Simulation essentials with PLECS

PN137 | Posted on March 21, 2021 | Updated on August 8, 2025

Stéphane LOVEJOY
Senior Software Developer
imperix • in

Table of Contents

This note provides in-depth content for an accurate and efficient offline simulation of an imperix controller and the corresponding plant model using ACG SDK on PLECS.

A general overview of software-related notes is given on this page.

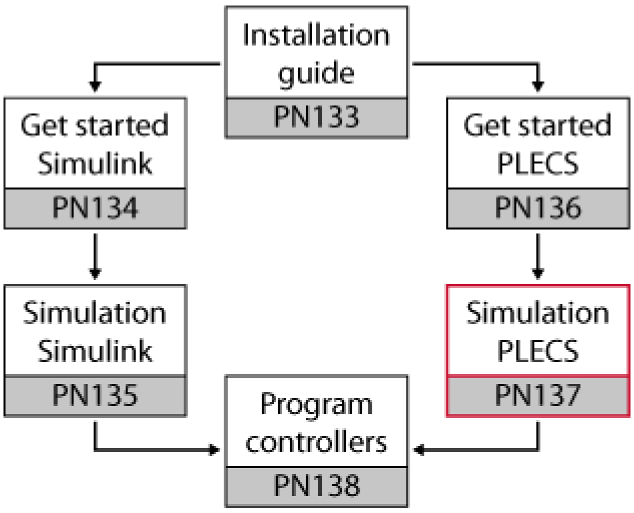
Related notes

Suggested prerequisites

- Installation guide for ACG SDK
- Getting started with PLECS

Suggested further readings

- Programming and operating imperix controllers
- Cockpit – User guide



Fundamental concepts

The offline simulation is meant to reproduce as faithfully as possible the behavior of the overall system (controller and plant). To that end, the PLECS blockset (part of the ACG SDK) was designed with the following guidelines in mind:

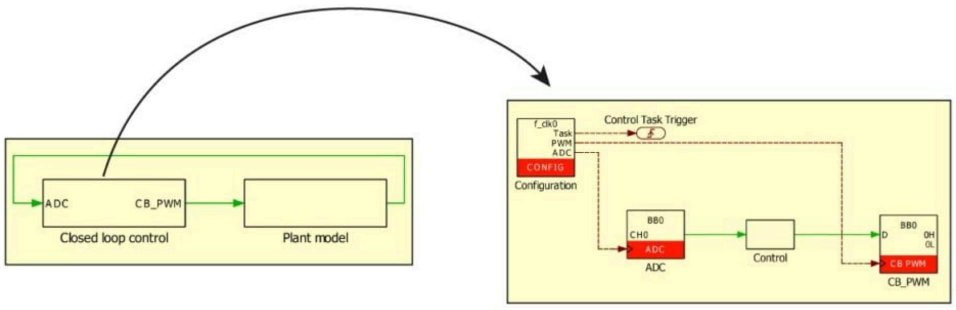
- The **plant** quantities are to be modeled with **continuous** signals
    - This requires a **variable-step solver**
- The **control** algorithm is modeled with **discrete** signals, sampled at a rate equal to the interrupt frequency, and with a phase corresponding to the sampling phase.
    - The "Control Task Trigger" block used within an atomic control subsystem forces the discretization of the control algorithm to the interrupt frequency
    - This is modeled accurately with the variable-step solver, as it is forced to take a major step at least at each interrupt execution
- The behavior of the real PWM generators is modeled, in particular:
    - The frequency and phase of the carrier
    - The instants when the duty-cycle and phase parameters are updated (at zero and/or max of the carrier)
- The duration of the algorithm execution is modeled
    - This induces a delay between the start of the interruption and the availability of the new data

With all the phases and delays modeled accurately, the simulated controller has the same dynamics as the real controller, which allows tuning the control parameters during the offline simulation.

The imperix blockset is already implemented in a way that automatically follows these guidelines.
As such, no particular action is required from the user.

Working principle of the main blocks

The four fundamental blocks are the 1) Config, 2) Control Task Trigger, 3) ADC, and 4) PWM blocks. Most applications can work with those four blocks only, as in the standard configuration shown below.



