

A Method to Import an FMU to a Hardware Description Language

Min Zhang



Agenda

- Introduction
- FMI/FMU vs MAST Model
- FMU2MAST conversion
- Bouncingball Example
- Motor Drive Example
- Q&A

Introduction FMI/FMU



<http://fmi-standard.org>

Import an FMU to an HDL(MAST)

- MAST Modeling Language

1. Invented in earlier of 80's
2. Multi-domain technology
3. Mixed-Mode signal
4. Used in automobile and aerospace industries for more than 30 years.

- Advantage

1. Reduce significant amount work in simulator engine.
2. Avoid duplicated work to import FMU in another simulator.
3. Reuse all the advanced HDL features and Simulator features.
4. Can be used with models written in other HDL to study more complex system.

FMI/FMU vs MAST (Data Types)

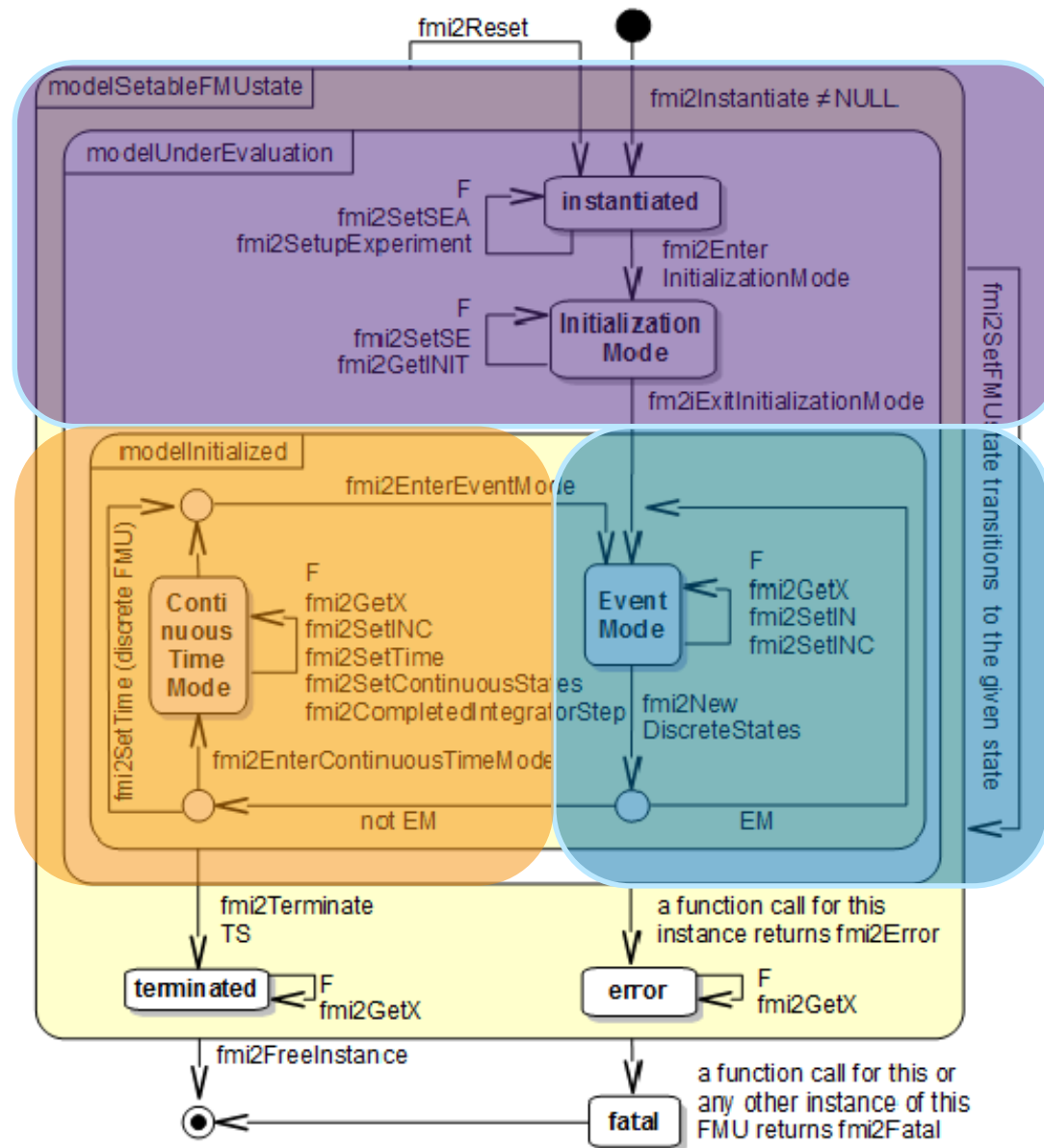
		FMI 2.0		
		causality	variability	derivative
MAST	connection	input	input	continuous
		output	output	continuous
		state	(in)	input
			(out)	output
	variable	Parameter	parameter	fixed
		Constant	calculatedParameter	fixed
			local	fixed
			local	constant
		State	parameter	tunable
			independent	
			local	discrete
			local	tunable
		Val	local	continuous
		Var	local	continuous
				yes

FMI/FMU vs MAST (Simulation flow)

