Rate Transition Information

The Rate Transition block transfers data from the output of a block operating at one rate to the input of a block operating at a different rate. Use the block parameters to trade data integrity and deterministic transfer for faster response or lower memory requirements.

The behavior of rate transition depends on,

1. The sample of the port to which this block is connected to.
2. Priorities of the tasks for the source and destination sample time.
3. Whether the model specifies a fixed or variable-step solver.

Block Labels

When you update your diagram, a label appears on the Rate Transition block to indicate simulation behavior.

A screenshot of a social media post

Description automatically generated

1. Sample of the Port

Sample time behavior is indicated differently when the sample time are periodic and non-periodic.

The following table summarizes how each label appears when the sample times of the input and output ports (inTs and outTs) are **Periodic**.A screenshot of a computer

Description automatically generated

Where Block label indicates the simulation behavior,

inTs, outTs: Sample times of input and output ports, &

inTsOffset, outTsOffset: Sample time offsets of input and output ports,

N: Integer value > 1

The following table summarizes how each label appears when the sample times of the input and output ports (inTs and outTs) are **Not Periodic**.

A screenshot of a social media post

Description automatically generated

1. Priorities of the tasks

The behavior of rate transition depends on which task has the higher priority and **Faster sample times must have higher priorities** which is adjusted from the **Sample Time Properties**.

**Format**

[period, offset, priority]

**Period**: The time interval (sample rate) at which updates occur during the simulation.

**Offset**: A time interval indicating an update delay. The block is updated later in the sample interval than other blocks operating at the same sample rate.

**Priority**: Execution priority of the real-time task associated with the sample rate.

#### Example

[[0.1, 0, 10]; [0.2, 0, 11]; [0.3, 0, 12]]

* Declares that the model should specify three sample times.
* Sets the fundamental sample time period to 0.1 second.
* Assigns priorities of 10, 11, and 12 to the sample times.
* Assumes higher priority values indicate lower priorities — the **Higher priority value indicates higher task priority** option is not selected.

### Tips

* If the model's fundamental rate differs from the fastest rate specified by the model, specify the fundamental rate as the first entry in the matrix followed by the specified rates, in order from fastest to slowest.
* If the model operates at one rate, enter the rate as a three-element vector in this field — for example, [0.1, 0, 10].
* When you update a model, Simulink software displays an error message if what you specify does not match the sample times defined by the model.
* If **Periodic sample time constraint** is set to Unconstrained, Simulink software assigns priority 40 to the model base sample rate. If **Higher priority value indicates higher task priority** is selected, Simulink software assigns priorities 39, 38, 37, and so on, to subrates of the base rate. Otherwise, it assigns priorities 41, 42, 43, and so on, to the subrates.
* Continuous rate is assigned a higher priority than is the discrete base rate regardless of whether **Periodic sample time constraint** is Specified or Unconstrained.

1. Fixed and Variable-step solver

A dynamic system is simulated by computing its states at successive time steps over a specified time span, using information provided by the model. The process of computing the successive states of a system from its model is known as *solving the model*. No single method of solving a model suffices for all systems. Accordingly, Simulink provides a set of programs, known as **solvers**, each of which embodies a particular approach to solving a model. The Configuration Parameters dialog box allows you to choose the solver best suited to your model.

You can choose the solvers provided by Simulink based on the way they calculate step size: fixed-step and variable-step

**Fixed-step solvers** solve the model at regular time intervals from the beginning to the end of the simulation. The size of the interval is known as the step size. You can specify the step size or let the solver choose the step size. Generally, decreasing the step size increases the accuracy of the results while increasing the time required to simulate the system.

**Variable-step solvers**vary the step size during the simulation. They reduce the step size to increase accuracy when a model's states are changing rapidly and increase the step size to avoid taking unnecessary steps when the model's states are changing slowly. Computing the step size adds to the computational overhead at each step but can reduce the total number of steps, and hence the simulation time required to maintain a specified level of accuracy for models with rapidly changing or piecewise continuous states.

***Fixed-step*** and ***Variable-step solvers*** compute the next simulation time as the sum of the current simulation time and the step size. The **Type** control on the **Solver** configuration pane allows you to select the type of solver. With a fixed-step solver, the step size remains constant throughout the simulation. With a variable-step solver, the step size can vary from step to step, depending on the model dynamics. In particular, a variable-step solver increases or reduces the step size to meet the error tolerances that you specify.

The choice between these types depends on how you plan to deploy your model and the model dynamics. If you plan to generate code from your model and run the code on a real-time computer system, choose a fixed-step solver to simulate the model. You cannot map the variable-step size to the real-time clock.

If you do not plan to deploy your model as generated code, the choice between a variable-step and a fixed-step solver depends on the dynamics of your model. A *variable-step* *solver* might shorten the simulation time of your model significantly. A variable-step solver allows this saving because, for a given level of accuracy, the solver can dynamically adjust the step size as necessary. This approach reduces the number of steps required. The *fixed-step solver* must use a single step size throughout the simulation, based on the accuracy requirements. To satisfy these requirements throughout the simulation, the fixed-step solver typically requires a small step.

A screenshot of a social media post

Description automatically generated

*Source:* <https://www.mathworks.com/help/rtw/ug/handle-rate-transitions.html>