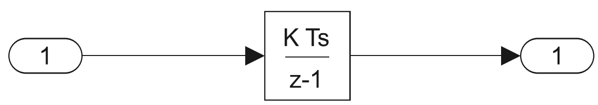
Simulink Automatic Code Generation for AMDC

This article explains how to implement the Simulink automatic code generation (Autogen) by demonstrating an example using a simple integrator.

Simulink Autogen Code

Autogen is the process of converting a user Simulink model for a controller into equivalent C code for an embedded system (such as the AMDC). The Autogen feature in Simulink can be used to conveniently convert complex controller implementations into C-code for implementing it on the AMDC. This article presents a step-by-step process of using Autogen to convert a simle integrator (as shown in the figure below) into C code.



Procedure

Pre-Requisites

User needs to install at least the following dedicated MATLAB/Simulink toolboxes/features:

- Simulink
- Embedded coder
- Simulink coders

File Organization

This article assumes that the uses has completed the Blink tutorial, where you set up your repository. To follow this Autogen tutorial, create a new `simulink` folder in your repository and organize the files as shown below:

```
my-AMDC-workspace/              <= master repo
    AMDC-Firmware/              <= AMDC-Firmware as library

         vate-C-code/           <= Your private user C code

    simulink/                   <= Now create this folder
```

Create a Simulink Model

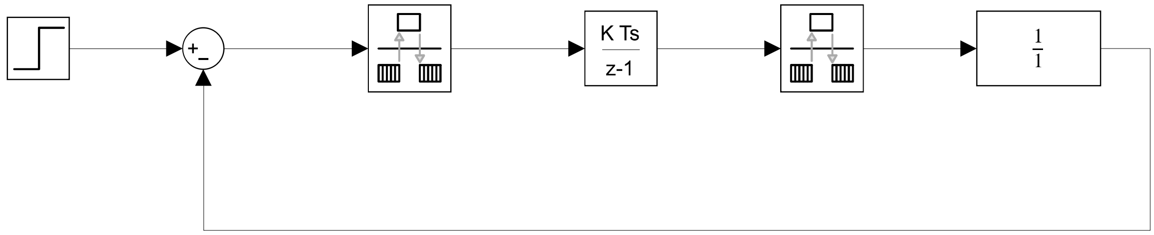
1. In `simulink` folder, create a new MATLAB file (e.g., `setup.m`).
2. In `setup.m`, define `fs = 10e3`, `Ts = 1/fs`, `Tsim = Ts/10`.

User can copy-paste the following MATLAB code:

```
fs = 10e3;       % sampling frequency (Hz)
Ts = 1/fs;       % sampling time (sec)
Tsim = Ts/10;    % simulation time (s)
```

3. Open a blank model of Simulink, and save as `setupModel.slx` in `simulink` folder.
4. Add a `Step` block with the default setting.
5. Add a `Discrete-Time Integrator` block with the default setting.
6. Add a `Rate Transition` block before the integrator. In this block, put `Ts` as a sampling time.
7. Add a `Rate Transition` block after the integrator. In this block, set the sampling time to `-1`.
8. Add a continuous-time `Transfer Fcn` block as a Plant (= 1).
9. Add a `Sum` function and connect each block as shown below.

Skip to content

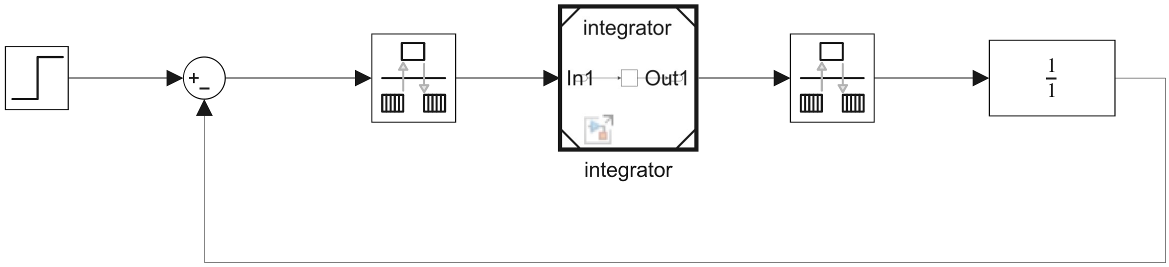


Model Setting

1. In `Modeling` tab, press `Model Settings` in `TOP MODEL` section.
   1. Under the `Solver` tree, in the `Solver Selection`, press `Fixed-step`.
   2. Set `Fixed-step-size` as `Tsim`.
2. Go to `Code Generation`.
   1. Click `Browse` for the `System target file`.
   2. Select `ert.tlc Embedded coder`.
   3. In the `Build process` section, check `Generate code only`.
3. Go to `Optimization` under `Code Generation`.
   1. Choose `None` for the `Leverage target hardware instruction set extensions` in the `Target specific optimizations`.
4. Go to `Templates` under `Code Generation`.
   1. Uncheck `Generate an example main program` in the `Custom templates` section.
5. Click `Apply` and `OK`.

Create a Referenced Model

1. Select the discrete-time integrator, and right-click.
2. Select `Create Subsystem from Selection`.
3. Right-click on the subsystem created. Select `Block parameters (Subsystem)`, check `Treat as atomic unit`, and click `OK`.
4. Right-click on the subsystem and select `Subsystem & Model Reference`. Select `Convert` and click `Referenced Model ...`.
5. In the `Input Parameters` section, define the `New model name` as `integrator`.
6. Click `Apply` and `Convert`.
7. Rename the referenced model block to be `integrator`. The expected Simulink model is shown below:



Referenced Model Setting

1. Double-click the `integrator` referenced model and click `Model Settings` under `Modeling` tab.
2. Click `Model Settings` in the `REFERENCED MODEL` section.
   1. Set `Fixed-step-size` as `Ts`.
3. Save the Simulink file.

The example of Simulink file along with the referenced model is stored here.

Generate C-code

1. Open the `setup.m`.
2. Copy and paste the following code.

```
%% Autogen code for the controller
model='integrator';  % name of the controller to be built
slbuild(model);      % generates the Autogen code
oldFolder = cd('C:integrator_ert_rtw\');
% Copy only .c and .h files in autogen folder
command = 'for /r %i in (*.c, *.h) do copy /y %i ..\autogen';
[status, cmdout] = system(command);
cd(oldFolder);
```

3. Run the `setup.m`, and Autogen code are created in `simulink/autogen` folder.

Integration with AMDC

Now, the user needs to update the user C code to incorporate the Autogen code generated from Simulink. To do this, update `task_controller.c` as follows:

`task_controller.c`:

Skip to content

```c
// ...

int task_controller_clear(void)
{
  // ...

  // Clear state struct for Simulink controller
  memset(((void *) &integrator_DW_DW), 0, sizeof(DW_integrator_T));

  // ...
}

int task_controller_init(void)
{
  // ...

  // Initialize Autogen step
  integrator_initialize();

  // ...
}

void task_controller_callback(void *arg)
{
  // ...

  // Update controller input parameters
  integrator_U.STEP = STEP;

  // Call Autogen code
  integrator_step();

  // ...
}
```

Results

THIS SECTION WILL BE UPDATED!

- After running the AMDC, show the input and output value through logging feature.