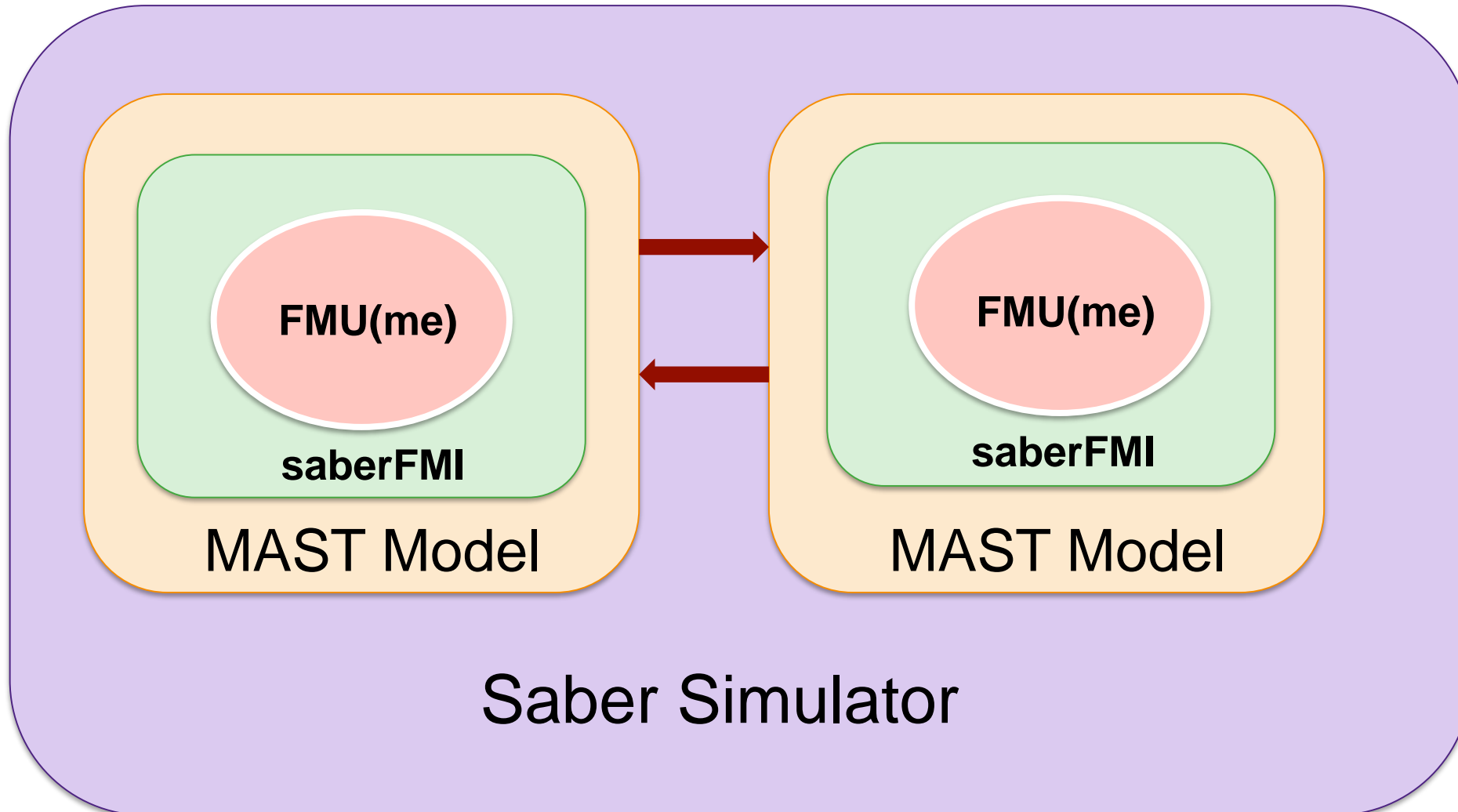
Elaboration:
Parameters
Section

Analog Engine:
Values +
Equations
Section

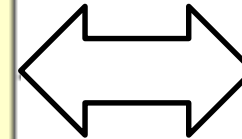
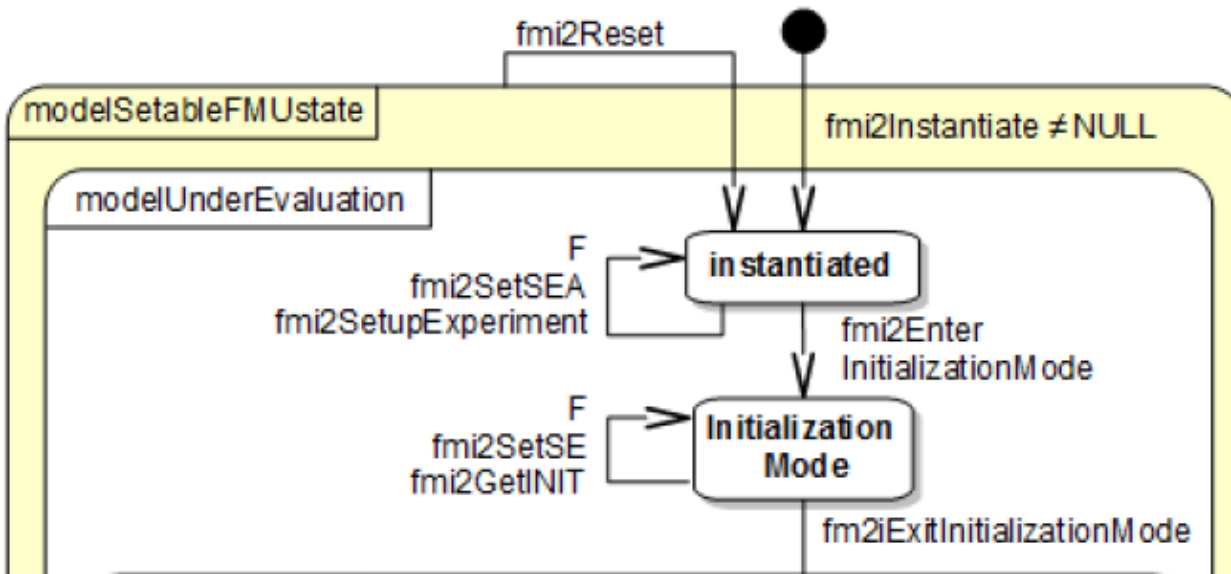
Digital Engine:
When Sections



Interface between MAST and FMI



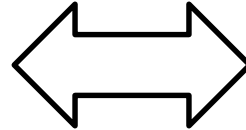
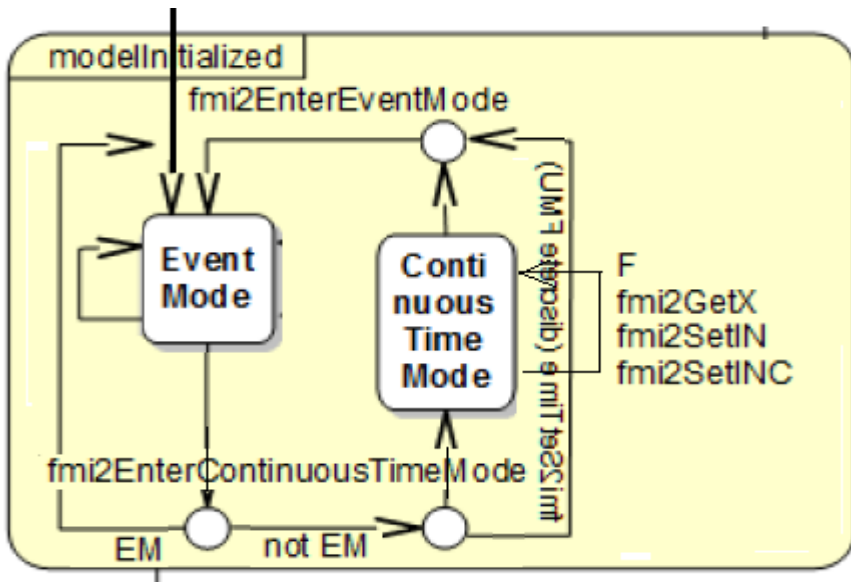
Parameter Section



```

Parameters {
    fmiHandle = saberfmi(initialization...) {
        fmi2Instantiate();
    }
    saberfmi(setValues,fmiHandle,...) {
        fmi2setReal/Integer/Boolean();
    }
    saberfmi(updateValues,fmiHandle...) {
        fmi2EnterInitializationMode();
        fmi2ExitInitializationMode();
    }
    h_ic = saberfmi(getValues,fmiHandle...) {
        fmi2GetReal/Integer/Boolean();
    }
}
  
```


Values Section

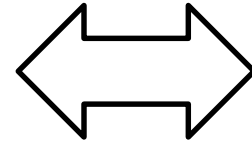
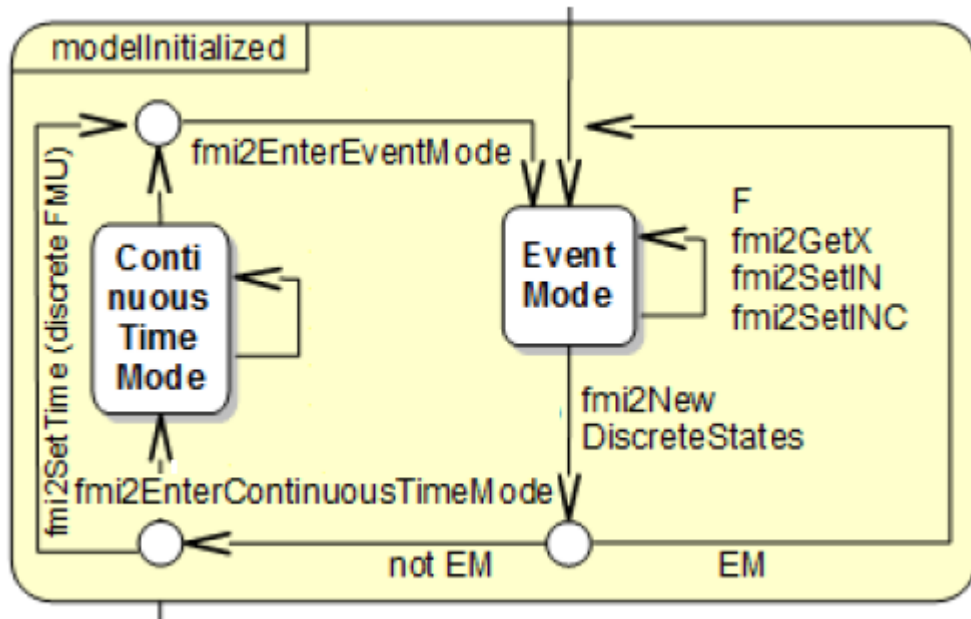


```

Values {
    saberfmi(timeValue,fmiHandle,input...) {
        fmi2setReal ();
    }
    xIndicator=saberfmi(checkCrosss,fmiHandle,time...) {
        fmi2getEventIndicators();
    }
    der_h=saberfmi(nonlinear,fmiHandle,time,h,v) {
        fmi2setReal();
        fmi2completedIntegratorStep();
        fmi2getReal();
    }
}

Equations {
    h : d_by_dt(h) = der_h
    v0: d_by_dt(v0) = der_v
    v : v = v_init + delta_v0
}
    
```