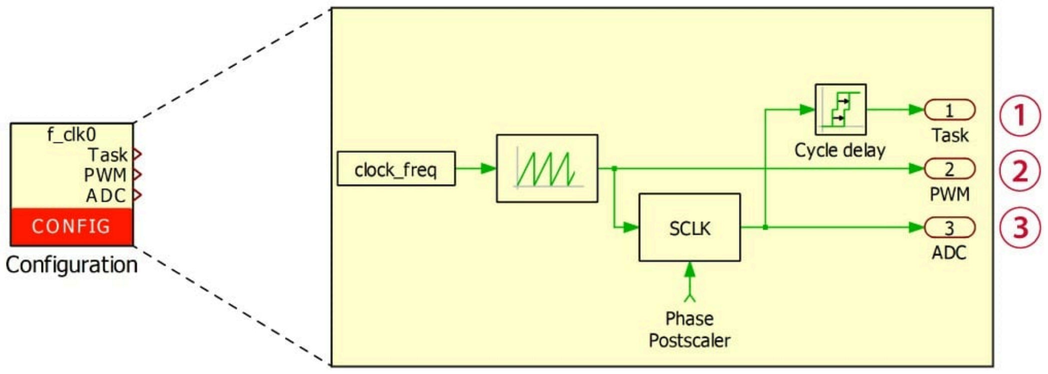
Typical content of the controller model

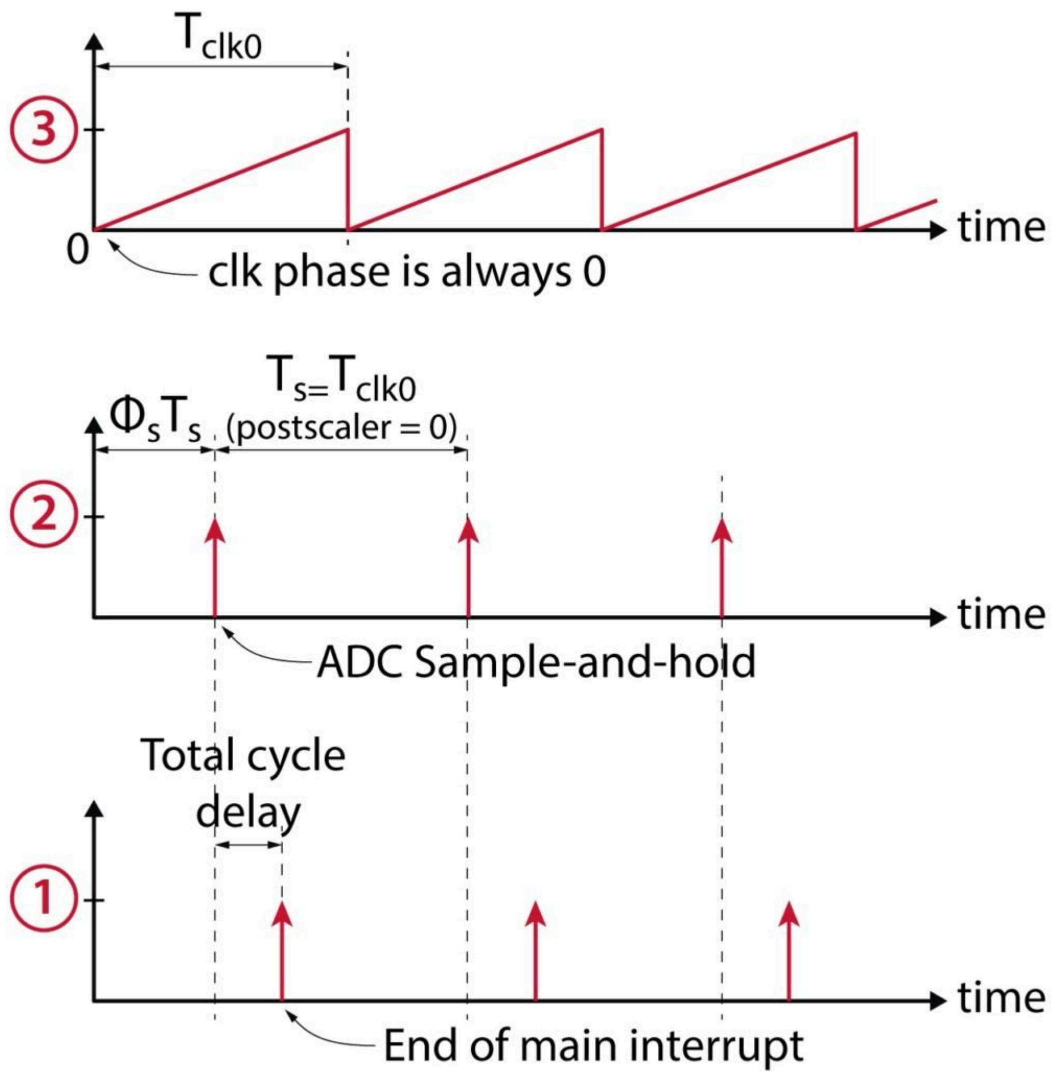
1) Configuration block (CONFIG)

The CONFIG block configures `CLOCK_0` frequency and oversees the configuration of the main interrupt frequency. `CLOCK_0` serves as a time reference for different peripherals such as ADC or PWM.

