SYNOPSYS®
Silicon to Software™

A Method to Import an FMU to a Hardware Description Language

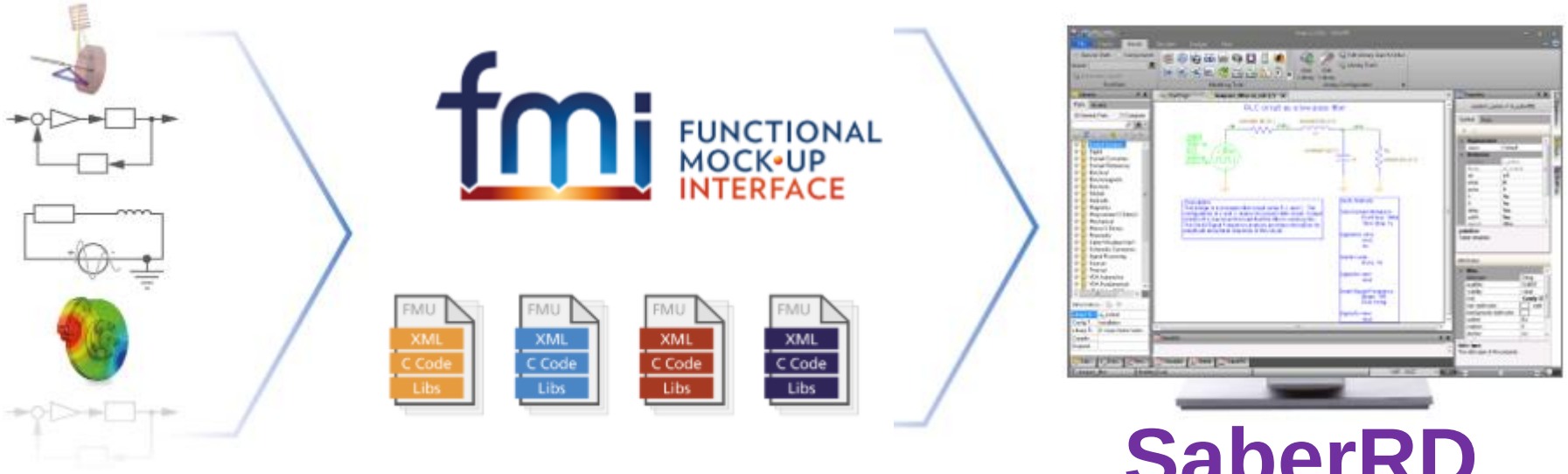
Min Zhang

Agenda

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

SYNOPSYS®

Introduction FMI/FMU



http://fmi-standard.org

SYNOPSYS®

Import an FMU to an HDL(MAST)

- MAST Modeling Language
  1. Invented in earlier of 80’s
  2. Multi-domain technolody
  3. Mixed-Mode signal
  4. Used in automible 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)

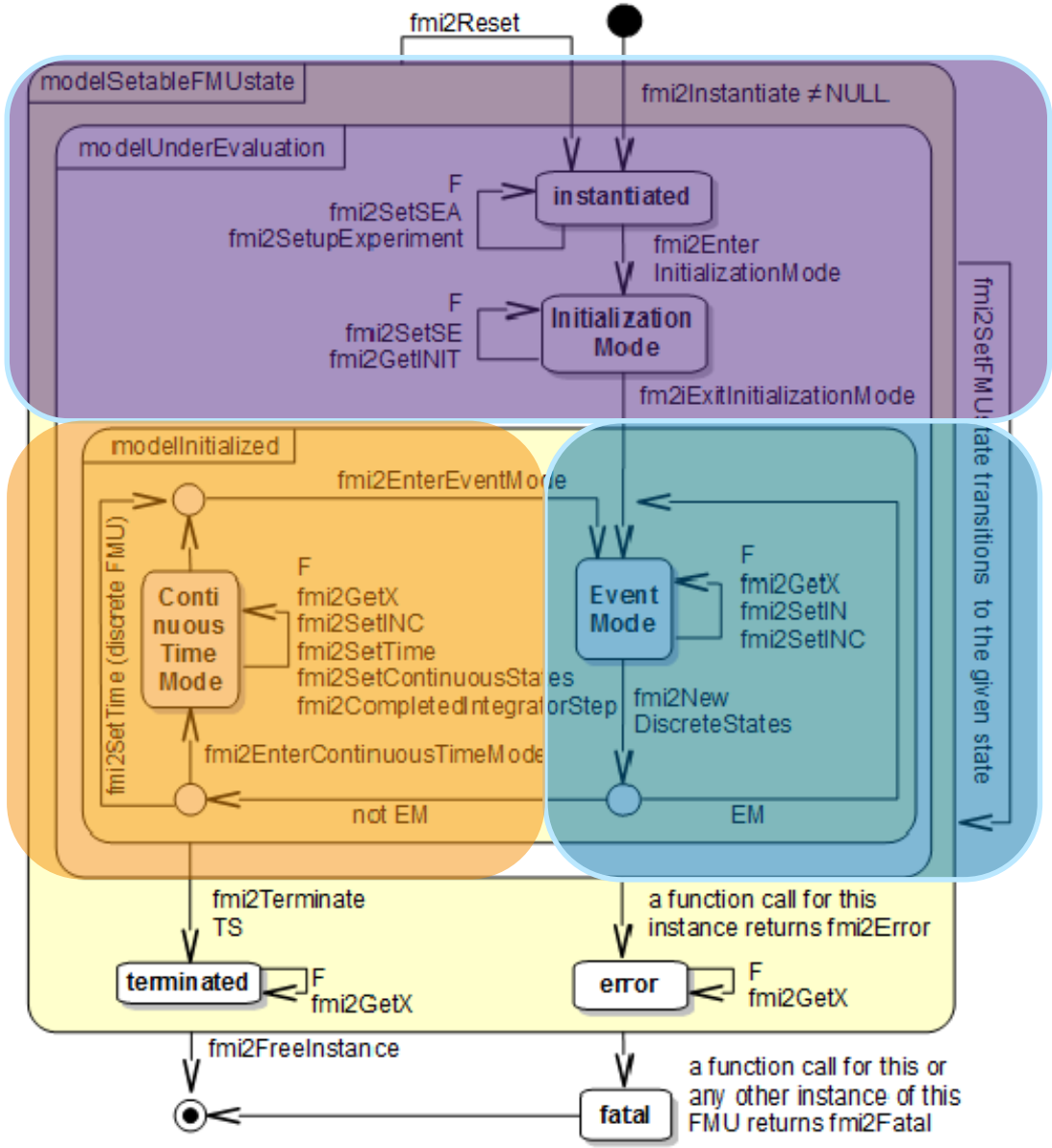
MAST				FMI 2.0		
				causality	**variability**	**derivative**
connection		input		input	continuous	
		output		output	continuous	
	state	(in)		input	discrete	
	state	(out)		output	discrete	
variable		Parameter		parameter	fixed	
		Constant		calculatedParameter	fixed	
				local	fixed	
				local	constant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

SYNOPSYS®

FMI/FMU vs MAST (Simulation flow)