When Sections



```
When(event_on(in1)) {
  saberfmi(setEvent,fmiHandle,...) {
    fmi2enterEventMode();
    fmi2setReal/Integer();
  }
  nextEvent=saberfmi(checkEvents,fmiHandle,...) {
    while(newDiscreteStatesNeeded) {
      fmi2newDiscreteStates();
    }
    fmi2enterContinuousTimeMode();
  }
  schedule_next_time(nextEvent)
}
```

FMU2MAST Conversion

The screenshot shows the Synopsys SaberRD software interface. The main window displays a block diagram of a motor test system. The diagram includes components like `modelica_demo_stimuli`, `fmi_state2real`, `modelica_demo_control`, `modelica_demo_drive`, and `loadtorque_nm`. A context menu is open over the 'fmu' library, showing options like 'New Part', 'New Subcategory', 'Add Items...', 'Import Spice...', 'Edit...', 'Export Library...', 'Update Library...', 'Compile Library...', and 'Update All Libraries...'. A 'Select Items to Add To Library' dialog is open, showing the selection of 'test_vanderpol.sin' from the 'Data_D (D:) > tmp > vanderpol' directory. The dialog also shows a list of files in the directory, including 'good' and 'test_vanderpol.sin'. The 'File name' field is empty, and the 'Source' dropdown is set to 'Source (*.sin,*.vhd,*.vhd)'. The 'Attributes' panel on the right shows the 'Misc' section with various properties like 'data type', 'qualifier', 'visibility', 'font', 'text style/color', 'background style/color', 'outline', 'rotation', and 'anchor'.

motor_test2 - SaberRD

File Home Model Simulate Analyze View

Generic Parts Components

Search Parametric Search Find Parts

Modeling Tools

Add Library Edit Library Library Configuration

Libraries

Parts Models

Generic Parts Compone

Motors & Drivers

Pneumatic

Saber-Virtualizer Interf

Schematic Connectors

Signal Processing

Sources

Thermal

VDA Automotive

VDA Fundamentals

VDA Hybrid EMC

VDA Megma

VDA Modelica Rotation

VDA Modelica Translati

VDA Spice

VDA Spice2vhd_device

ai_User_Library

fmu

bouncingball

modelica_bloc

vanderpol

Information

Library Name fmu

Config Type user

Library Root C:/Users/m

Compile T... 05/07/18 0

Dependen... ai_std mast

Libr... Desi...

motor_test2

motor_test2.ai_sch (/)* x

StartPage

fmi_state2real

in

out w_desired

w

v

v

w

loadtorque_nm

modelica_demo_stimuli

modelica_demo_control

modelica_demo_drive

fmi_state2real

in

out

Select Items to Add To Library

<< Data_D (D:) >> tmp >> vanderpol >>

Search vanderpol

Organize New folder

Recent Places

Libraries

Documents

Music

Pictures

Videos

Computer

OS (C:)

Data_D (D:)

saberstore (\\slowfs-sabers

Name

Date modified

good

test_vanderpol.sin

5/3/2018 10:55 AM

5/3/2018 10:55 AM

File name:

Source (*.sin,*.vhd,*.vhd)

Symbols/Source (*.ai_sym,*.sin,*.vhd,*.v

Symbols (*.ai_sym)

Source (*.sin,*.vhd,*.vhd)

Spice Models (*.sp,*.spi,*.cir,*.ckt,*.mod

FMU Models (*.fmu)

Foreign Routines (*.c,*.cc,*.C,*.cxx,*.cpp

All Files (*.*)

Attributes

Misc

data type String

qualifier

visibility off

font

text style/color

background style/color

outline No

rotation 0

anchor c

data type

The data type of the property.

1.20", -1.21"

100%

SaberRD

© 2018 Synopsys

synopsys

FMU2MAST conversion

- Verified by 44 different tests, 5 different tools.

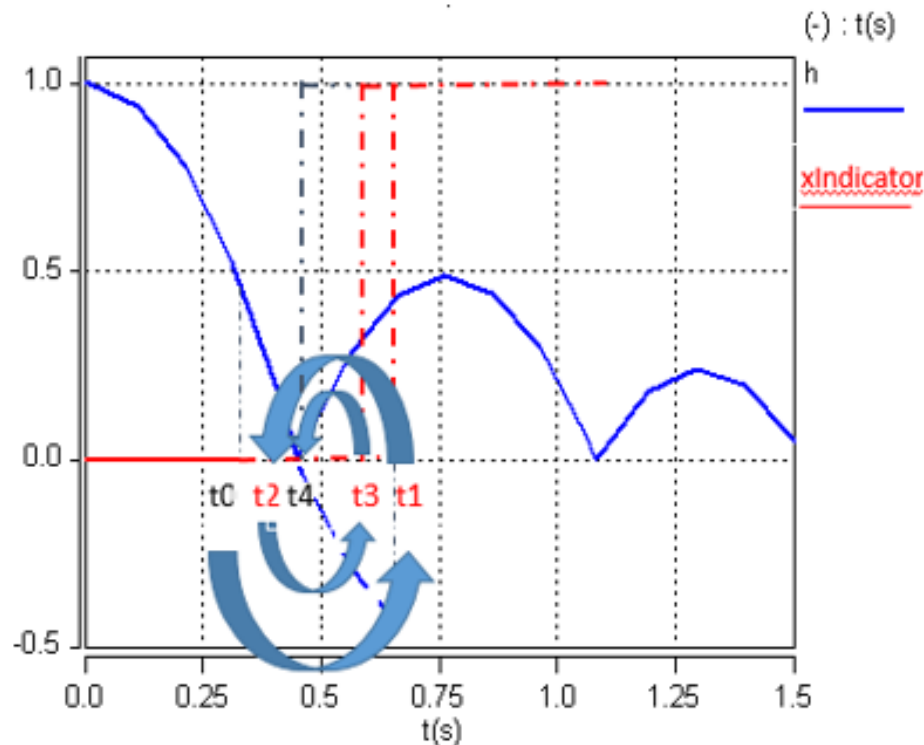
	vendor	input	output	discrete	state	others
<i>Controlledtanks</i>	OpenModelica	0	0	246	4	535
<i>Motor_drive2</i>	Dymola	0	1	39	15	253
<i>MixtureGases</i>	Dymola	0	2	4	16	76
<i>DFFREG</i>	Dymola	0	2	40	0	44
<i>CoupledClutches</i>	Dymola	1	4	50	18	123
<i>Rectifier</i>	Mworks	0	8	6	4	177
<i>ControlledTemp</i>	Mworks	0	2	4	1	64
<i>BooleanNetwork</i>	Mworks	1	9	57	0	29
<i>FullRobot</i>	Mworks	0	6	109	36	5259
<i>BouncingBall</i>	FMUSDK	0	0	1	2	3
<i>vanderpol</i>	FMUSDK	0	0	0	2	3

BouncingBall Example (Accurate Event Detection)

