Formal Verification and Simulation: Co-Verification for Subway Control Systems

Huixing Fang,Jian Guo,Huibiao Zhu and Jianqi Shi

East China Normal University

Outline

Outline

1. **Motivation:** Pros and cons of formal verification and simulation.

Introduction

1. **Motivation:** Pros and cons of formal verification and simulation.
2. **Hybrid Automata:** To model and verify hybrid systems that consists of digital controllers within a continuous environment.

1. **Motivation:** Pros and cons of formal verification and simulation.
2. **Hybrid Automata:** To model and verify hybrid systems that consists of digital controllers within a continuous environment.
3. **Matlab Simulink:** Simulink supports linear and non-linear systems, continuous time, sampled time, or a hybrid of the two.

1. **Motivation:** Pros and cons of formal verification and simulation.
2. **Hybrid Automata:** To model and verify hybrid systems that consists of digital controllers within a continuous environment.
3. **Matlab Simulink:** Simulink supports linear and non-linear systems, continuous time, sampled time, or a hybrid of the two.
4. **Matlab Stateflow:** Stateflow works with Simulink, interactive graphical design, event-driven systems, transitions in response to events and conditions. (fan, motor, or pump, etc.).

Hybrid Automata

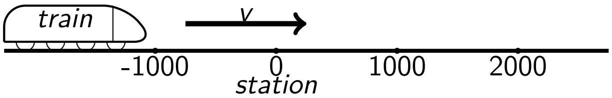


Figure: A train on a line.

Hybrid Automata

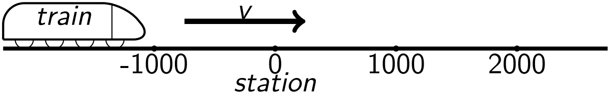


Figure: A train on a line.

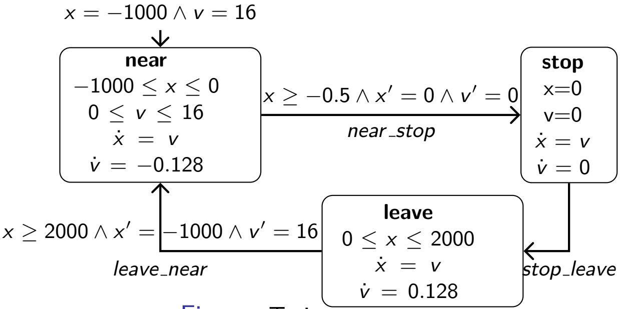


Figure: Train automaton.

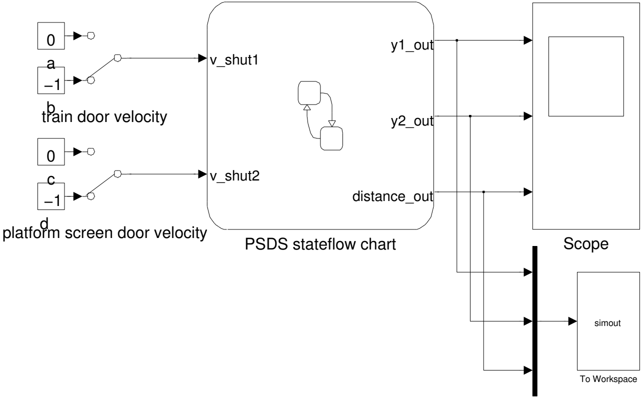


Figure: Simulink model of PSDS.

Stateflow

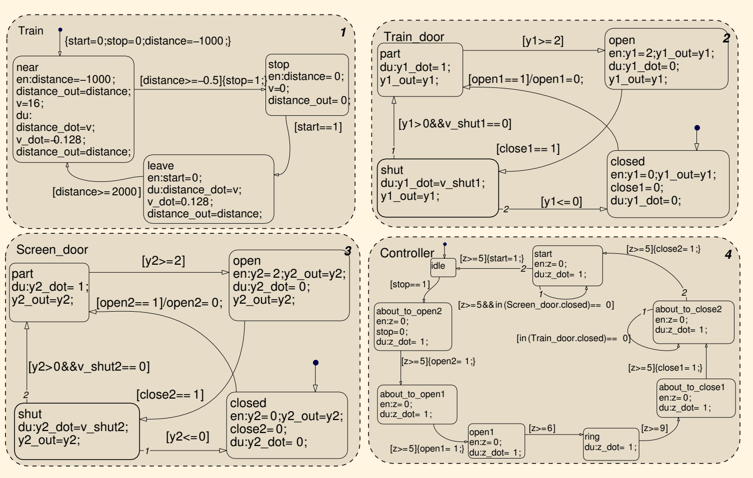


Figure: Stateflow chart of PSDS.