Elaboration: Parameters Section

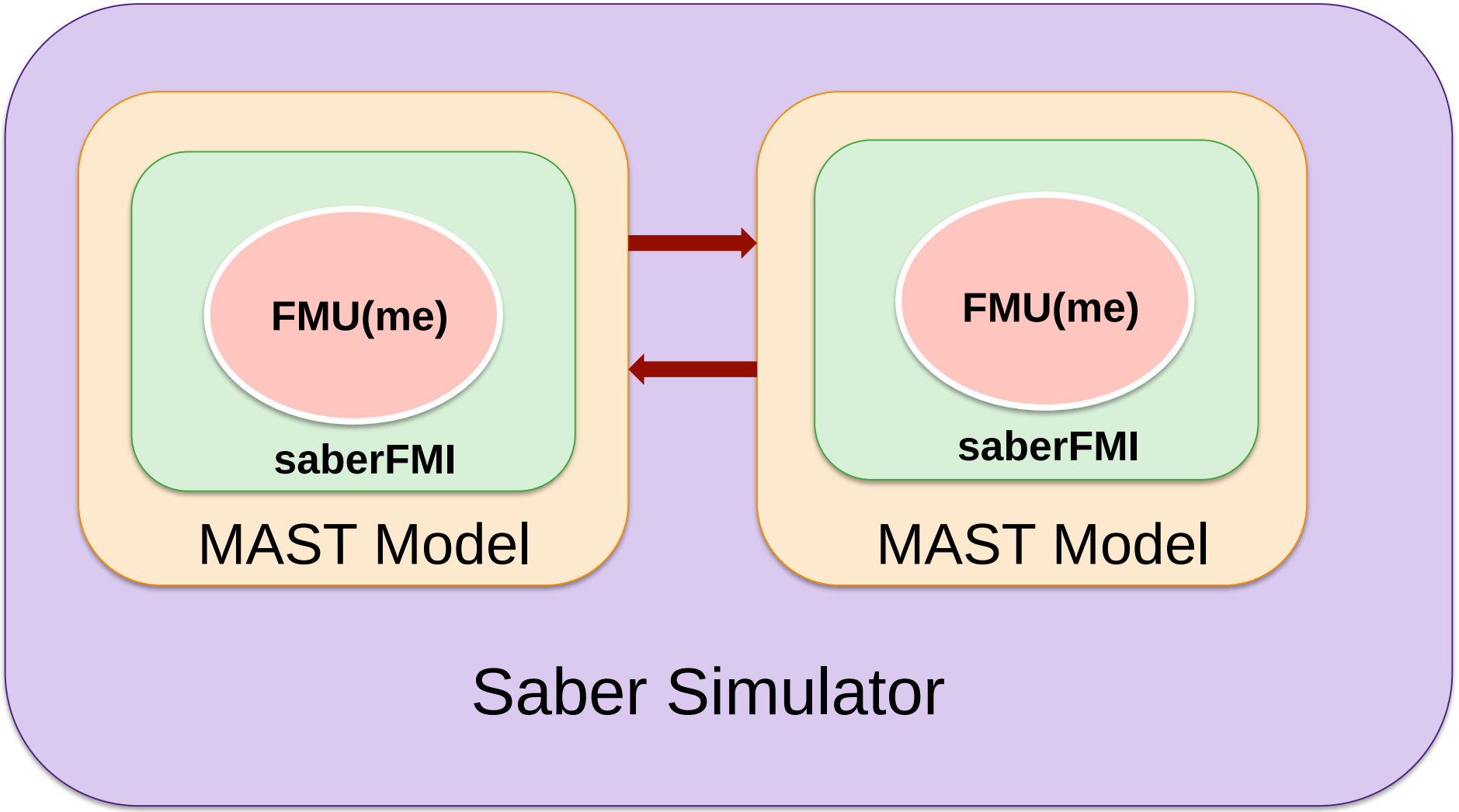
Analog Engine: Values + Equations Section



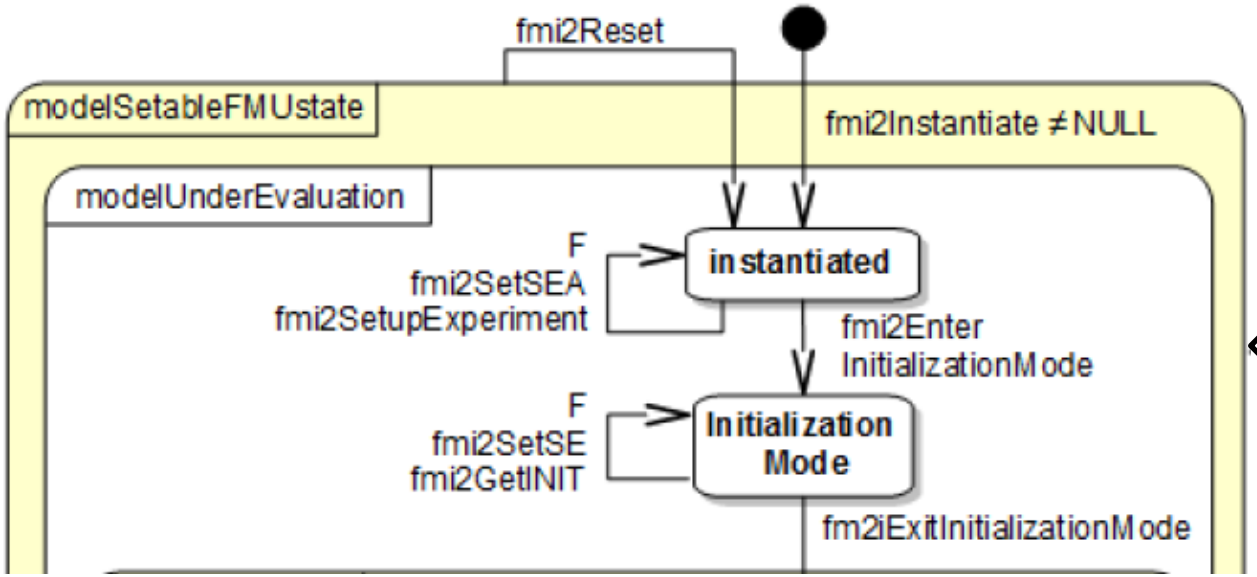
Digital Engine: When Sections

SYNOPSYS®

Interface between MAST and FMI



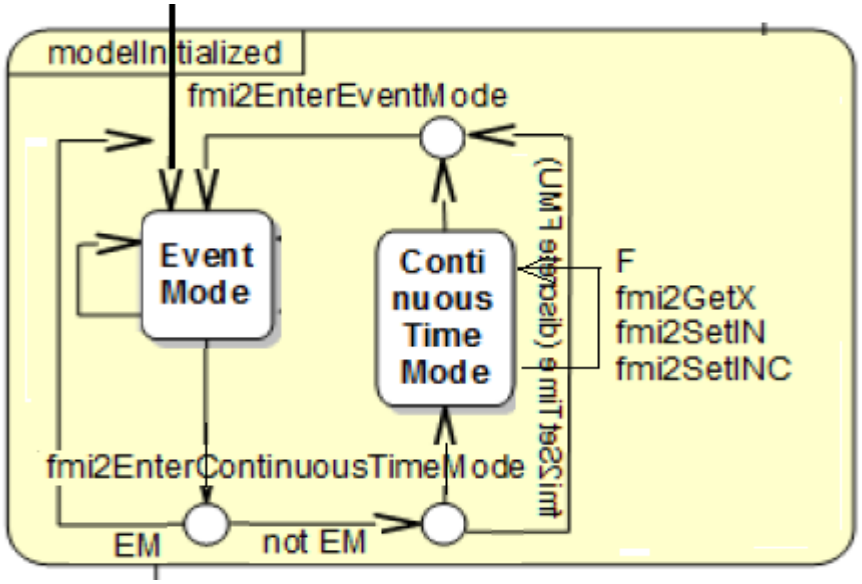
Parameter Section

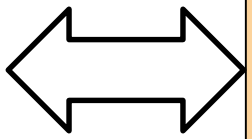



```
Parameters {
    fmiHandle = saberfmi(initialization…) {
        fmi2Instantiate();
    }
    saberfmi(setValues,fmiHandle,…) {
        fmi2setReal/Integer/Boolean();
    }
    saberfmi(updateValues,fmiHandle…) {
        fmi2EnterInitializationMode();
        fmi2ExitInitializationMode();
    }
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


SYNOPSYS®

Values Section

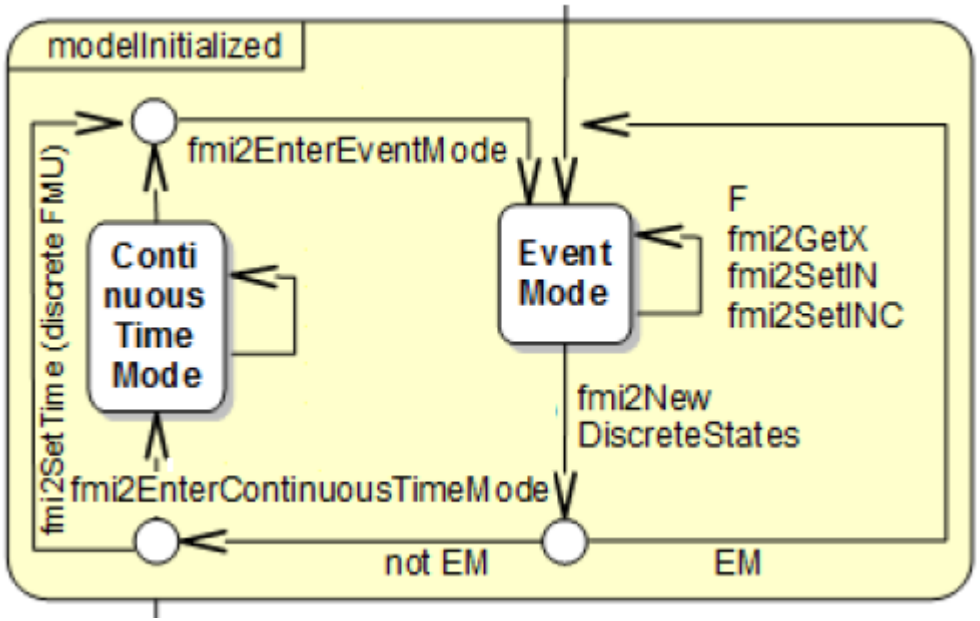