Four parameters are available to configure `CLOCK_0` and the execution of the main interrupt:

1. The frequency of `CLOCK_0`
2. The phase of the main interrupt
3. The postscaler of the main interrupt
4. The “total cycle delay” (explained below)





The CONFIG block outputs three different configuration signals (brown dashed wires):

- A main clock signal (`CLOCK_0`) with a period $T_{clk0} = 1/f_{clk0}$ and a phase of zero. This output can be connected to a PWM modulator block to define its switching frequency.
- A sampling clock, generated from `CLOCK_0` with the addition of the sampling phase and the postscaler parameters. This output can be connected to an ADC block to define its sampling instants. The sampling period is defined by:

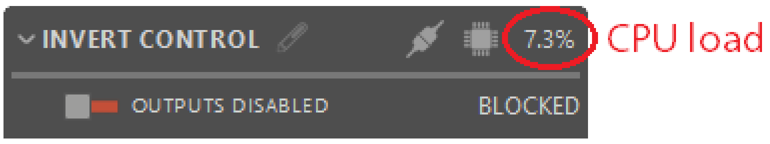
$$T_s = \begin{cases} T_{\text{clk0}} & \text{if postscaler} = 0 \\ 2 \cdot \text{postscaler} \cdot T_{\text{clk0}} & \text{otherwise} \end{cases}$$

- A control task signal, generated from the sampling clock, with the addition of the “total cycle delay” parameter. This corresponds to an “end of interrupt” signal and must be connected to the “Control Task Trigger” block. The addition of a cycle delay allows the simulation to correctly model the effect of the interrupt execution time and the different read/write delays, as explained below.

Total cycle delay

The *total cycle delay* is a parameter used in simulation only. It is introduced to simulate the delay from the moment the ADCs are sampled to the moment the calculated duty-cycle is propagated to the FPGA PWM management. It takes into account the ADC sampling time (e.g. 2 µs on a B-Box RCP), the control algorithm execution time, and the FPGA read and write transfer time.

The total cycle delay is displayed in the “Target info” pane of Cockpit (formerly Timing Info tab in BB Control) when a code is running (requires software above ACG SDK v3.6). In previous versions, an estimation of the total cycle delay can be derived from the CPU load displayed by BB Control, by multiplying it by the interrupt period.



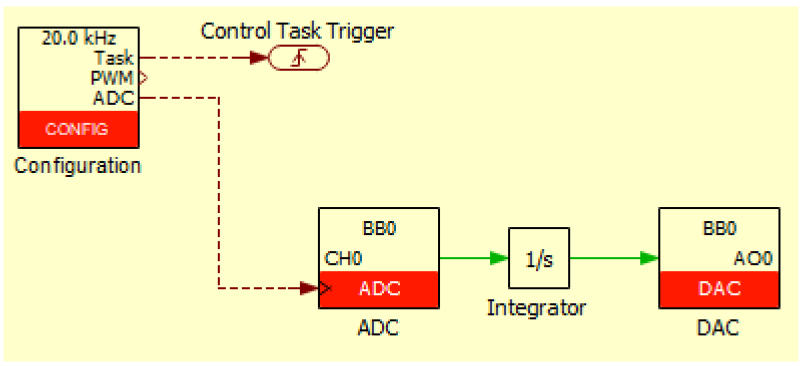
Display of the CPU load in imperix Cockpit

The PN142 provides more details on the various delays and their effect on the control dynamics.