$$\frac{dv}{dt} = -g \quad (1)$$

$$\frac{dh}{dt} = v \quad (2)$$

$$v = -e * v_0 \quad (3)$$



```

values {
  xIndicators = saberfmi(checkCross, fmiHandle, time)
}

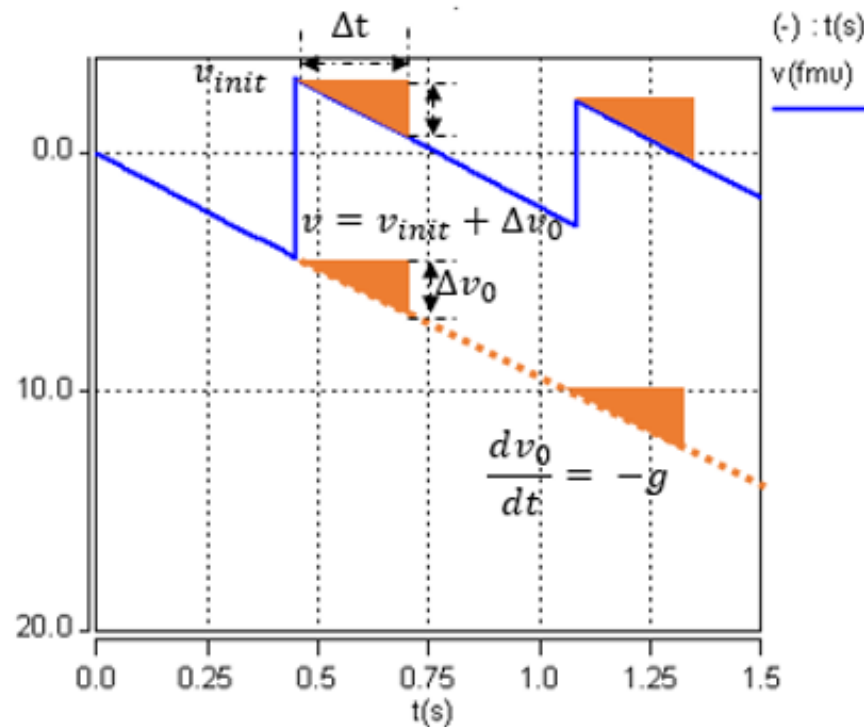
when(threshold(xIndicators, 0.5, before, after) & after > 0 ) {
  schedule_event(time, cross, 1.0)
}

when(event_on(cross)) {
  stateErr = saberfmi(updateCross, fmiHandle, time)
  nextEvent = saberfmi(checkEvents, fmiHandle, time)
  hasBreak = saberfmi(valuesChanged, fmiHandle)
  v_init = saberfmi(timeValues, fmiHandle, time, v_id, fmiReal)
  prev_v_0 = v_0
  schedule_next_time(time)
}
    
```

BouncingBall Example (Re-Initialization)

$$\frac{dv_0}{dt} = -g \quad (1)$$

$$v = v_{init} + \Delta v_0 \quad (2)$$



```

values {
  xIndicators = saberfmi(checkCross, fmiHandle, time)
  delta_v_0 = v_0 - prev_0_0
}

when(threshold(xIndicators, 0.5, before, after) & after > 0 ) {
  schedule_event(time, cross, 1.0)
}

when(event_on(cross)) {
  nextEvent = saberfmi(checkEvents, fmiHandle, time)
  hasBreak = saberfmi(valuesChanged, fmiHandle)
  v_init = saberfmi(timeValues, fmiHandle, time, v_id, fmiReal)
  prev_v_0 = v_0
  schedule_next_time(time)
}

equations {
  v_0: d_by_dt(v_0) = -g # eqn. (1)
  v: v = v_init + delta_v_0 # eqn. (2)
}
    
```

BouncingBall Example (Results)

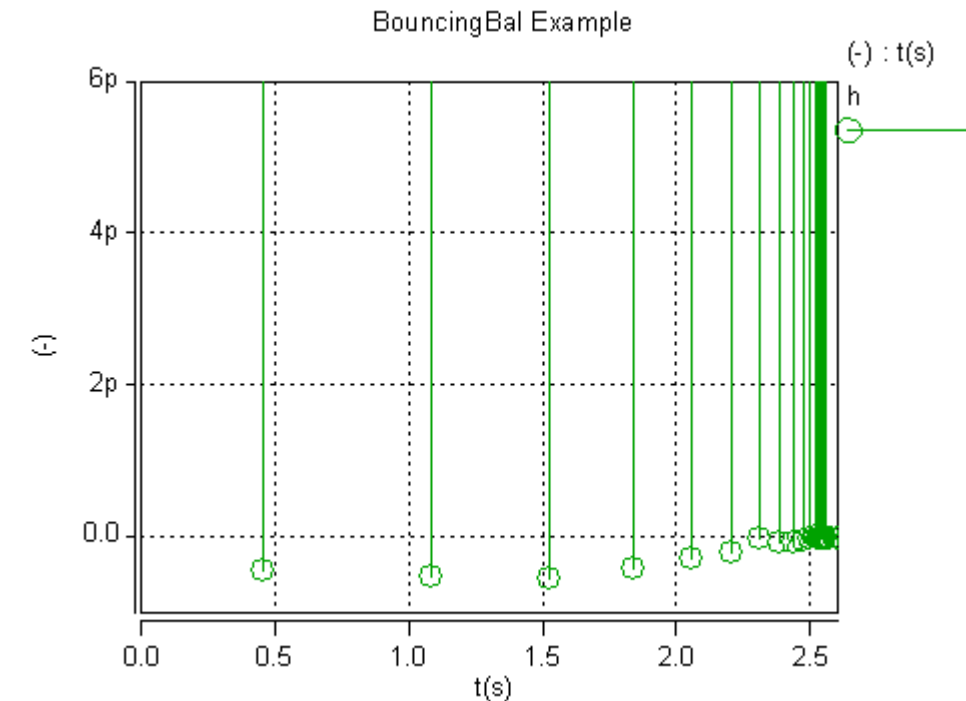
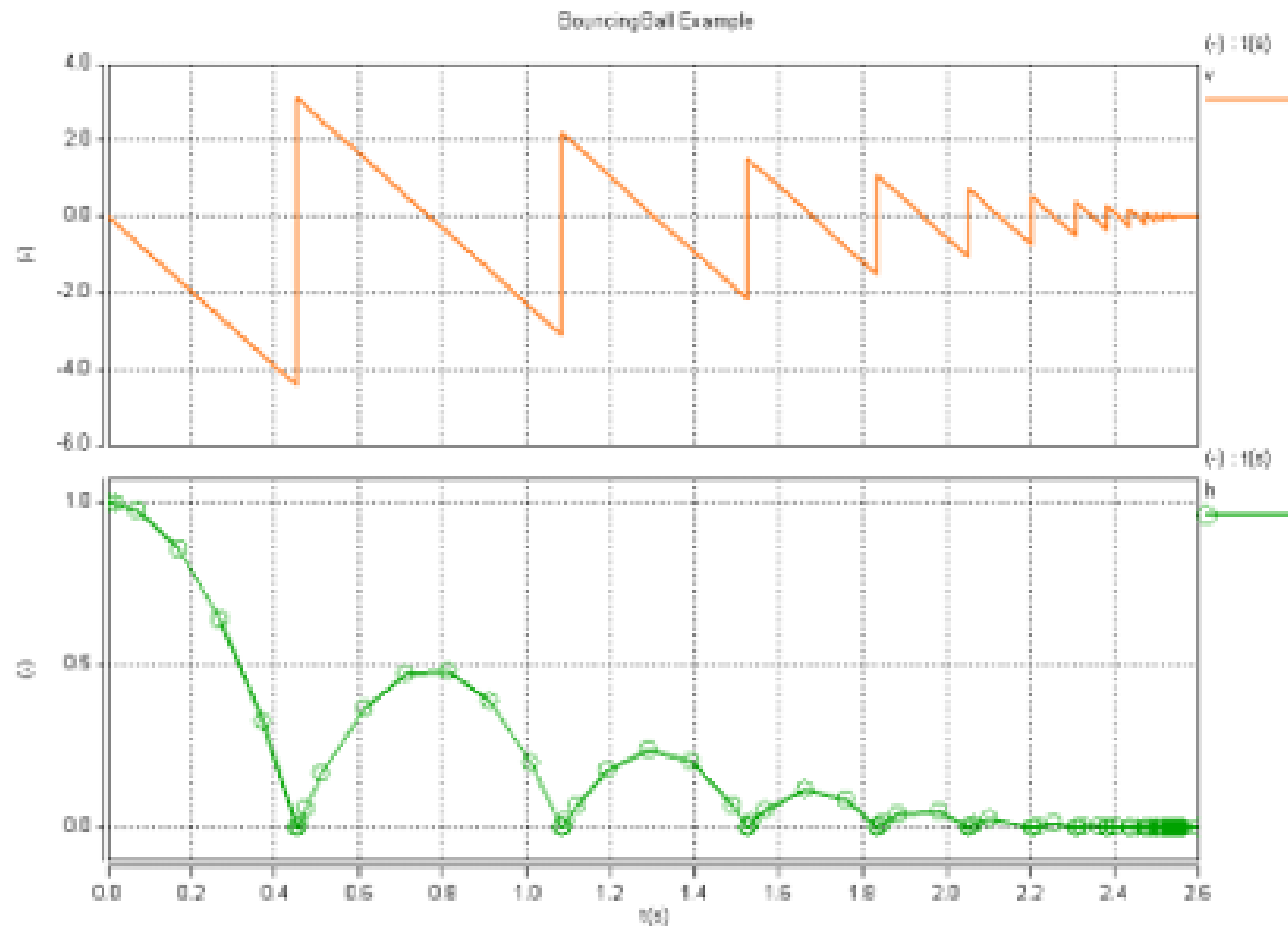