```
Values {
    saberfmi(timeValue,fmiHandle,input…) {
        fmi2setReal ();
    }
    xIndicator=saberfmi(checkCrosss,fmiHandle,time…) {
        fmi2getEventIndicators();
     }
     der_h=saberfmi(nonlinear,fmiHandle,time,h,v) {
         fmi2setReal();
         fmi2completedIntegratorStep();
         fmi2getReal();
     }
}
Equations {
    h : d_by_dt(h) = der_h
    v0: d_by_dt(v0) = der_v
    v : v = v_init + delta_v0
}
```


SYNOPSYS®

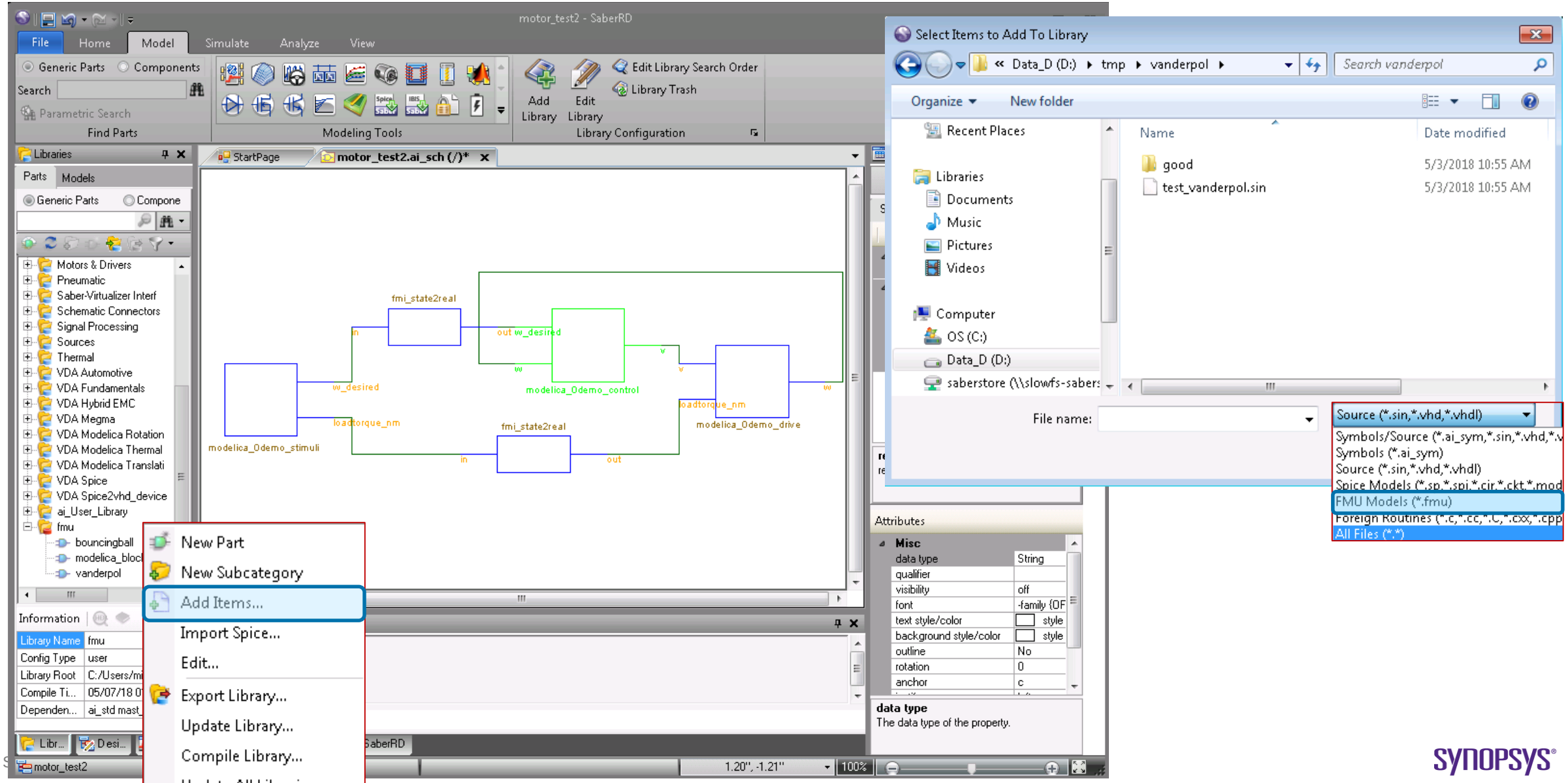
When Sections




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


SYNOPSYS®

FMU2MAST Conversion



SYNOPSYS®

FMU2MAST conversion

- Verified by 44 different tests, 5 different tools.

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

BouncingBall Example (Accurate Event Detection)

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

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

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

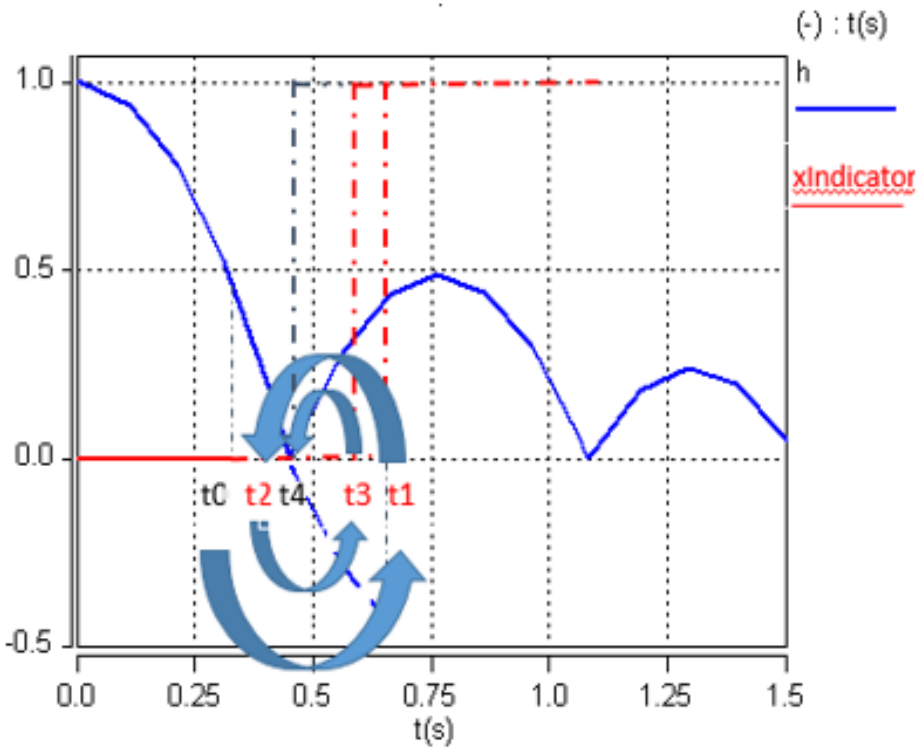


Figure 8 Threshold Crossing Detection