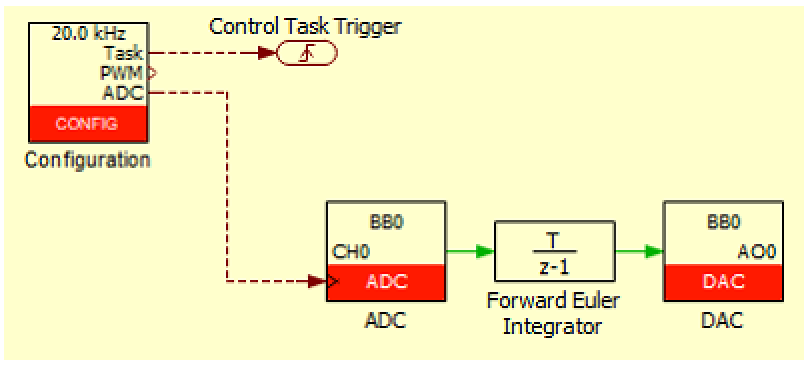
2) Control Task Trigger

The Control task trigger block is used to model the execution of the control algorithm within an interrupt routine. Therefore, the discretization of this subsystem and all block contained within are forced at the main interrupt frequency. The Forward Euler method is used to discretize continuous states within the subsystem (it forces the subsystem to be “atomic”).

As an example, the following two systems have strictly identical behaviors when the sample time of the discrete integrator and the Control Task trigger are set to `1/f_clk0`:



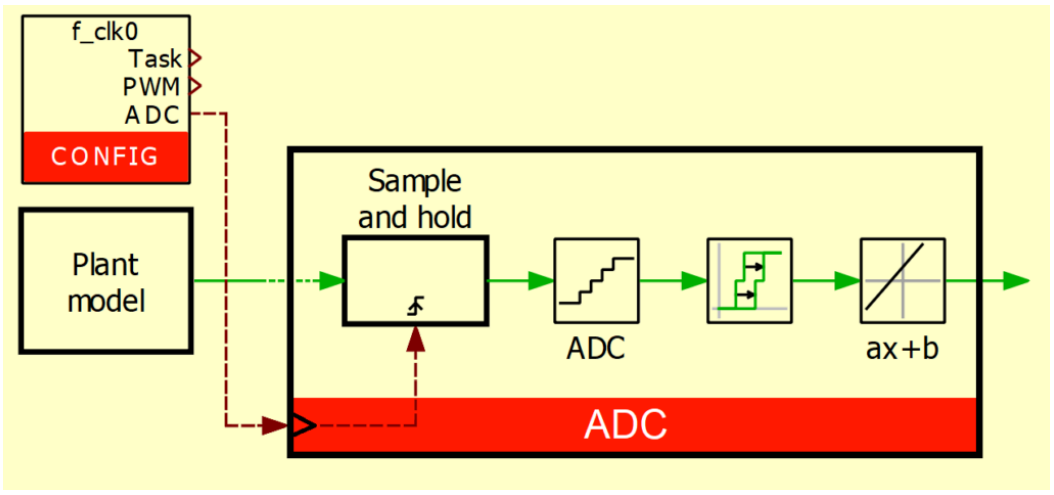
When continuous functions are used within an atomic subsystem containing a Control Task Trigger, its execution is discretized.



When continuous functions are used within an atomic subsystem containing a Control Task Trigger, its execution is discretized.

3) Analog-to-digital converter input (ADC)

The ADC block enables us to retrieve the sampled value of a given ADC channel and to convert it to its value in physical unit. The input signal is typically a continuous signal coming from the plant model and representing a measurement value.

