

PN145: Multi-rate control with Simulink

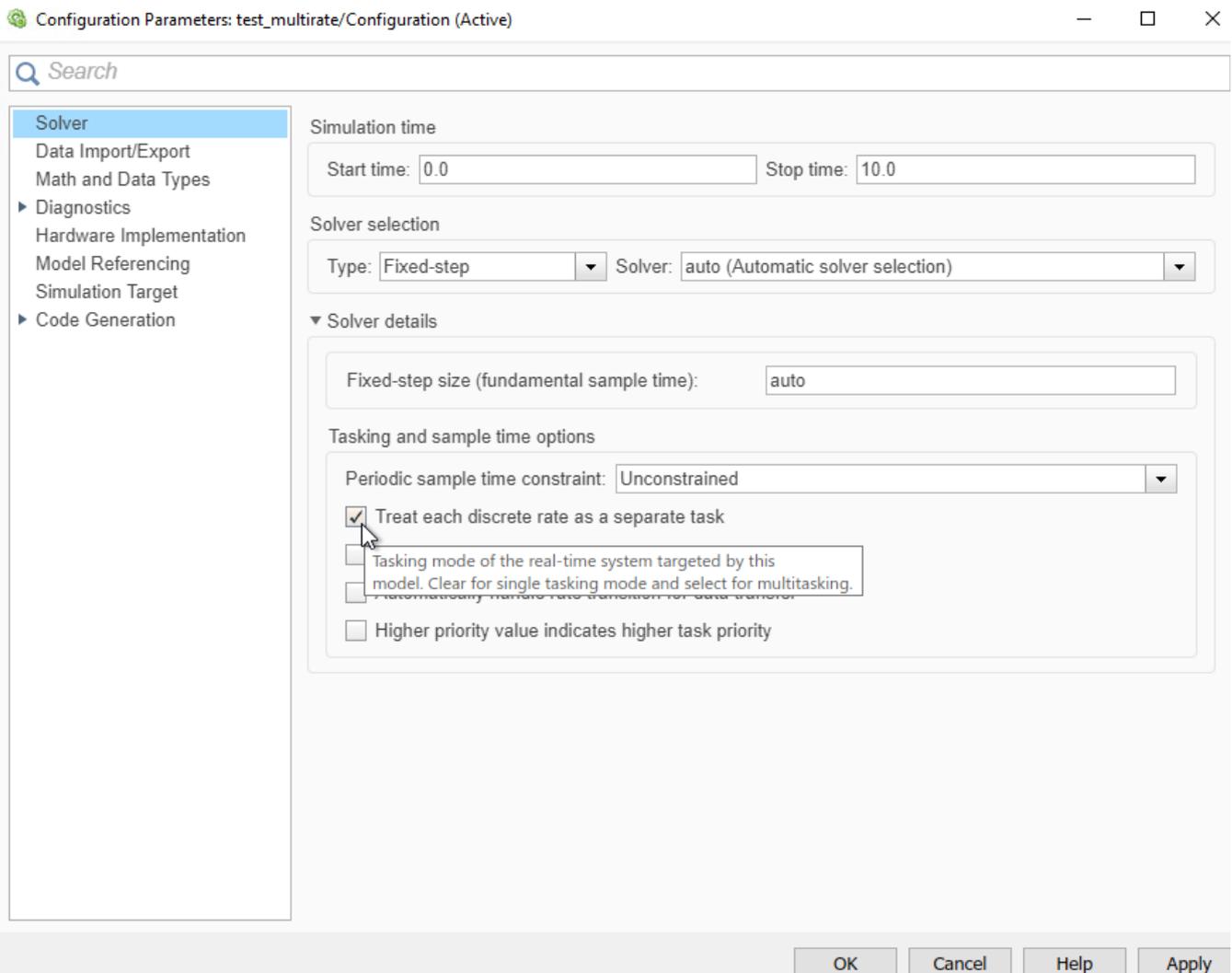
This document provides instructions on how to implement a MultiTask Simulink model with imperix ACG SDK. It allows running part of the control at a slower rate, without impacting the real-time control running at the base rate. This is typically useful for implementing a finite state machine or running a tracking algorithm.

 An example of use is presented in [TN117: Maximum Power Point Tracking \(MPPT\)](#).

 This feature has been integrated in **ACG SDK version 3.5.0.0**, which is available from <https://imperix.com/downloads>.