```
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)

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

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

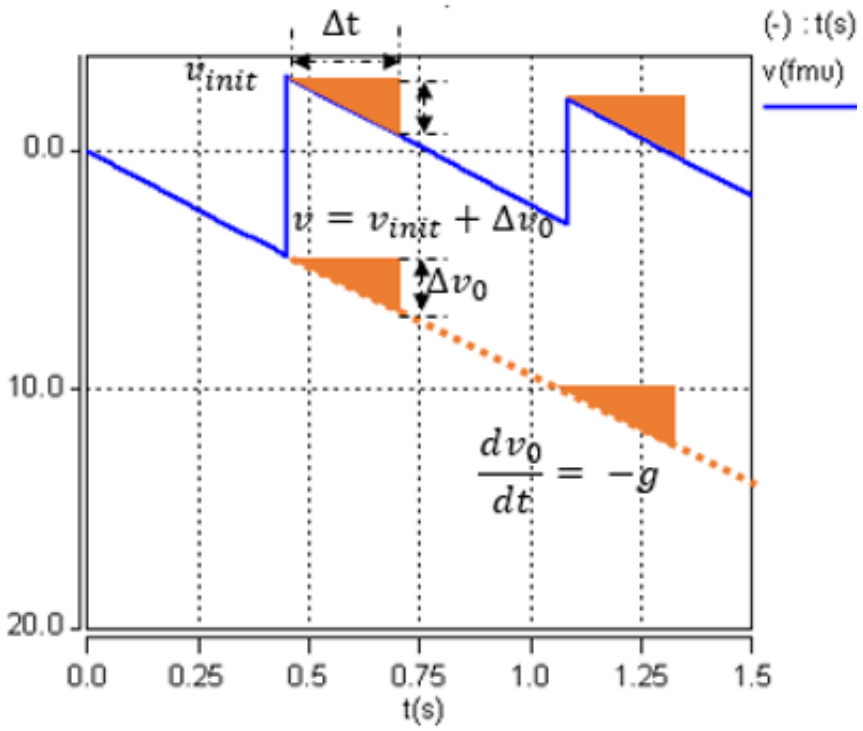


Figure 9 Solution for State Variable with reinit

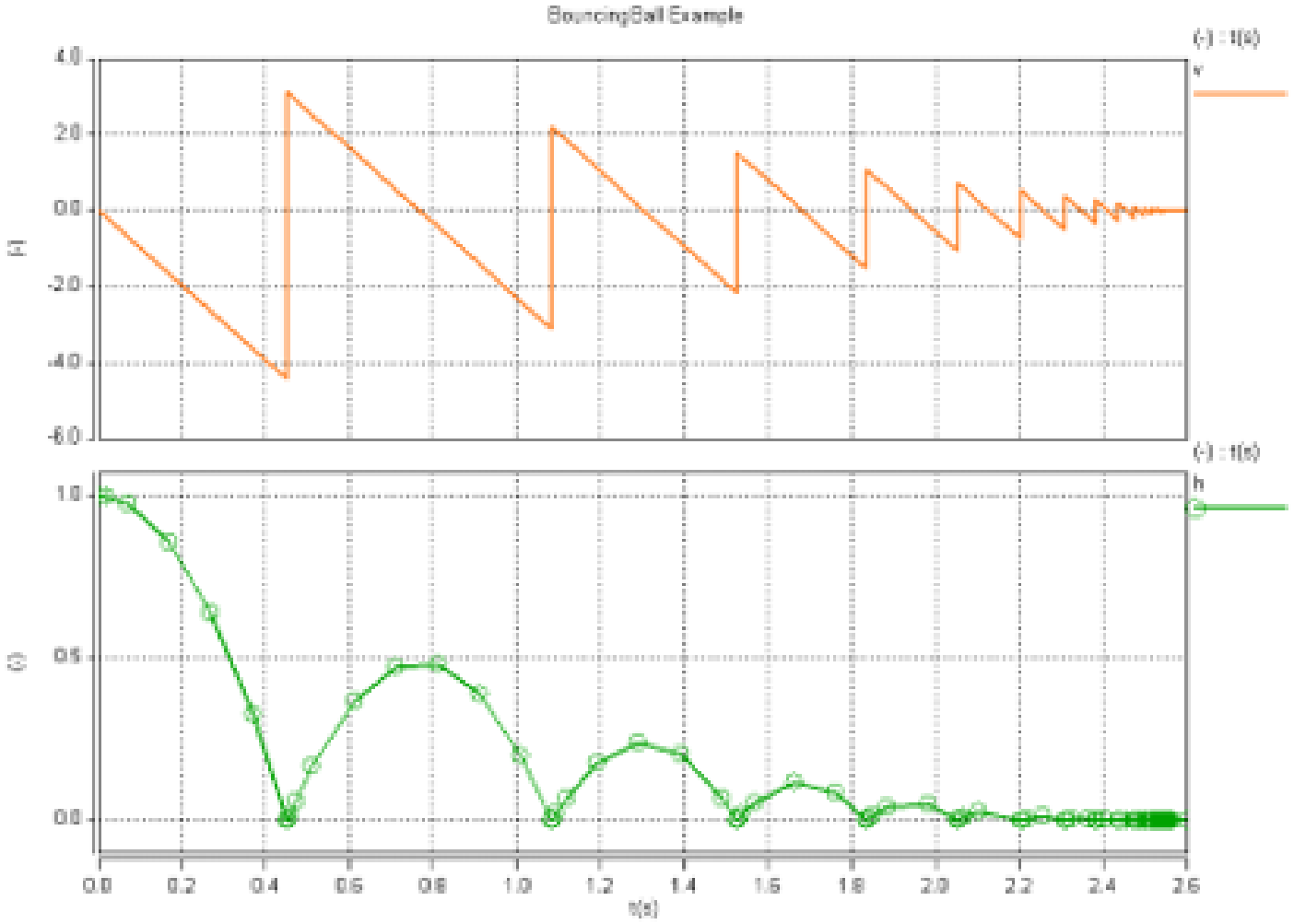
```
values {
   xIndicators = saberfmi(checkCross, fmiHandle, time)
   delta_v_0 = v_0 – prev_0_0
}

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)
}
```


SYNOPSYS®

BouncingBall Example (Results)



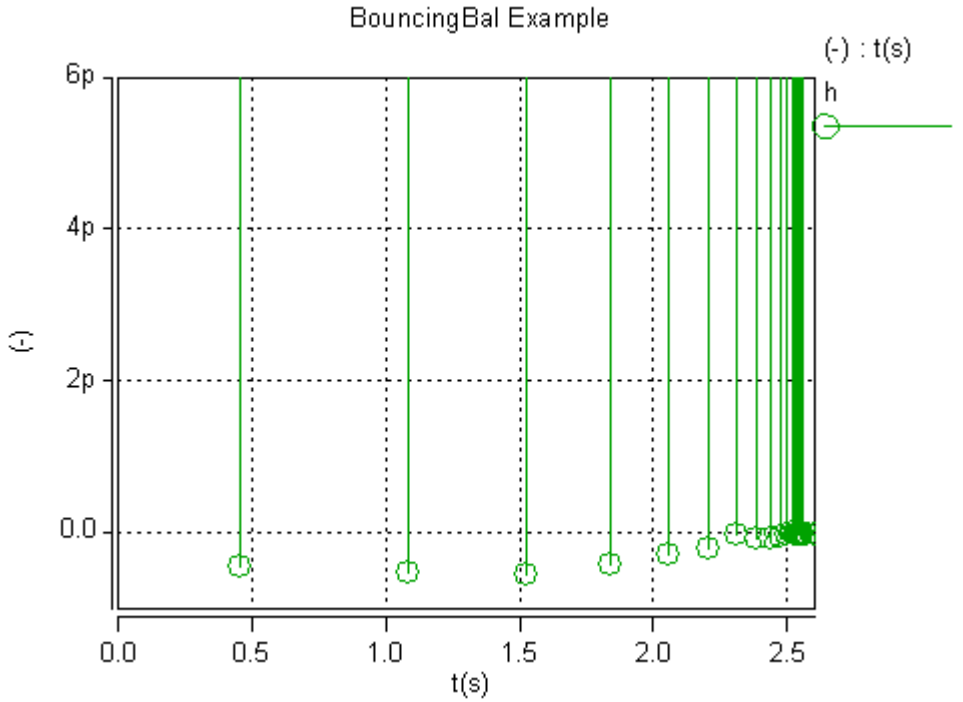


Figure 10 Bouncing Ball Results

SYNOPSYS®

Motor Drive Example (Equations)

