The much more elaborate process control model PC(C) (Fig. 8) incorporates all six phases, pictured in Figure 2, as well as possible repetition of the heat-up and hold phases. According to the theory in [8], it is a hybrid DEV&DESS model. It is modeled based on the state machine approach, but it includes continuous state event handling and has been mainly implemented using *Stateflow*. The two auxiliary subsystems (*schedTEvent, checkTemperature*) create events when the waiting time has changed or the heat-up or cool-down temperature has been reached. The model PC(C) is essentially an adapted version of a model described in [9].
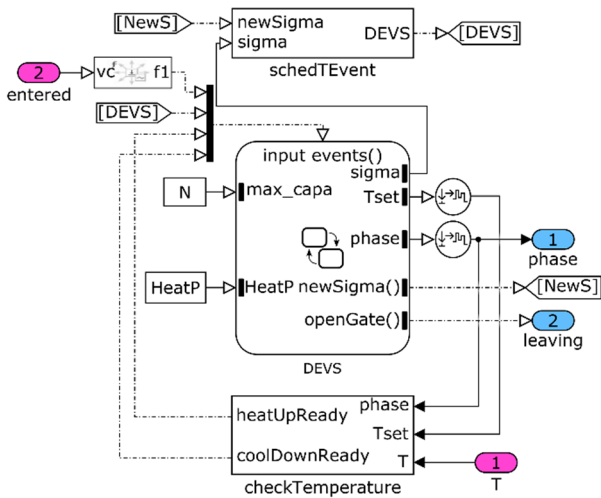

Figure 8: Discrete event-based model with continuous state event handling of PC(C) mainly using *Stateflow*.

4.3 Process physics models

The basic process physics model PP(A) (Fig. 9) uses a simple power balance to compute the change of the oven temperature T_0:

$$C_0 \dot{T}_0 = P_{heat} - P_{loss} \tag{1}$$

where C_0 is the total heat capacity of the oven. The power loss is computed with Newton's simple law of cooling:

$$P_{loss} = k_a(T_0 - T_e) \tag{2}$$

where T_e is the temperature of the surroundings and k_a a constant that subsumes all convective and conductive processes. The heating power is assumed to be constant during the heat-up phase and to match the losses during the hold phase:

$$P_{heat} = \begin{cases} P_H & heat-up \\ k_a(T_0 - T_e) & hold \end{cases} \tag{3}$$

For the computation of the total power demand of the furnace, a constant power P_0 is added, which subsumes all non-heating processes, such as a base load or the power electronics.
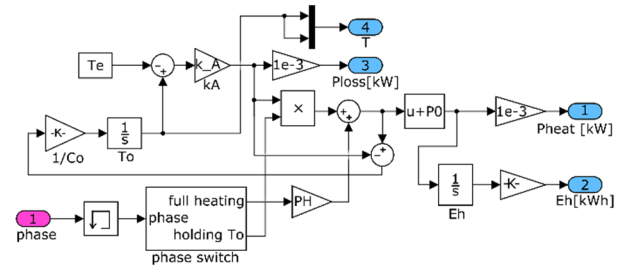

Figure 9: ODE-based model of PP(A) using Simulink.

The model PP(C) (Fig. 10) uses physical modeling based on *Simscape* to incorporate much more physical details, such as convection and radiation from the furnace – during load and unload phases – to the environment. The governing differential algebraic equations are not built up explicitly here; instead, the physical components such as heat capacities and various kinds of heat flow are represented directly, which makes the physical structure of the model much clearer.
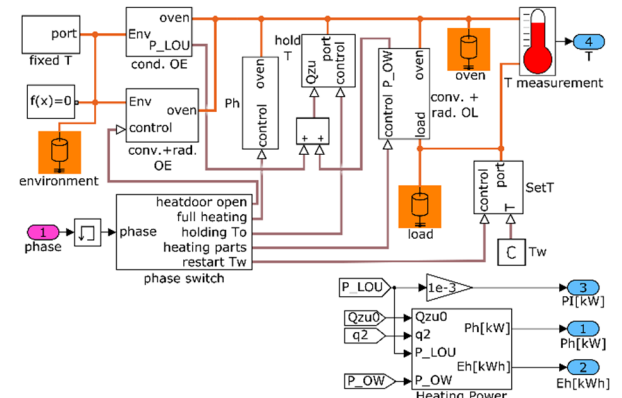

Figure 10: Physical model (DAE) of PP(C) using *Simscape*.

4.4 Implementation pitfalls

The extension of *MATLAB/Simulink* by several packages (blocksets) is necessary to make a multi-paradigm model possible, but it also leads to small inconsistencies, which have to be overcome. A minor nuisance here is the large number of different ways used to express values and signals in the packages used: Simple scalar values can be time-based in *Simulink*, event-based in *SimEvents* or physical-valued in *Simscape*. An event can be defined by a sample time hit, a rapid value change (*edge*), a special trigger signal or a function call.