Outline

Methodology:Feedback-Advancement Verification

Base Phase:

Construct models by formal method and simulation technology, respectively. Liveness properties, coarse-grained.

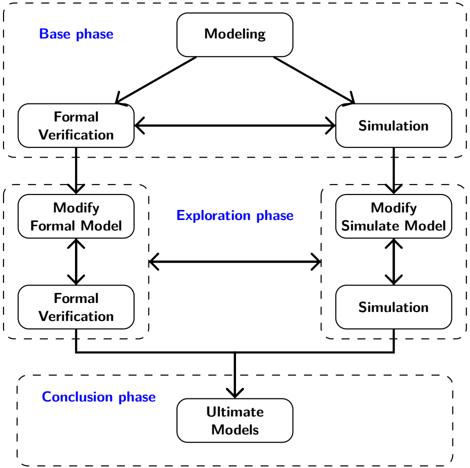


Figure: Feedback-Advancement Verification.

Methodology:Feedback-Advancement Verification

Base Phase:

Construct models by formal method and simulation technology, respectively. Liveness properties, coarse-grained.

Exploration Phase:

Safety critical properties. Feedback between simulation and verification. Refine models if necessary.

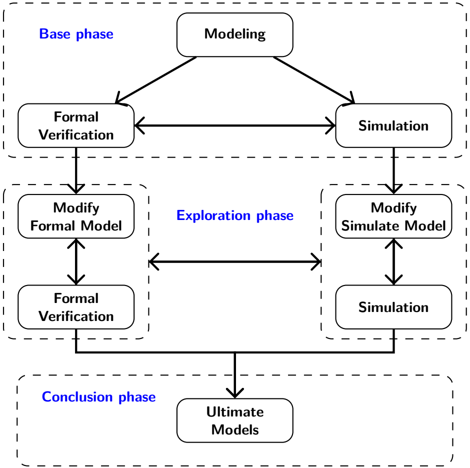


Figure: Feedback-Advancement Verification.

Methodology:Feedback-Advancement Verification

Base Phase:

Construct models by formal method and simulation technology, respectively. Liveness properties, coarse-grained.

Exploration Phase:

Safety critical properties. Feedback between simulation and verification. Refine models if necessary.

Conclusion Phase:

Final models selection.

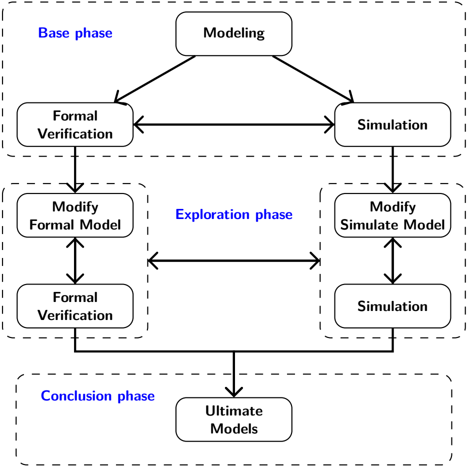


Figure: Feedback-Advancement Verification.

Outline

Different Behaviors of Platform Screen Doors:

1. **Simultaneously:** The platform screen (edge) doors and train doors close simultaneously.

Different Behaviors of Platform Screen Doors:

1. **Simultaneously:** The platform screen (edge) doors and train doors close simultaneously.
2. **Platform Screen Doors First (PSDF):** First, close the platform screen (edge) doors. And then close the train doors.

