

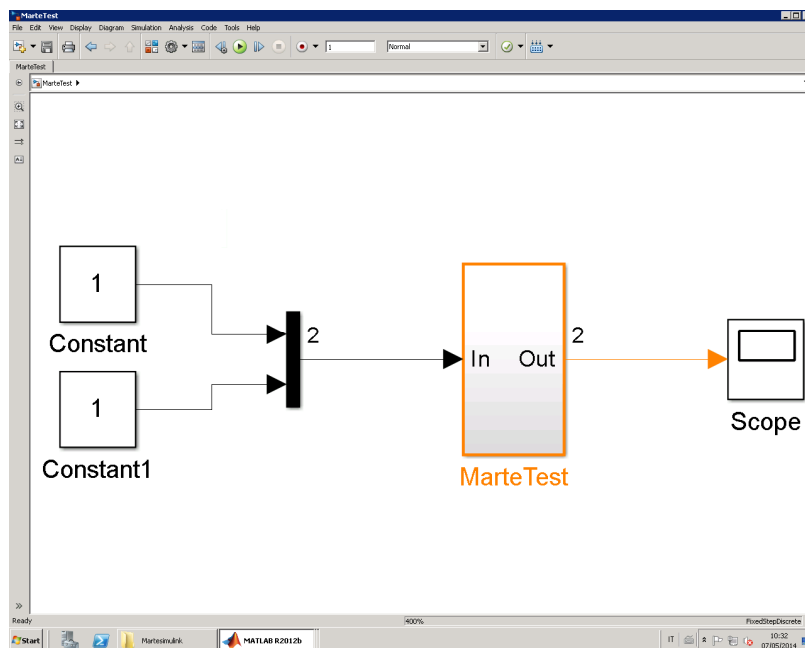
# Automatic GAM generation from Simulink blocks

This document describes the required steps for the automated generation of MARTe GAMs from Simulink blocks. The Matlab reference version is Matlab 2012b. Other Matlab version may have some slight differences which could prevent the automatic code generation.

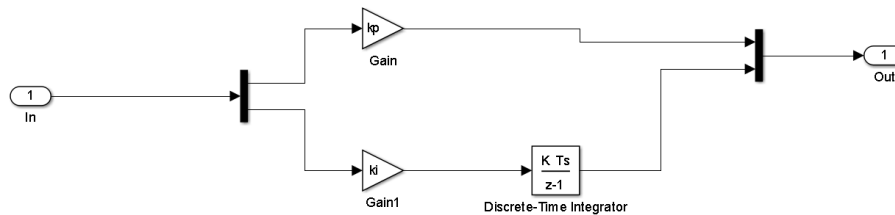
A Simulink block is assumed to have an input array of real value and an output array of real values. Input and outputs will be mapped directly into GAM inputs and outputs. A set of parameters can be defined in the Simulink block. These parameters, with the same name, will be mapped onto GAM parameters which can be defined in the MARTe Configuration file.

In the following, a simple Simulink block with 2 inputs and 2 outputs, computing the multiplication of the first input by a parameter **kp** and the time integration of the second input with integral parameter **ki** will be used to illustrate the required steps in the GAM code generation.

The block, names MarteTest is shown below, where the two inputs are joined into an input array. It is worth noting that The generated GAM will carry out the computation of the Simulink block, not of the rest of the system shown in figure (required when simulating the block behavior within Simulink)



The internal representation of the block is the following. Note that discrete time integration is required.



In the definition of the block two parameters have been defined:  $k_p$  and  $k_i$ . In order to run the block within Matlab it is necessary to assign a value to these parameters and to specify the time step in simulation. The following commands will assign a value to the parameters so that execution within Matlab can be carried out.

```
clc
clear all
kp=1
ki=1
time_step=1e-3
```