Figure 10 Bouncing Ball Results

Motor Drive Example (Equations)

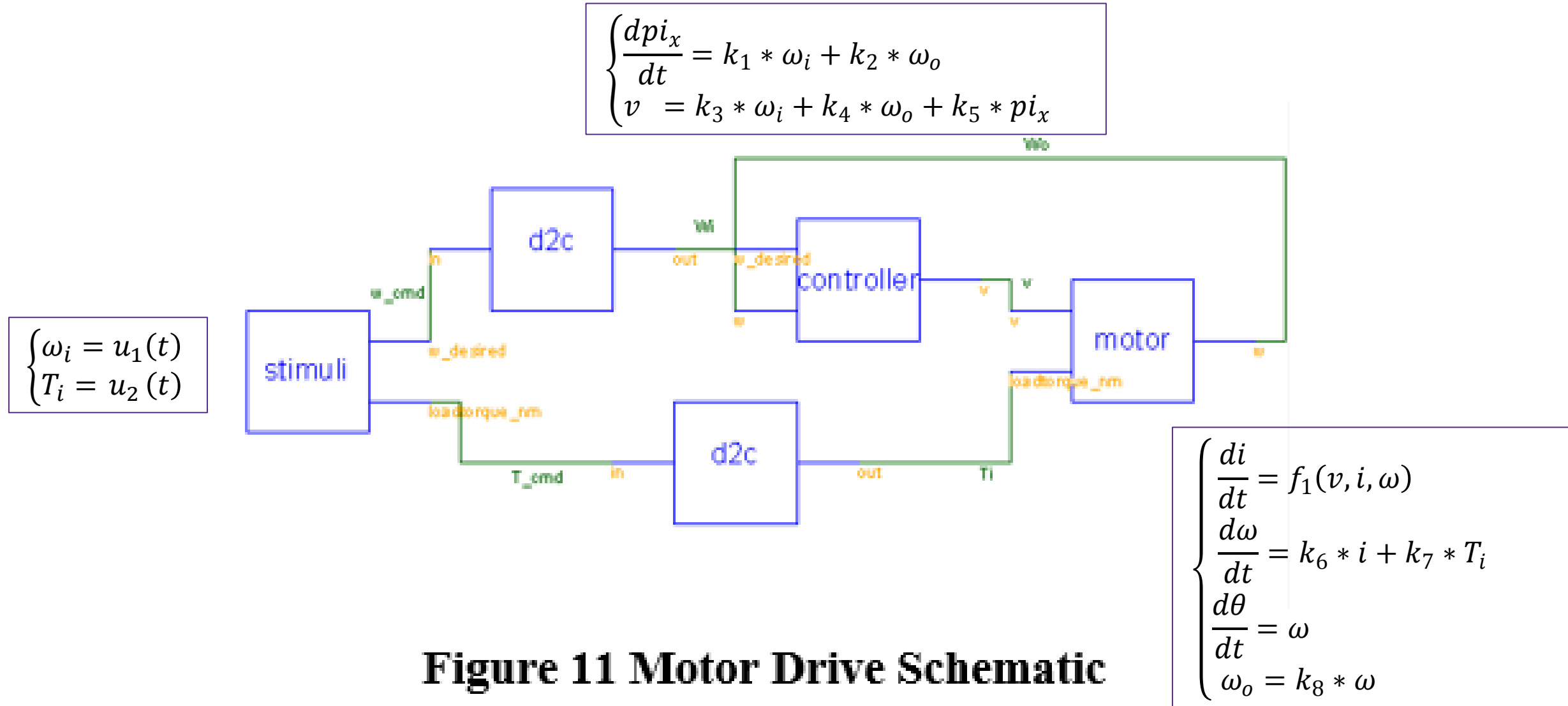


Figure 11 Motor Drive Schematic

Motor Drive Example (DAE formulation)

$$Ax + E\dot{x} = B(t)$$

$$F'(X_{k-1})X_k = -F(X_{k-1}) + X_{k-1}F'(X_{k-1})$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & 0 & 0 & 0 & 0 & 0 & 0 & -k_2 \\ -k_3 & 0 & -k_5 & 0 & 0 & 0 & 0 & -k_4 \\ 0 & 0 & -\frac{\partial f_1}{\partial v} & 0 & -\frac{\partial f_1}{\partial \omega} & 0 & 0 & 0 \\ 0 & -k_7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_8 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \omega_i \\ T_i \\ p i \\ v \\ i \\ \omega \\ \theta \\ \omega_o \end{pmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} \dot{\omega}_i \\ \dot{T}_i \\ \dot{p} i \\ \dot{v} \\ \dot{i} \\ \dot{\omega} \\ \dot{\theta} \\ \dot{\omega}_o \end{pmatrix}$$

$$= \begin{pmatrix} u_1(t) \\ u_2(t) \\ 0 \\ -f_1(v_{k-1}, i_{k-1}, \omega_{k-1}) - \frac{\partial f_1}{\partial v} v_{k-1} + \frac{\partial f_1}{\partial i} i_{k-1} + \frac{\partial f_1}{\partial \omega} \omega_{k-1} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Motor Drive Example (Results)