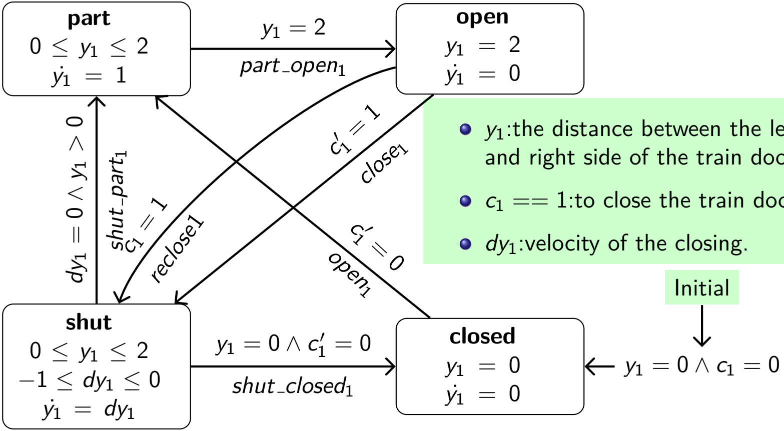
Requirements of Platform Screen Doors System

Different Behaviors of Platform Screen Doors:

1. **Simultaneously:** The platform screen (edge) doors and train doors close simultaneously.
2. **Platform Screen Doors First (PSDF):** First, close the platform screen (edge) doors. And then close the train doors.
3. **Train Doors First (TDF):** First, close the train doors. And then close the platform screen doors.

Outline

Train Doors Automaton



- y_1:the distance between the left side and right side of the train door.
- $c_1 == 1$:to close the train doors.
- dy_1:velocity of the closing.

Figure: Train doors automaton.

Train Doors Chart

- open1: local variable, as a shared variable between train doors and controller.
- v_shut1: the velocity of closing.
- y1_out: output variable for plotting in the Scope.
- transition: event[condition]condition_action/transition_action.

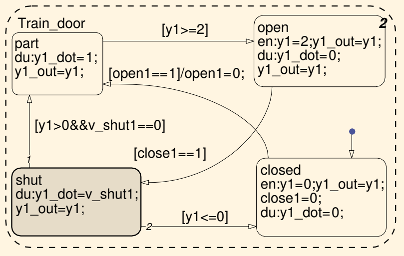


Figure: Train doors chart

Controller Automaton

- z: stop watch.
- Synchronization labels:
  - open2: open the platform screen doors.
  - open1: open the train doors.
  - close1: close the train doors.
  - close2: close the platform screen doors.

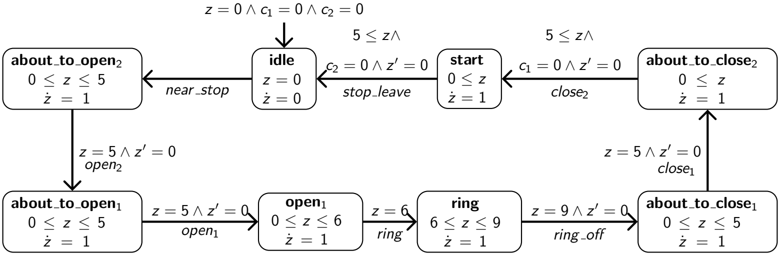


Figure: Controller automaton.

- in(Train_door.closed): denotes whether train doors are closed.

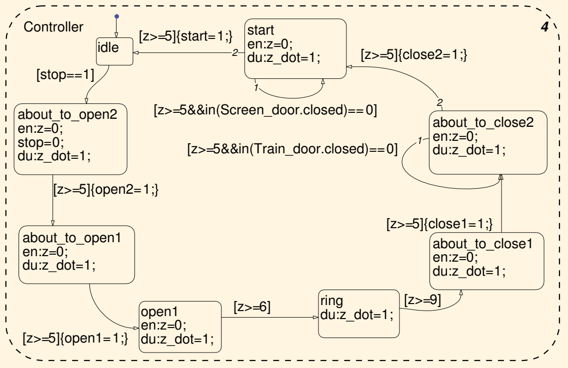


Figure: Controller chart

Outline

Verification

Tool: PHAVer(Goran Frehse, 2005-2007).

Tool: PHAVer(Goran Frehse, 2005-2007).

Property 1: Leaving and Stopping

whether the train can leave from the station or stop.

Tool: PHAVer(Goran Frehse, 2005-2007).