$$\begin{cases} \dfrac{dpi_x}{dt} = k_1 * \omega_i + k_2 * \omega_o \\ v = k_3 * \omega_i + k_4 * \omega_o + k_5 * pi_x \end{cases}$$

$$\begin{cases} \omega_i = u_1(t) \\ T_i = u_2(t) \end{cases}$$

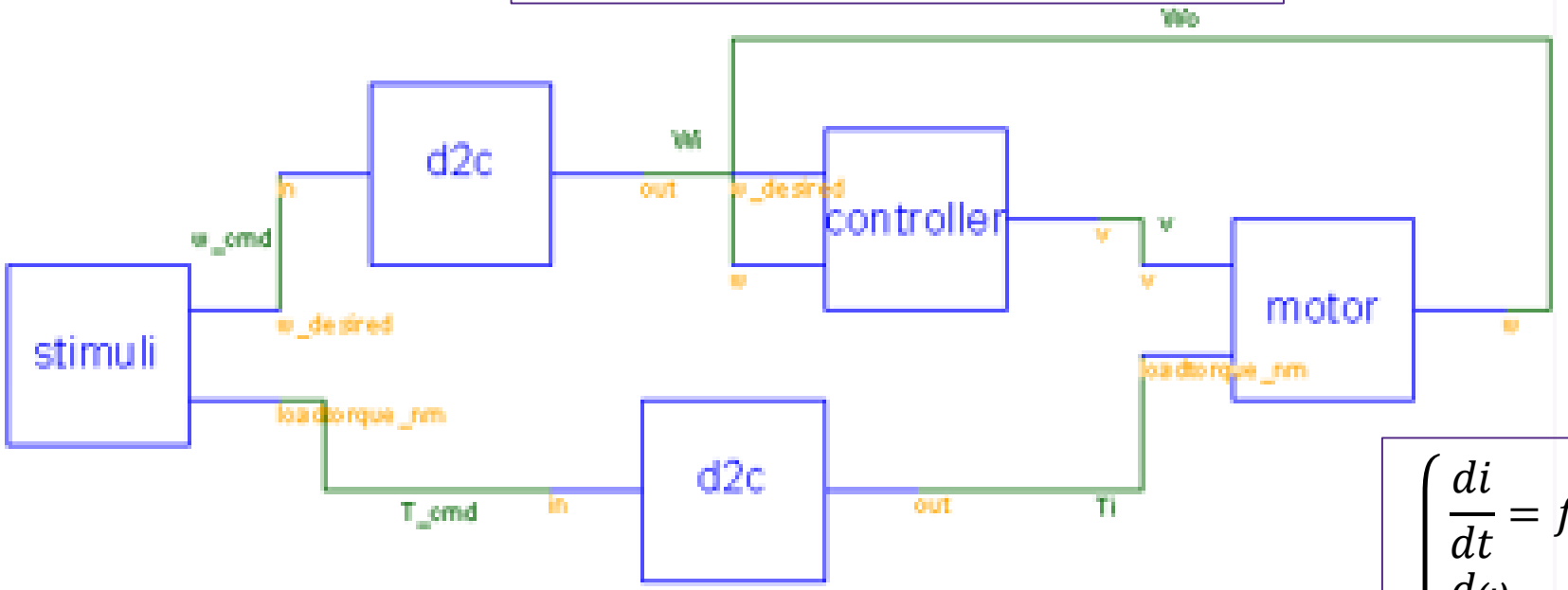


Figure 11 Motor Drive Schematic

$$\begin{cases} \dfrac{di}{dt} = f_1(v, i, \omega) \\ \dfrac{d\omega}{dt} = k_6 * i + k_7 * T_i \\ \dfrac{d\theta}{dt} = \omega \\ \omega_o = k_8 * \omega \end{cases}$$

SYNOPSYS®

Motor Drive Example (DAE formulation)

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

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

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & 0 & 0 & 0 & 0 & 0 & 0 & -k_2 \\ -k_3 & 0 & -k_5 & 0 & 0 & 0 & 0 & -k_4 \\ 0 & 0 & -\frac{\partial f_1}{\partial v} & \frac{\partial}{\partial i} & 0 & -\frac{\partial f_1}{\partial \omega} & 0 & 0 \\ 0 & -k_7 & 0 & 0 & -k_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_8 & 0 & 1 \end{vmatrix} \cdot \begin{pmatrix} \omega_i \\ T_i \\ pi \\ v \\ i \\ \omega \\ \theta \\ \omega_o \end{pmatrix} + \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{pmatrix} \dot{\omega}_i \\ \dot{T}_i \\ \dot{pi} \\ \dot{v} \\ \dot{i} \\ \dot{\omega} \\ \dot{\theta} \\ \dot{\omega}_o \end{pmatrix}$$

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

SYNOPSYS®

Motor Drive Example (Results)