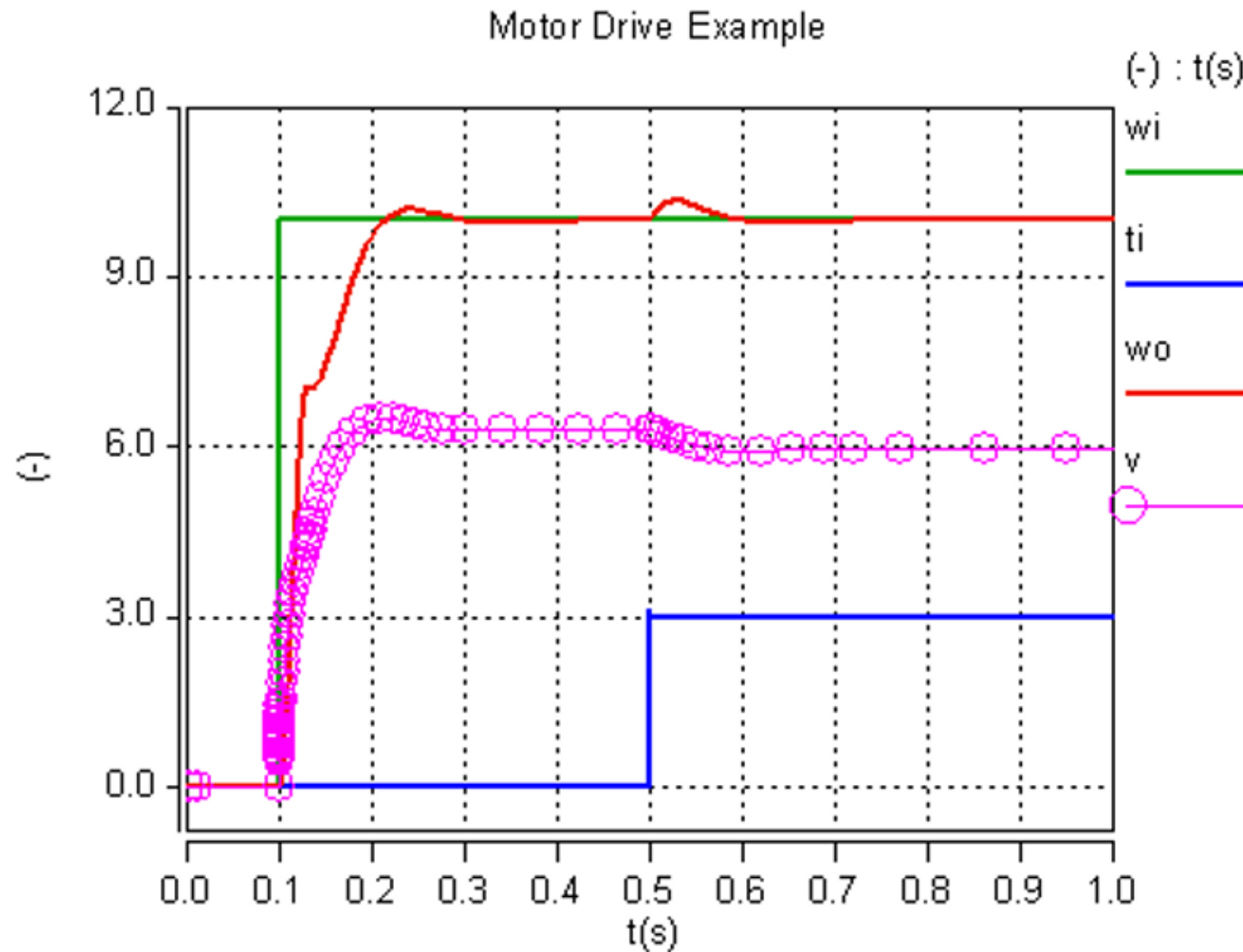
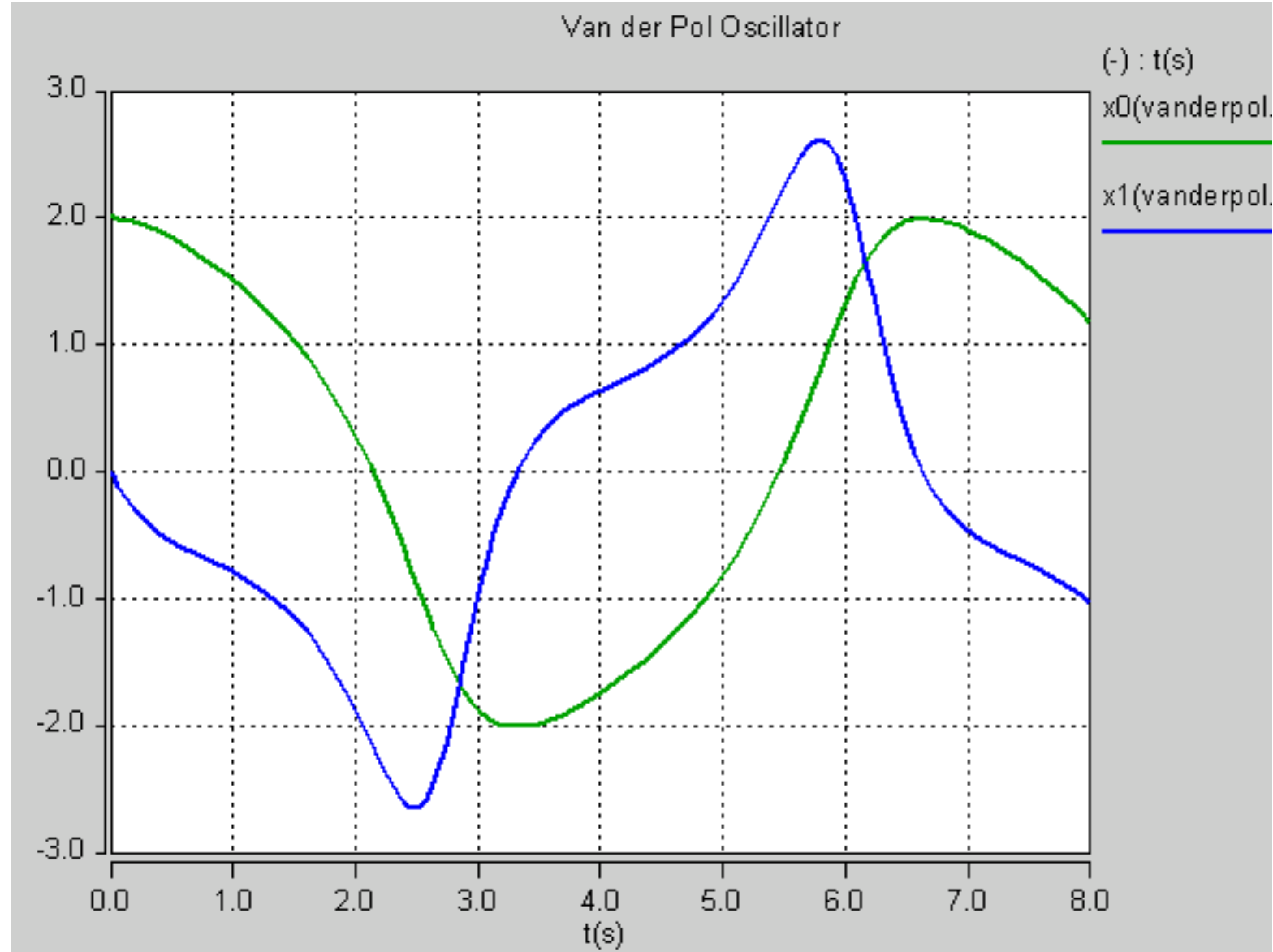


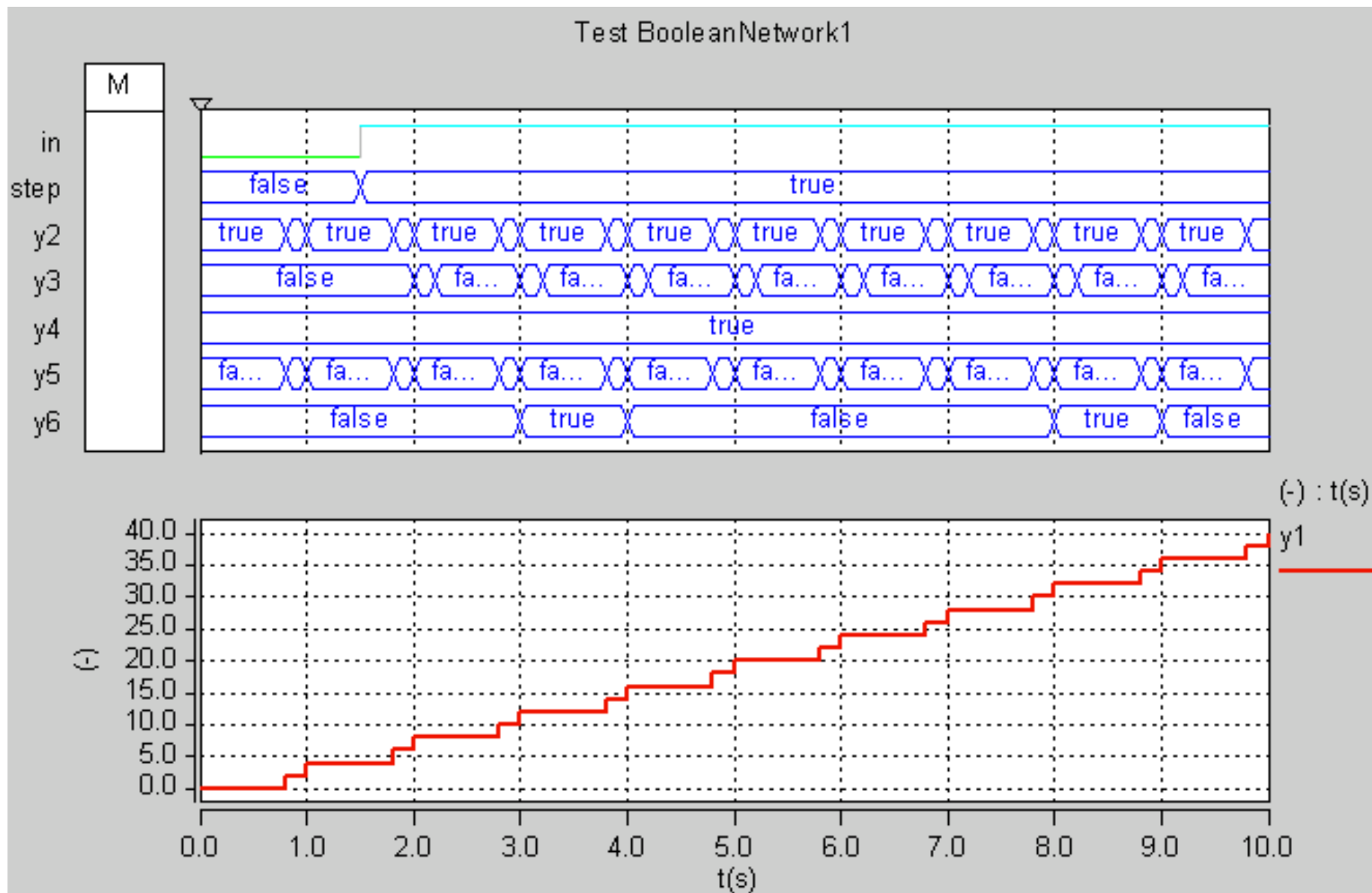
Figure 12 Motor Drive Results

Other Examples (Van der Pol oscillator)