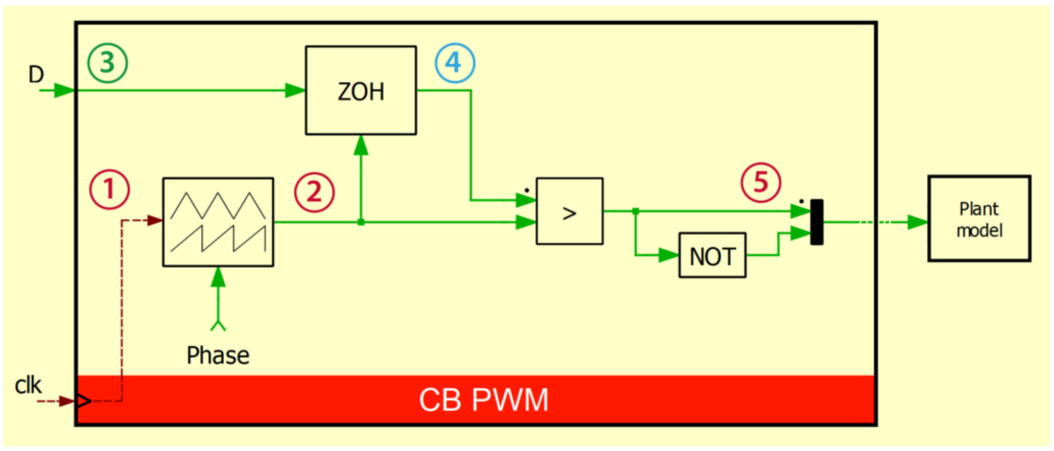


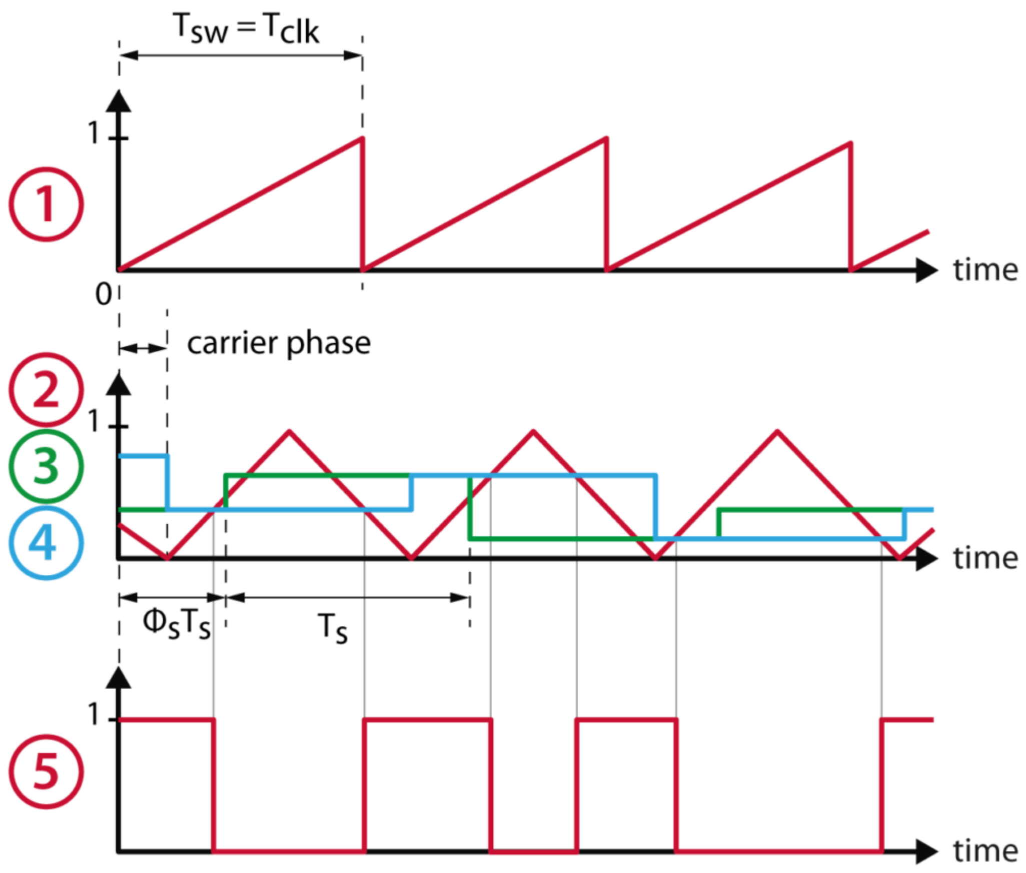
The simulation model of the ADC block simply sample-and-holds the input signal on the rising edges of the ADC trigger input. The sampled signal is then scaled according to the sensor sensitivity parameter to be converted to its value in physical unit.

4) Pulse-width modulators (xx-PWM)

Various types of pulse-width modulators exist within the imperix control library. Among them, the carrier-based PWM modulator (CB PWM block) is certainly the most commonly used.

The CB-PWM block configures the corresponding FPGA peripheral and generates the PWM signals according to the duty-cycle and carrier phase parameters.





In the simulation model, the clock signal (1) connected to the clock input is used as a time reference to generate the carrier signal (2). In parallel, the duty-cycle value (3) is sampled once or twice per switching period (depending on the update-rate parameter) and compared with the carrier to produce the output PWM signal. A dead-time is then applied to the complementary PWM signals (5). (A more complex model is used to generate a carrier with a variable phase, which is beyond the scope of this document.)

Further readings

- Programming and operating imperix controllers (PN138)
- Cockpit – User guide (PN300)