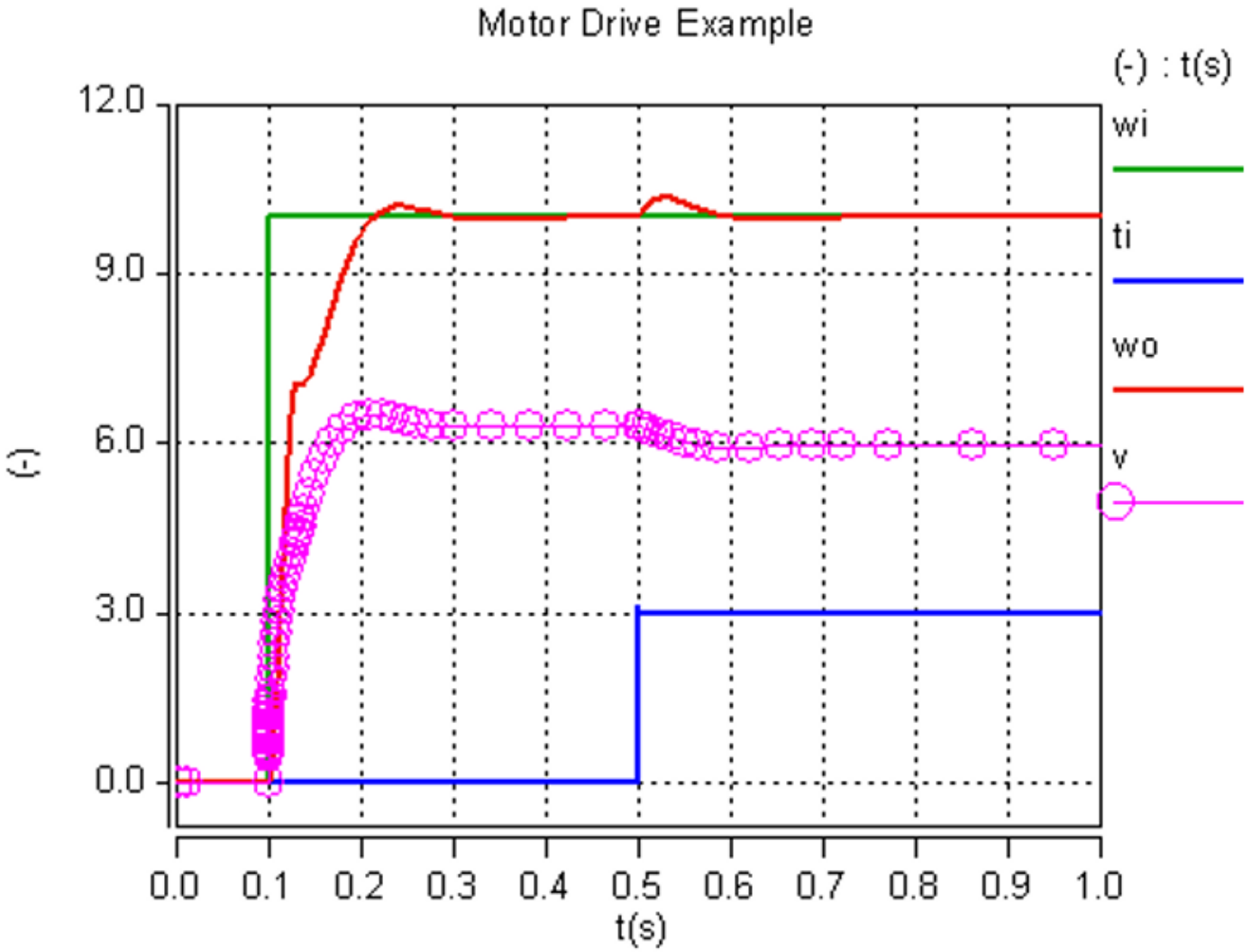
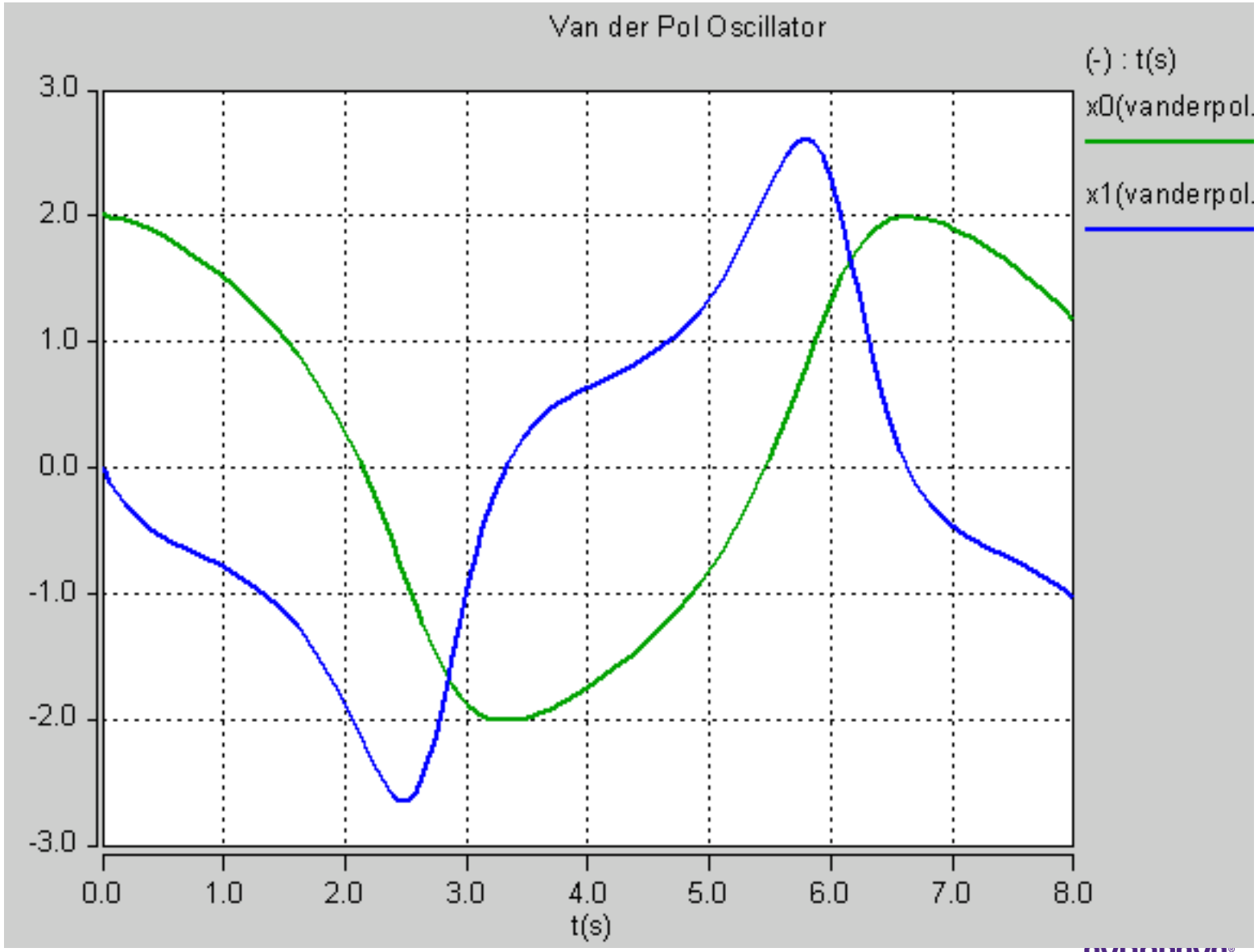


Figure 12 Motor Drive Results

SYNOPSYS®

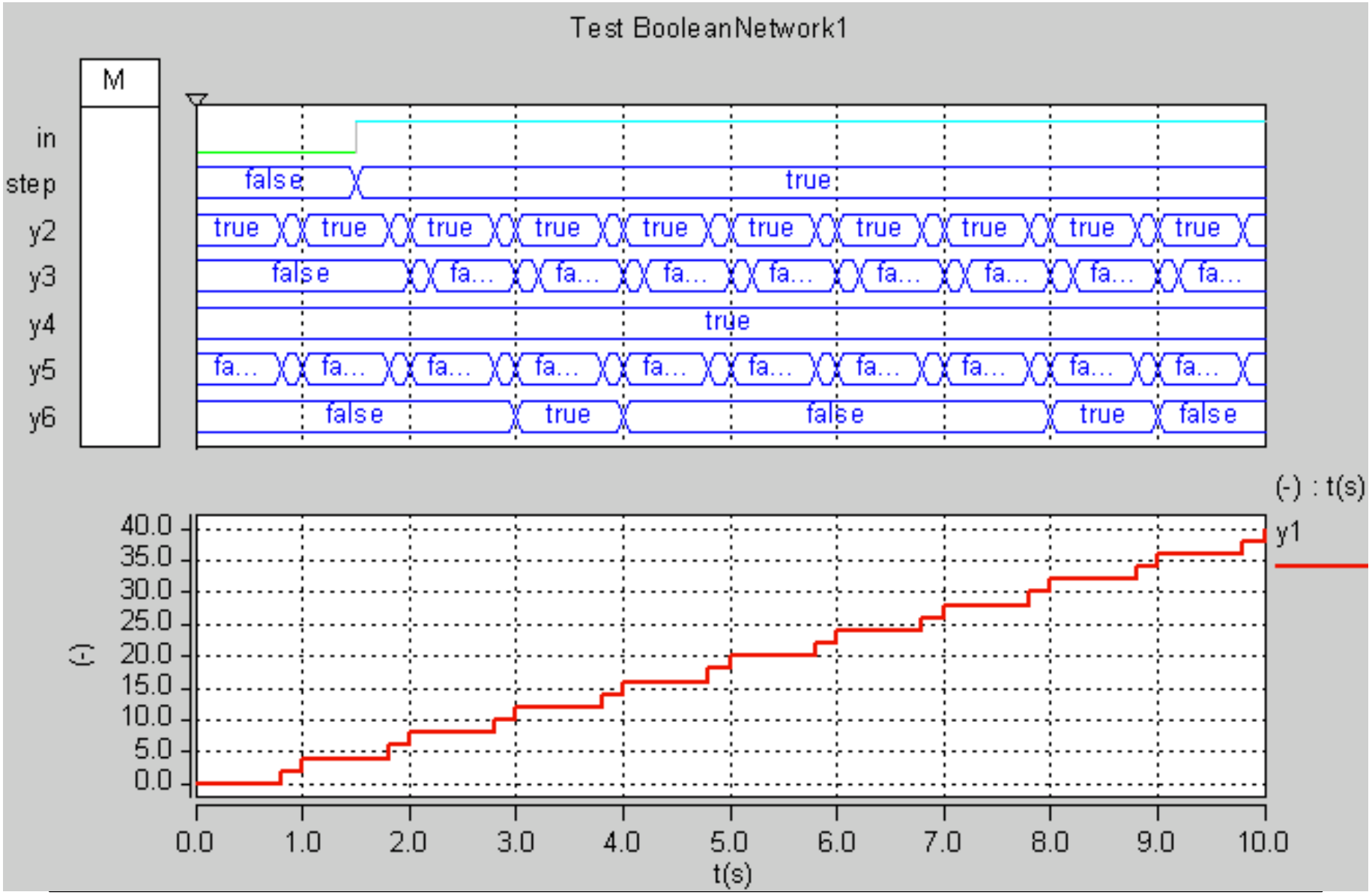
Other Examples (Van der Pol oscillator)

