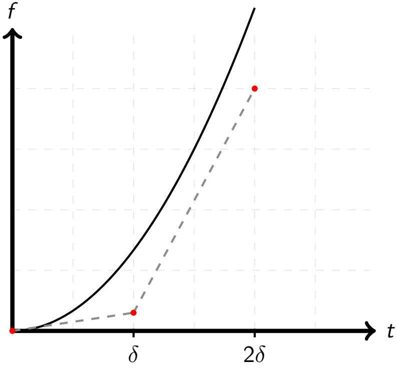
Forward-Euler Method and Truncation Errors

- Forward-Euler:
  $f_e(n+\delta)$=$f_e(n)$ + $\dot{f}$(n, f_n)*δ



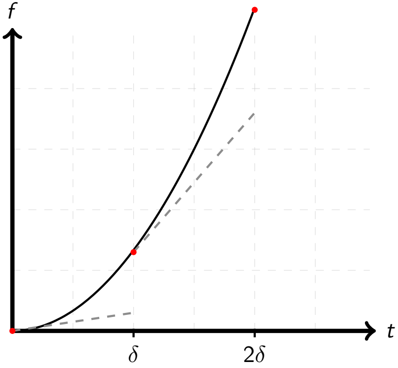
Forward-Euler Method and Truncation Errors

- Global truncation errors



Forward-Euler Method and Truncation Errors

- Local truncation errors



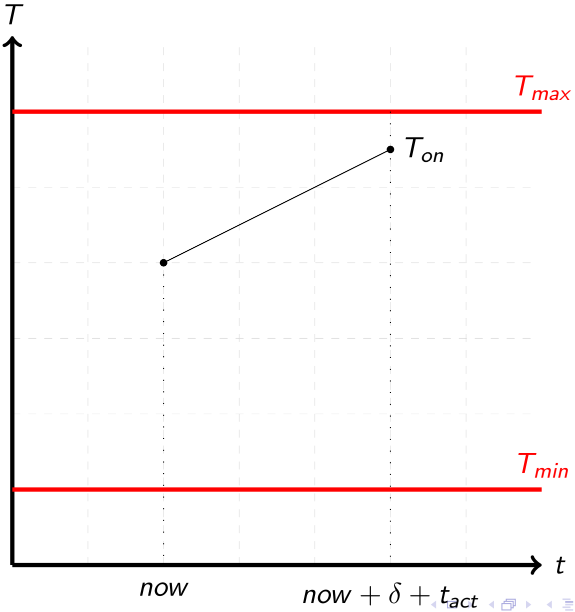
Properties of Forward-Euler Method and Truncation Errors

- Global truncation errors:
  $\mid$ f(δ) - $f_e(\delta) \mid \leq \epsilon_{gte} = \frac{\delta M}{2K}(e^{K(t-t_0)} - 1)$
- Local truncation errors:
  $\mid$ f($\delta + \triangle$) - $f_e(\delta + \triangle) \mid \leq \epsilon_{lte} =$ M
- Derivation of these properties can be found at this tutorial: [ref]

New Properties of Heating System

- (prop^{lte}_{on}) $|T_{on}(now + \delta + t_{act})$ - $Te_{on}(now + \delta + t_{act})| \leq \epsilon^{lte}_{on}$
- $(\text{prop}^{lte}_{off})$ $|T_{off}(now + \delta + t_{act})$ - $Te_{off}(now + \delta + t_{act})| \leq \epsilon^{lte}_{off}$
- $(prop_{\dot{T_{on}}})$ $\dot{T_{on}}(now,\ T_{on}(now)) = 1$
- $(prop_{\dot{T_{off}}})$ $\dot{T_{off}}(now,\ T_{off}(now)) =$ -1

Case 1: ON mode safe



Case 1: ON mode safe

$$\begin{aligned} T_{on}(now + \delta + t_{act}) &\leq Te_{on}(now + \delta + t_{act}) + \epsilon_{on}^{lte} \\ &= T_{on}(now) + \dot{T}_{on}(now, T_{on}(now)) \cdot (\delta + t_{act}) + \epsilon_{on}^{lte} \\ &= T_{on}(now) + (\delta + t_{act}) + \epsilon_{on}^{lte} \\ &\leq T_{max} \end{aligned}$$