Property 1: Leaving and Stopping

whether the train can leave from the station or stop.

Property 2: Ringing

Both train doors and platform screen doors are opened when the bell is ringing

Verification

Tool: PHAVer(Goran Frehse, 2005-2007).

Property 1: Leaving and Stopping

whether the train can leave from the station or stop.

Property 2: Ringing

Both train doors and platform screen doors are opened when the bell is ringing

Property 3: Ordering

The train doors need to be closed at first, and then the platform screen doors.

Verification

Tool: PHAVer(Goran Frehse, 2005-2007).

Property 1: Leaving and Stopping

whether the train can leave from the station or stop.

Property 2: Ringing

Both train doors and platform screen doors are opened when the bell is ringing

Property 3: Ordering

The train doors need to be closed at first, and then the platform screen doors.

Property 4: Operation of Doors

There are no operations of doors when the train is running.

One Normal Running:

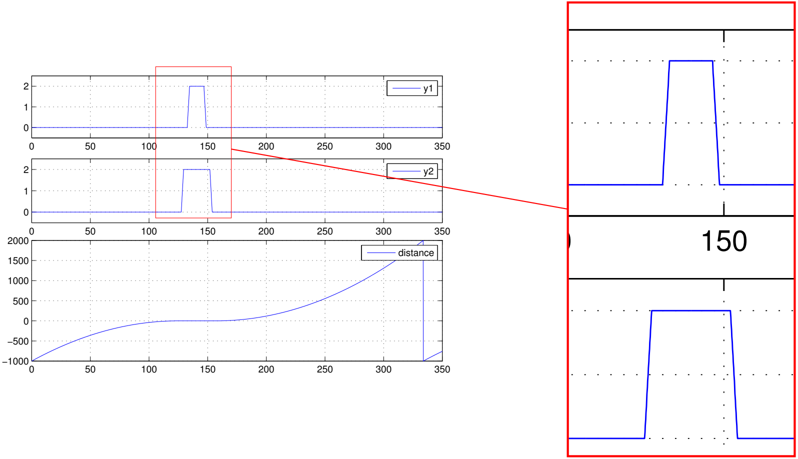


Figure: One normal running.

Sandwich Simulation

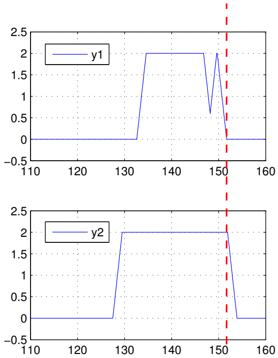


Figure: Sandwich simulation.

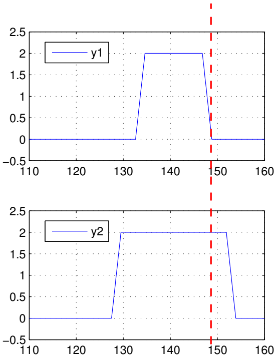


Figure: Normal simulation.

The platform screen doors begin to close almost at the moment the train doors closed.

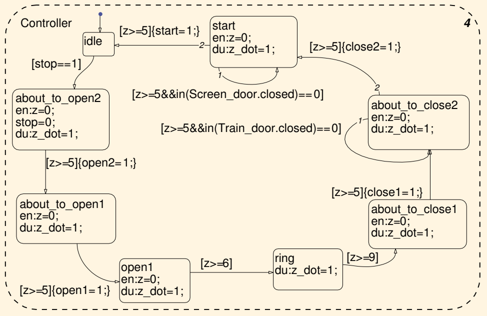


Figure: Controller chart

If the train doors are closed when z == 4.9, then the predicate in(Train_door.closed) == 0 is false. Thus, the transition from about_to_close2 to start will occur when z is 5, the time interval is 0.1s.

Outline

Simulation Refinement

Remove $z >= 5$ in the predicate $z >= 5$ && $in(Train\_door.closed) == 0$:

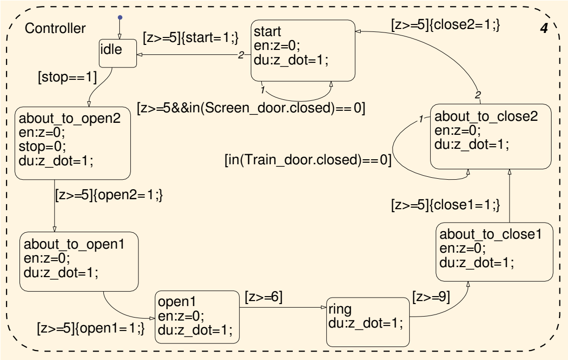


Figure: Refined Controller chart.