Enable MultiTasking

- Make sure the imperix CONFIG block mode is set as *code generation*
- Go in the **Model Configuration Parameters** -> **Solver** -> **Solver details**
- Check **Treat each discrete rate as a separate task** (for older MATLAB version: for the option **Tasking mode for periodic sample times** select *MultiTasking*)



Configuration Parameters: test_multirate/Configuration (Active)

Search

Solver

- Data Import/Export
- Math and Data Types
- ▶ Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- ▶ Code Generation

Simulation time

Start time: 0.0 Stop time: 10.0

Solver selection

Type: Fixed-step Solver: auto (Automatic solver selection)

▼ Solver details

Fixed-step size (fundamental sample time): auto

Tasking and sample time options

Periodic sample time constraint: Unconstrained

- Treat each discrete rate as a separate task
- Tasking mode of the real-time system targeted by this model. Clear for single tasking mode and select for multitasking.
- Higher priority value indicates higher task priority

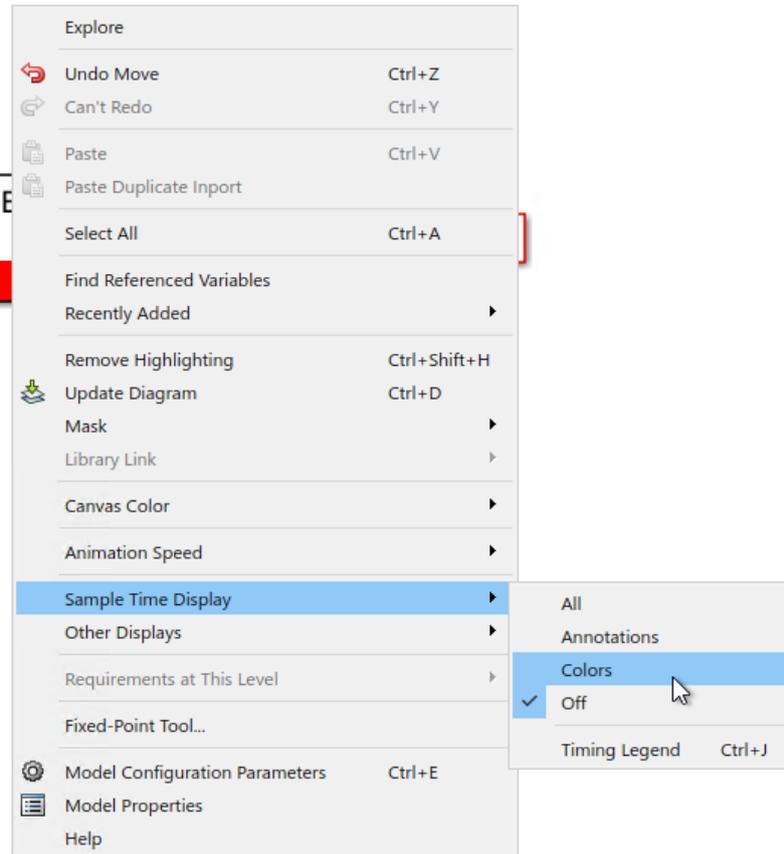
OK Cancel Help Apply

Using MultiTasking

The base rate frequency is set using the CONFIG block, it defines the interrupt frequency. The resulting sample time is available through the global variable CTRLPERIOD.

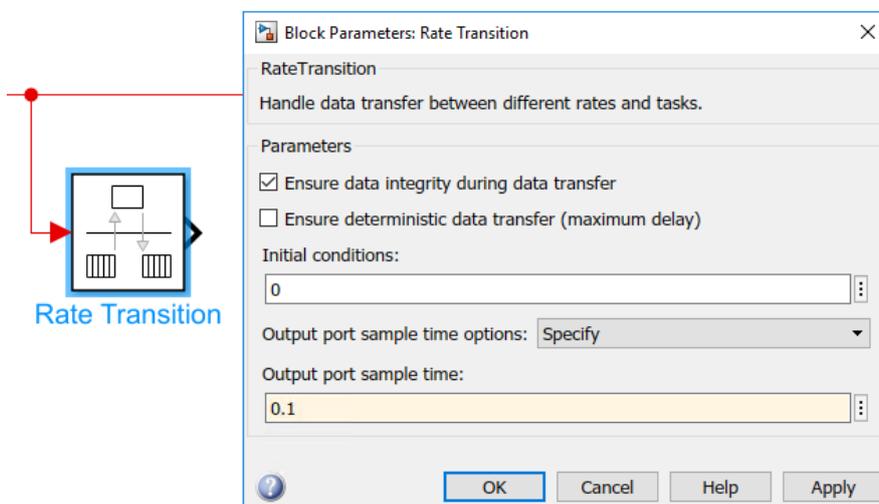
i The CPU-FPGA communication is tightly coupled to the interrupt. As such, FPGA drivers blocks (CLK, ADC, PWM, GPI, GPO,...) can only be called at the base rate frequency.

To enable colors to display sample time do **Right-click on the model Sample Time Display Colors**



To interface logic running at a different rate, add **Rate Transition** blocks. In the block parameters:

- set the **output port sample time** (for instance 0.1 for a 10 Hz frequency)
- check **Ensure data integrity during data transfer** so the generated code will use an intermediate variable (and thus avoid data corruption)



An example of use is presented in [TN117: Maximum Power Point Tracking \(MPPT\)](#). It shows how one part of the control (in red in the figure below) can be executed at a slower rate than the main control code.