Case 2: ON mode unsafe

$$\begin{aligned} T_{on}(now + \triangle) &= ... \\ &> T_{max} \end{aligned}$$

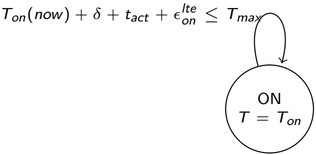
Practice

In M_5_euler,

1. Encode control logic of case 1 in terms of Euler approximation in the $grd4$ of event $Ctrl\_ON\_safe$.
2. Using the derivation on page.36, prove $Ctrl\_ON\_safe/thm01/THM$ - $Ctrl\_ON\_safe/thm04/THM$.
3. Finsh the derivation on page.37, encode this control logic of case 2 in terms of Euler approximation in the $grd4$ of event $Ctrl\_ON\_unsafe$.
4. Prove $Ctrl\_ON\_unsafe/thm01/THM$ - $Ctrl\_ON\_unsafe/thm04/THM$.

Simulation: Automata from Event-B

Event $Prediction_1$ $\widehat{=}$
Any reading
Where
...
grd_i: q = ON
grd_j:
$T_{on}(now) + \delta + t_{act} + \epsilon_{on}^{lte} \leq T_{max}$
Then
...
act_i: q = ON
act_j: fa = T_{on}
End

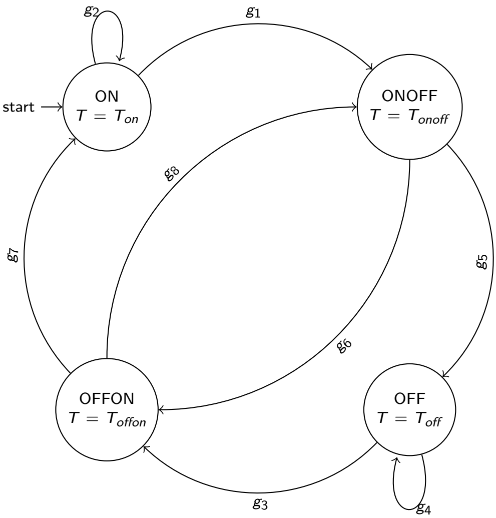


Practice

In *M_5_euler*,

1. Examine all the control logic events, draw the automata for the heating system.

Simulation



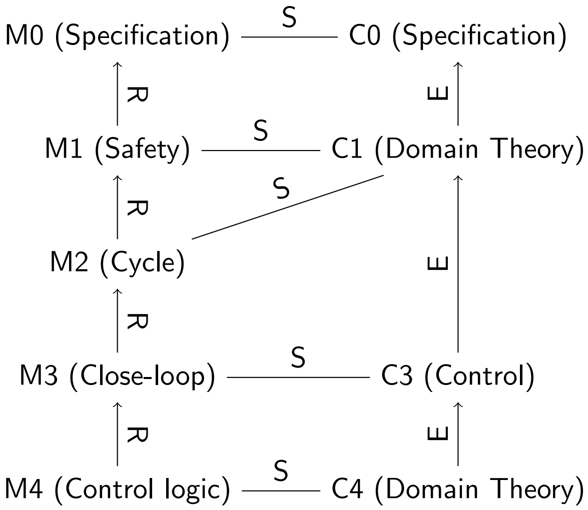
start
ON
$T = T_{on}$
ONOFF
$T = T_{onoff}$
OFFON
$T = T_{offon}$
OFF
$T = T_{off}$
g_1
g_2
g_3
g_4
g_5
g_6
g_7
g_8

Simulation in Stateflow

- Demo
- More reference:
  - Download matlab for UL students: [link]
  - Getting started with Stateflow: [link]
  - Temporal logic operators in Stateflow: [link]

Conclusion

- A refinement strategy for design dependable hybrid system



Conclusion

- A refinement strategy for design dependable hybrid system
- Propose different refinement strategies to design control logic
  - Based on modelling numerical solutions, and coping with truncation errors
  - Adaptable to deal with sensor errors or round-off errors