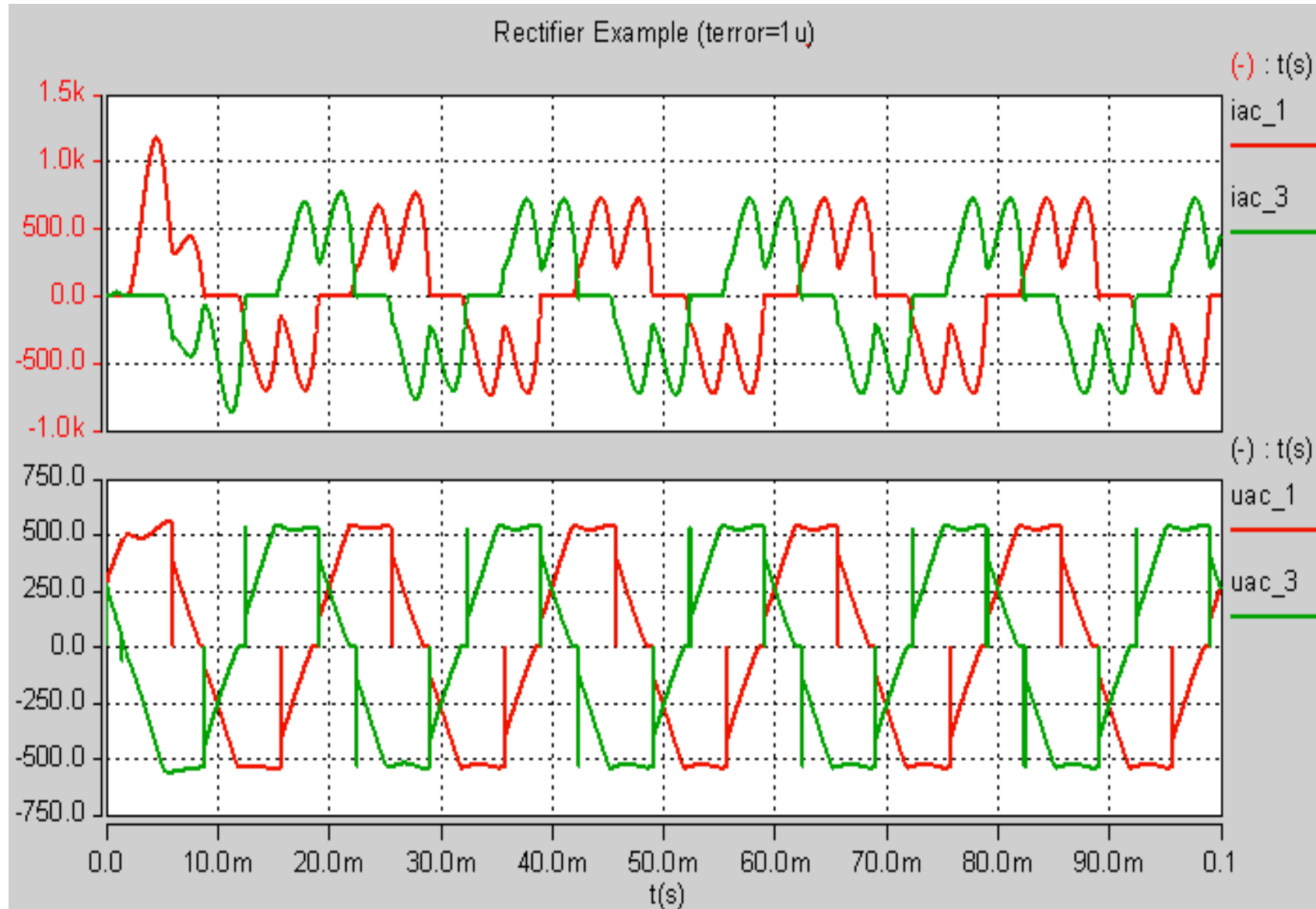
$$\begin{cases} \frac{dx_0}{dt} = x_1 \\ \frac{dx_1}{dt} = \mu * ((1 - x_0 * x_0) * x_1) - x_0 \end{cases}$$