Simulation after Refinement

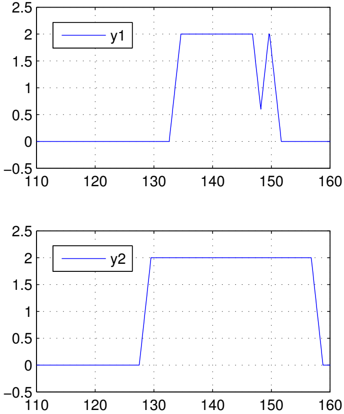


Figure: Simulation after the refinement.

Formal Verification Refinement

Add a self-loop control switch of mode *about_to_close*$_2$ in the controller automaton. *shut_closed*$_1$ is the synchronization label between train doors automaton and controller automaton.

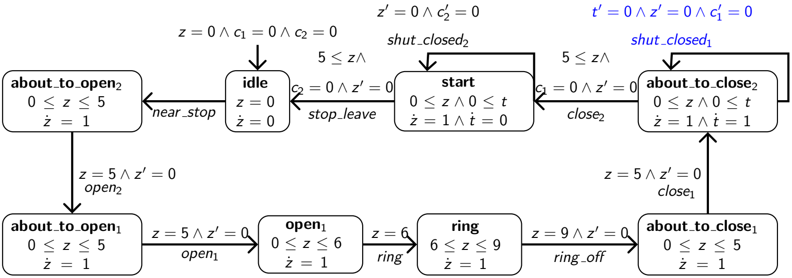


Figure: Refined Controller automaton.

Verification after Refinement

The system is represented by:

$sys = controller \& traindoor \& screendoor \& train.$

The following states are not reachable in the system:

$sys.\{start \sim \$ \sim \$ \sim \$ \& t < 5\}.$

All states when the controller is in mode *start*:

$$\begin{aligned}
controller \sim traindoor \sim screendoor \sim train.\{ \\
start \sim closed \sim \textbf{\textit{shut}} \sim stop\& \\
-1 <= dy2 <= 0\&0 <= y2 <= 2\&x == 0\&y1 == 0\& \\
close2_flag == 1\&close1_flag == 0\& \\
\boldsymbol{t >= 5}\&z + y2 >= 2, \\
start \sim closed \sim \textbf{\textit{open}} \sim stop\& \\
x == 0\&y2 == 2\&y1 == 0\&close2_flag == 1\& \\
close1_flag == 0\&\boldsymbol{t >= 5}\&z >= 0, \\
start \sim closed \sim \textbf{\textit{closed}} \sim stop\& \\
x == 0\&y2 == 0\&y1 == 0\&close2_flag == 0\& \\
close1_flag == 0\&\boldsymbol{t >= 5}\&z >= 0, \\
start \sim closed \sim \textbf{\textit{part}} \sim stop\& \\
0 < y2 <= 2\&x == 0\&y1 == 0\&close2_flag == 1\& \\
close1_flag == 0\&\boldsymbol{t >= 5}\&z + y2 >= 2\}.
\end{aligned}$$

Outline

Conclusion

- Feedback-Advancement Verification: Combine formal verification with simulation.
- Application: Platform Screen Doors System

Future work,Models Translation

- Co-Verification on subway control systems.
- The Compositional Interchange Format for Hybrid Systems(CIF). `http://se.wtb.tue.nl/sewiki/cif/Start`.
- HyLink, translate a subset of Simulink/Stateflow to HyXML format, tools, such as HyTech, UPPAAL, etc. `http://hsver.crhc.illinois.edu/index.php/HyLink`.
- Translate to Hybrid Systems Interchange Format(HSIF, XML format), `http://wiki.grasp.upenn.edu/hst/index.php?n=Main.HSIF`.

Thanks

Thank you very much!