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



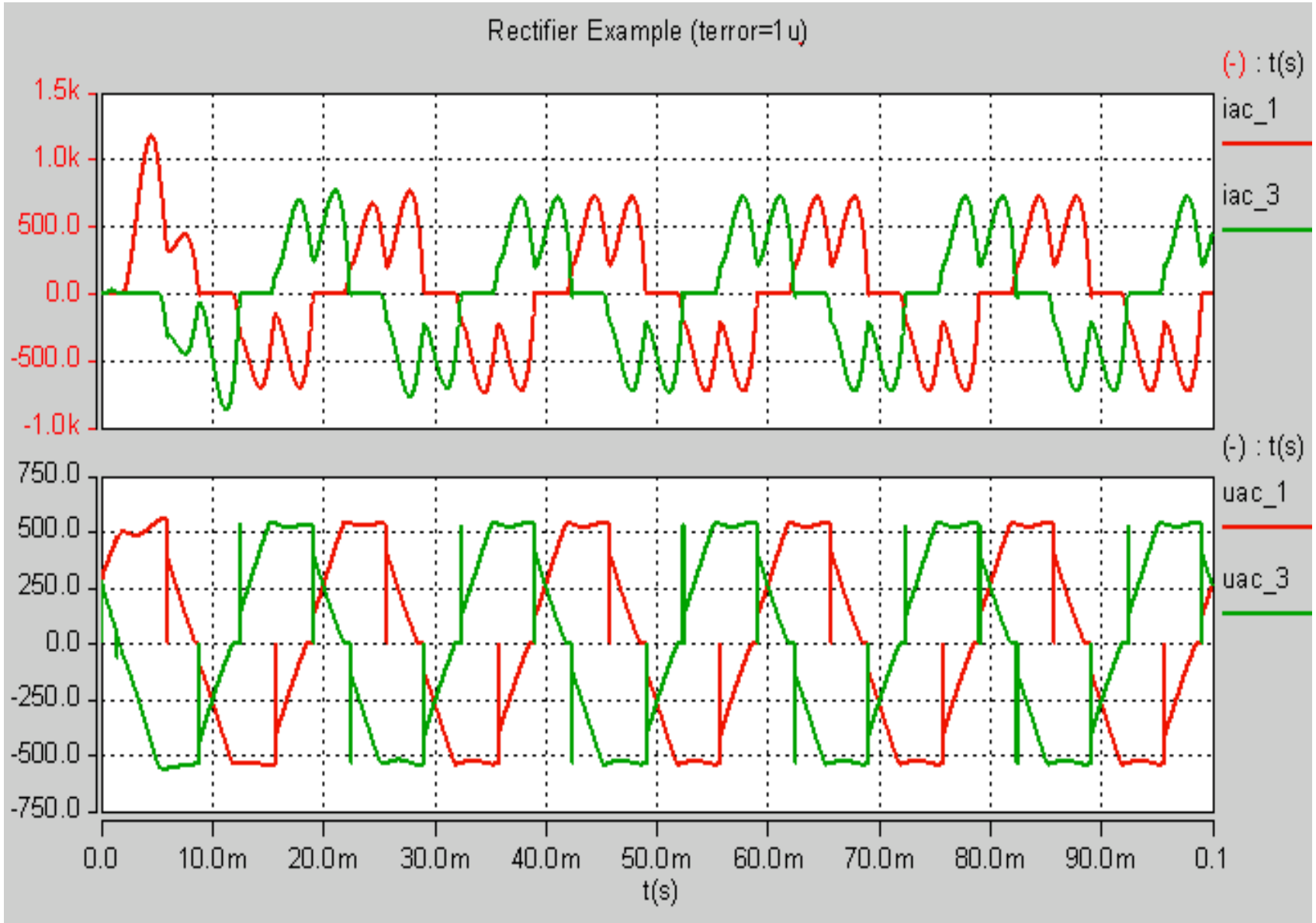
SYNOPSYS®

Other Examples (booleanNetwork1)