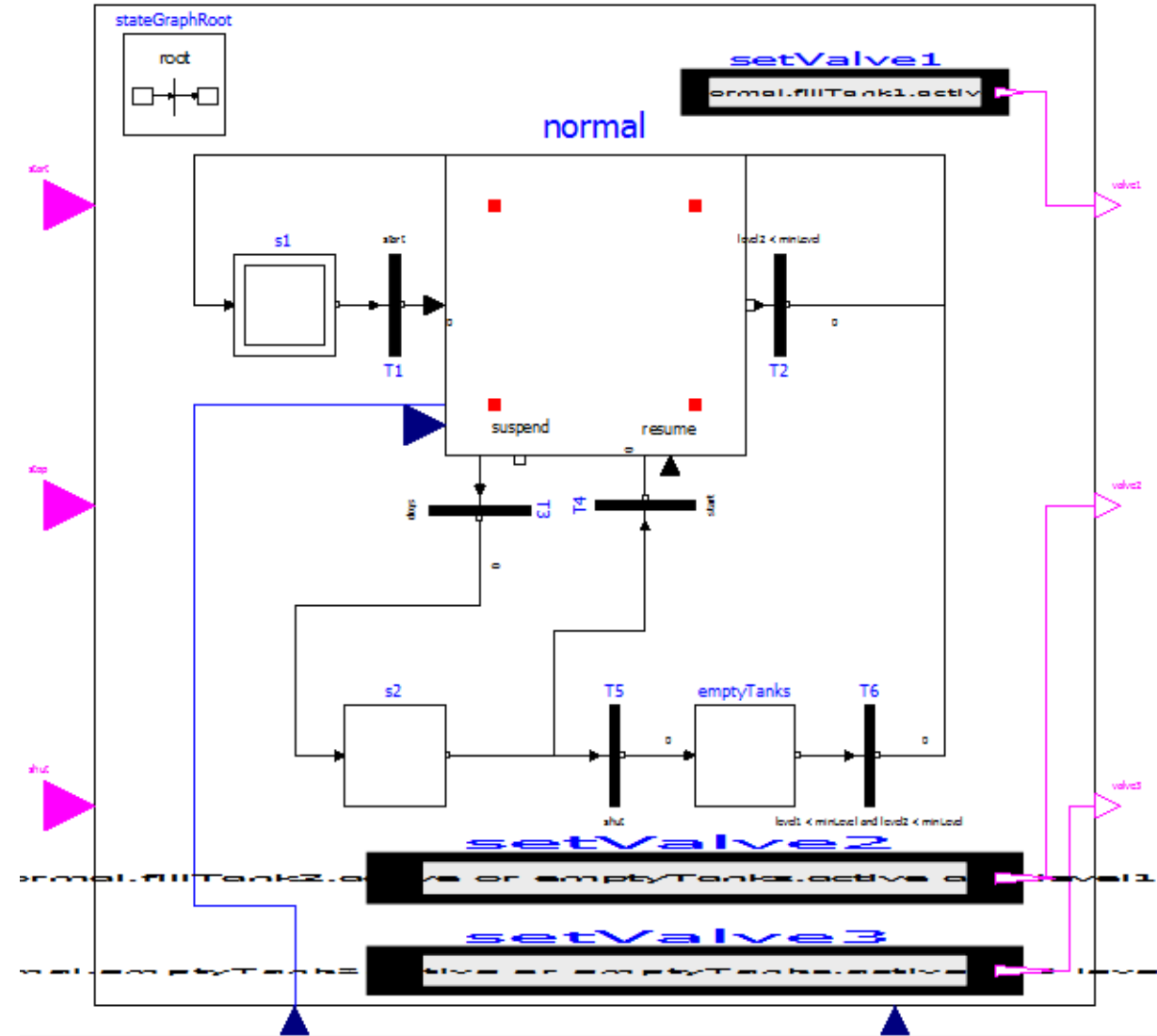
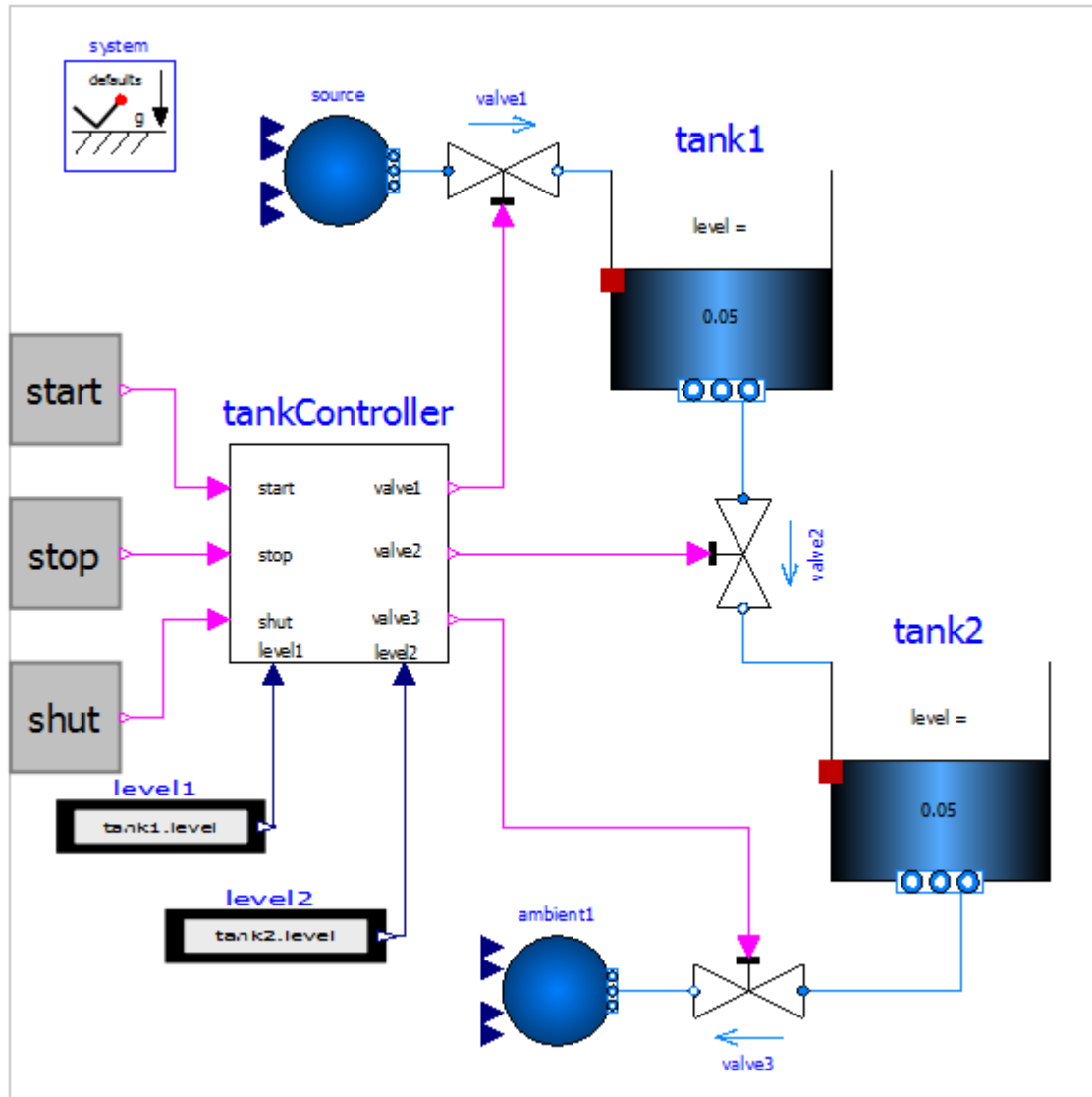
Other Examples (booleanNetwork1)



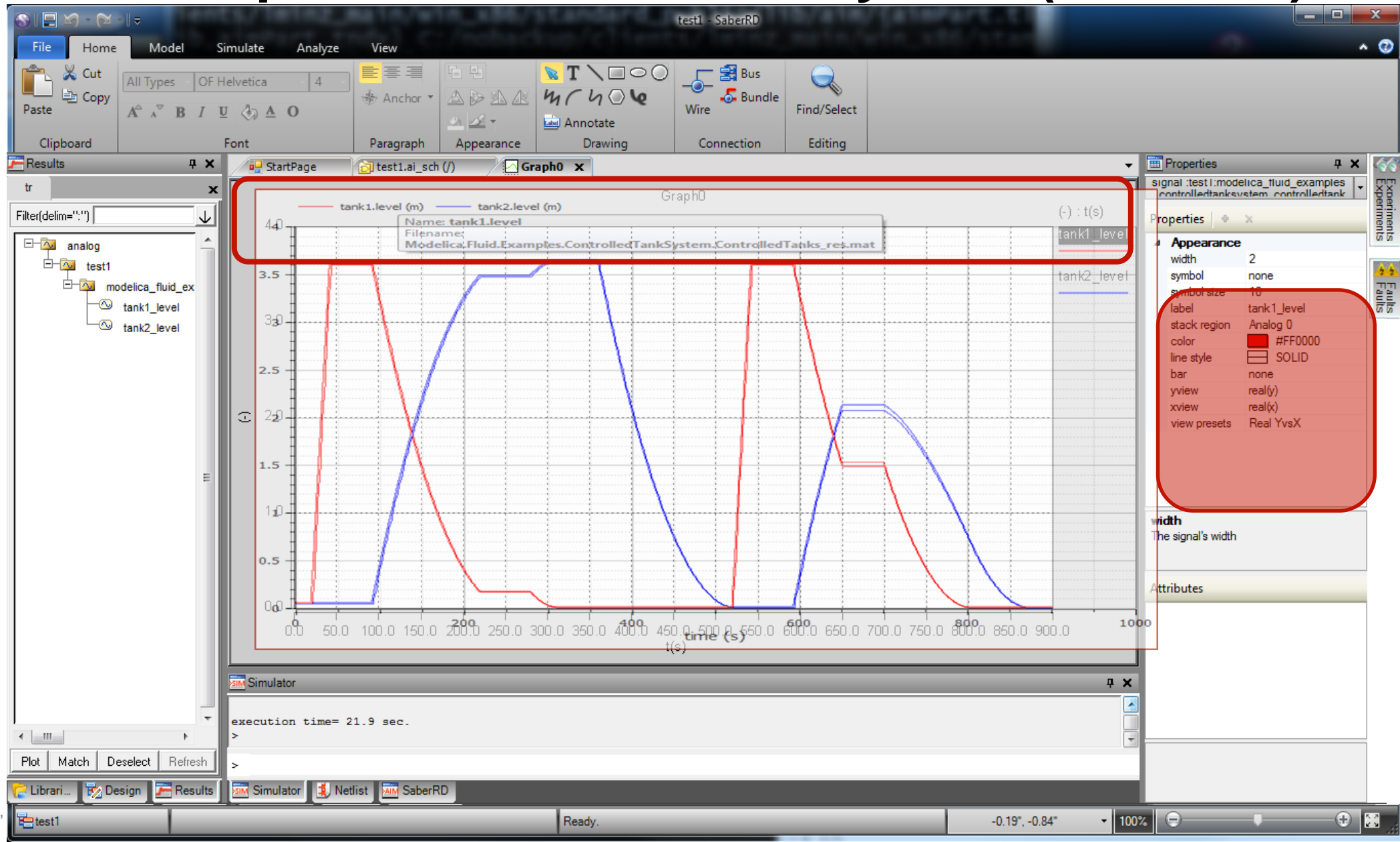
Other Examples (Three Phase Rectifier)



Other Example: Controlled Tank System (OpenModelica)



Other Example: Controlled Tank System (saberRD)



Thank You