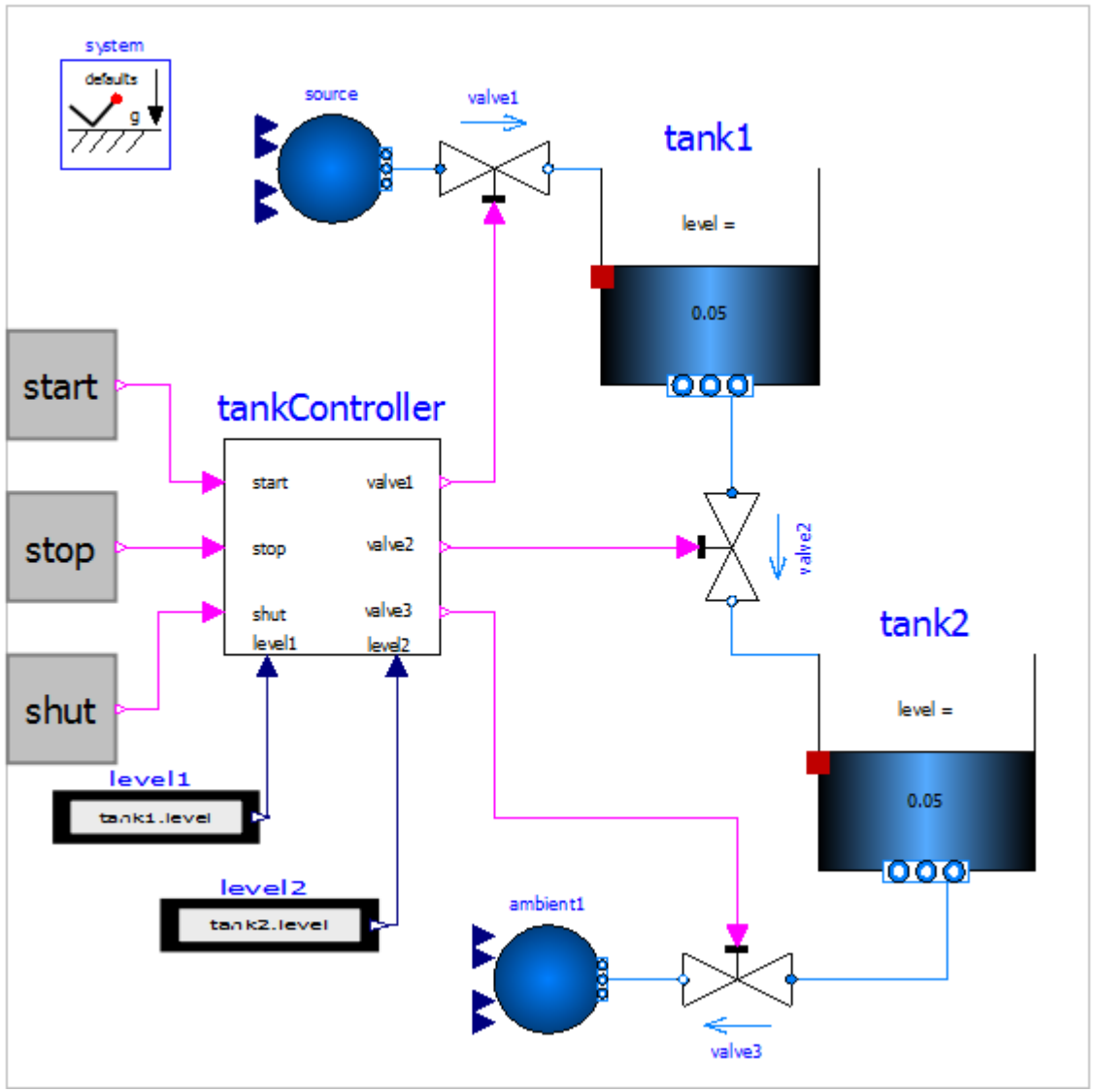
SYNOPSYS®

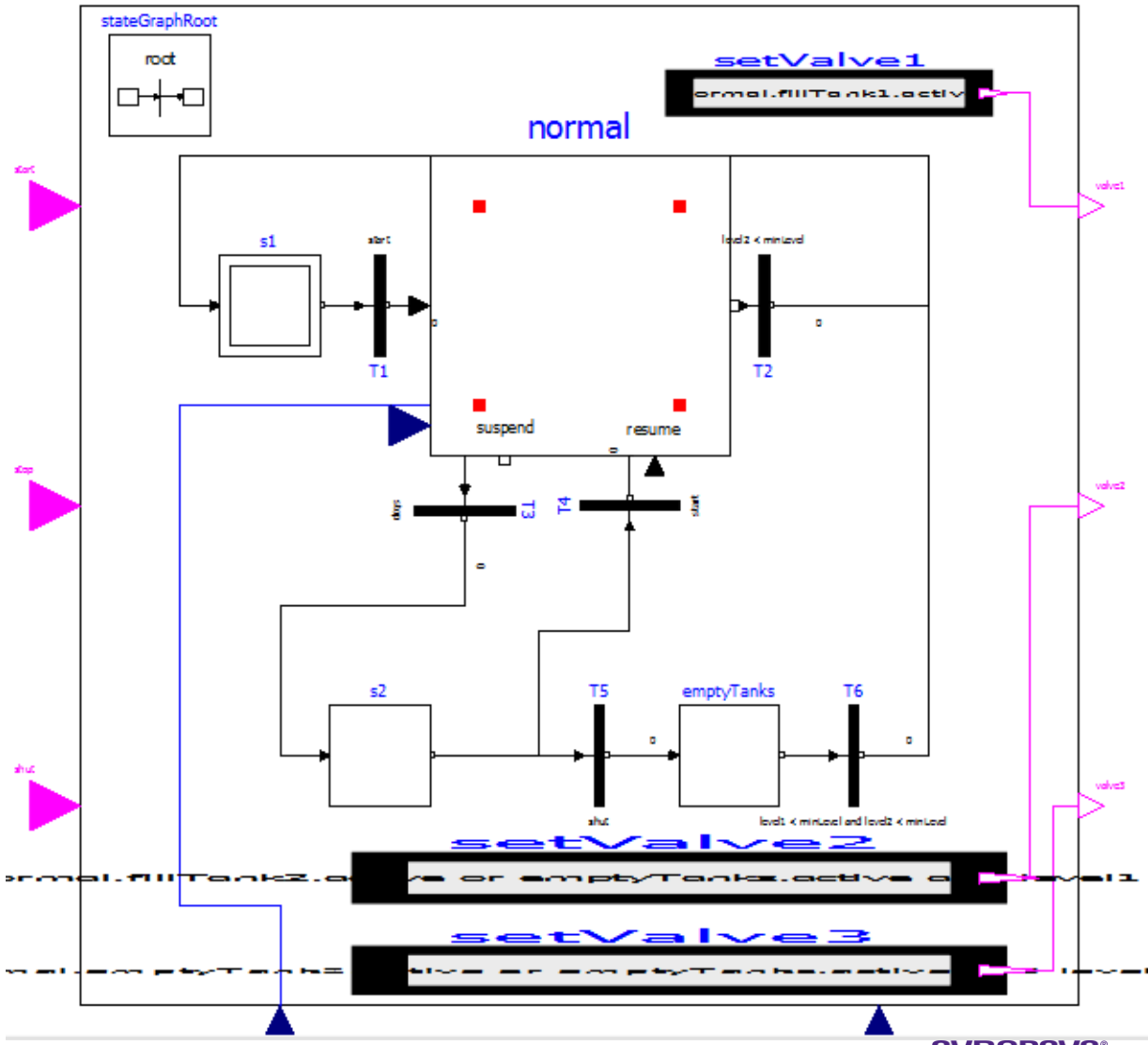
Other Examples (Three Phase Rectifier)



SYNOPSYS®

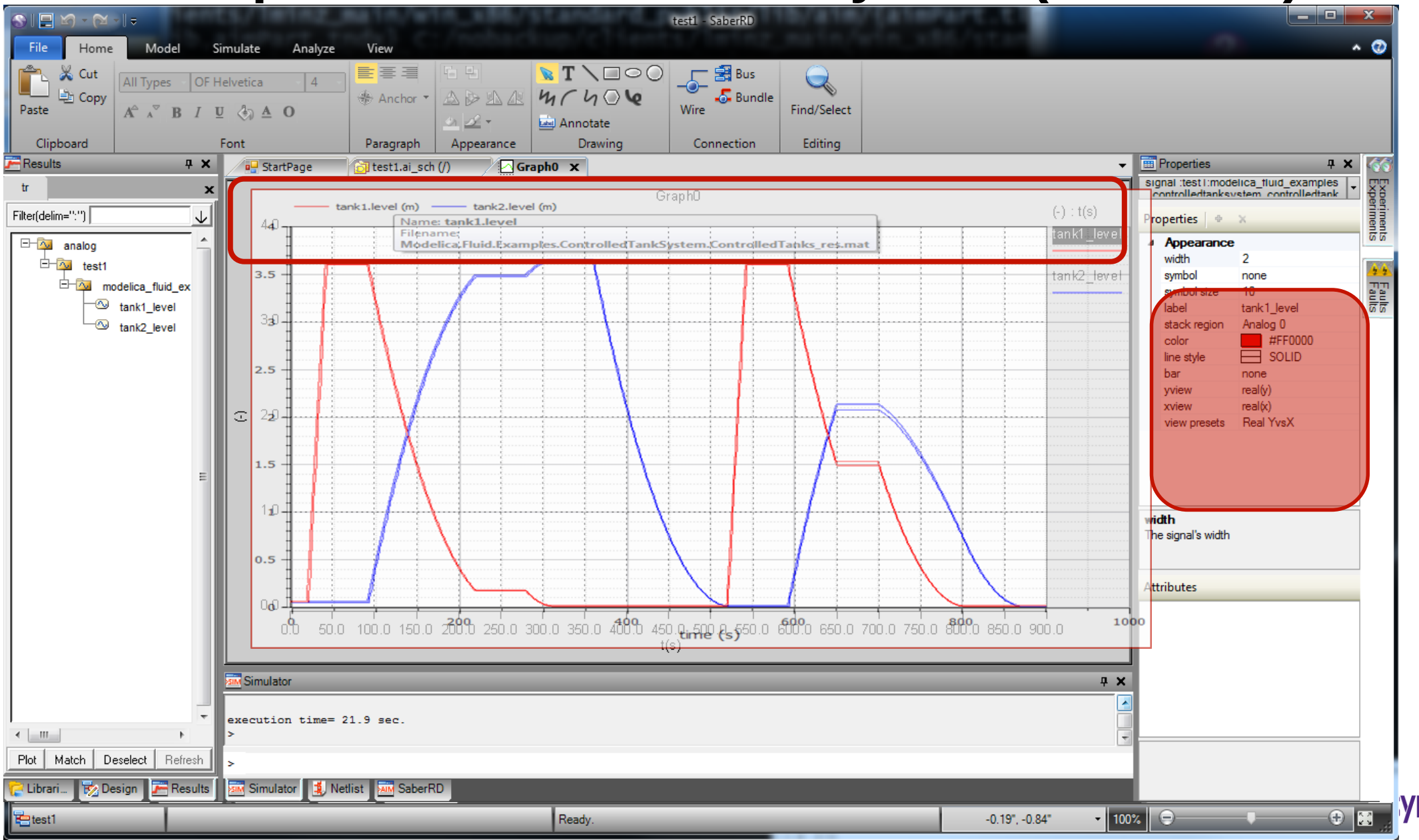
Other Example: Controlled Tank System (OpenModelica)





SYNOPSYS®

Other Example: Controlled Tank System (saberRD)



SYNOPSYS®
Silicon to Software™

Thank You