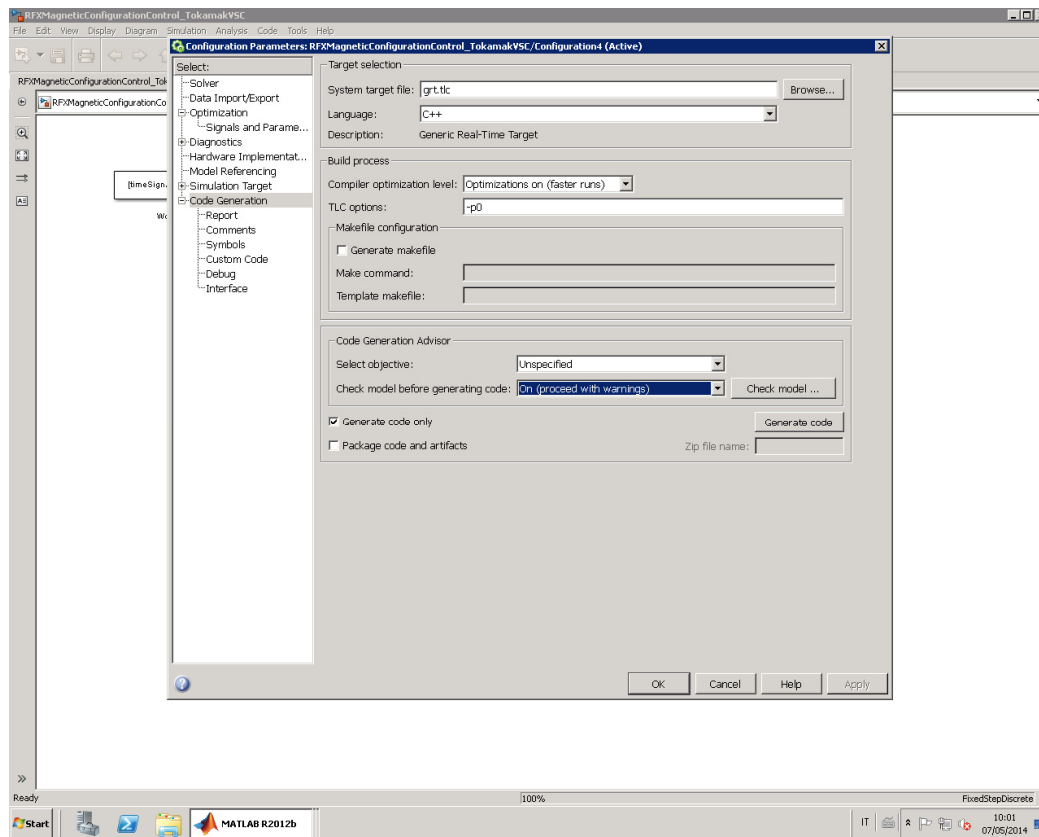
The parameter `time_step` will be reported also in the configuration of the gain and integrator blocks and represents the time step internally used by the generated simulink code to compute integrals and other time-dependent functions.

It is necessary to perform an initial parameter assignment also before generating code from Simulink. The specified values will represent the default values of the parameters is not explicitly changed. Note that the time step value represents an exception to this rule as this value will be hardcoded in the code produced by Simulink. Therefore, whenever the produced GAM has to be executed with a different time step, a new code generation is required.

To generate the code carrying out the block function, select first

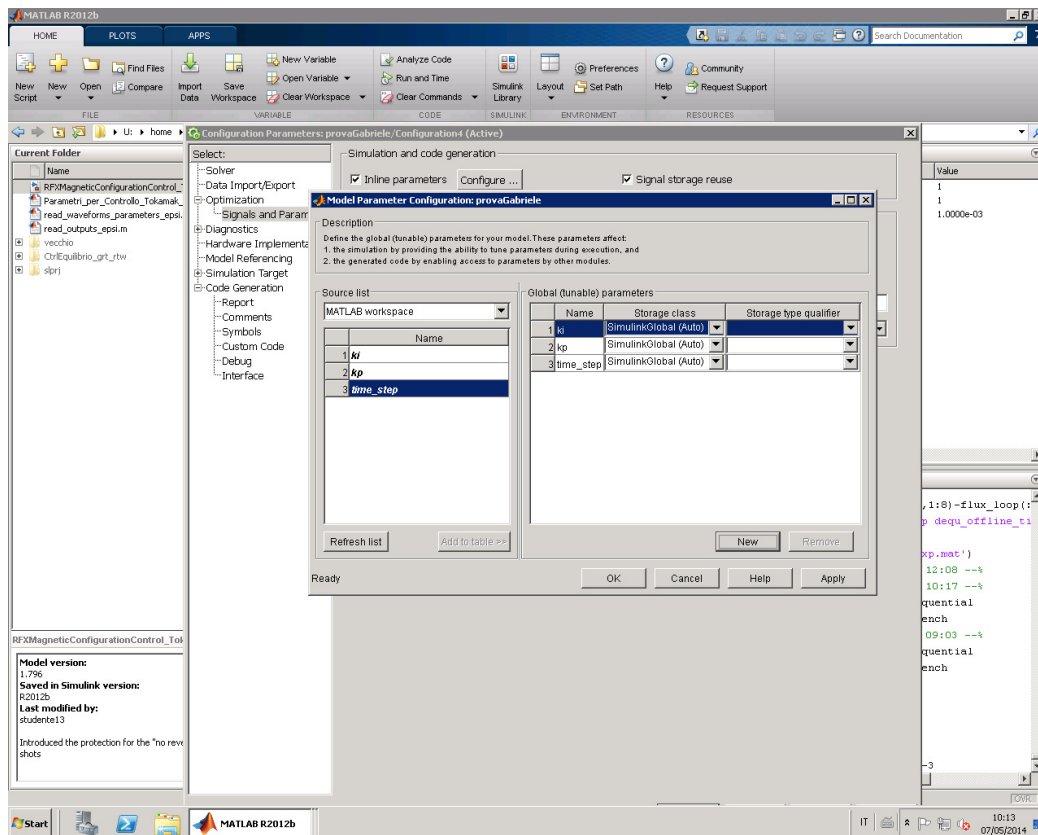
***Simulation->ModelConfigurationParameters***

The window shown below will appear, where several parameters required for the proper code generation must be defined.

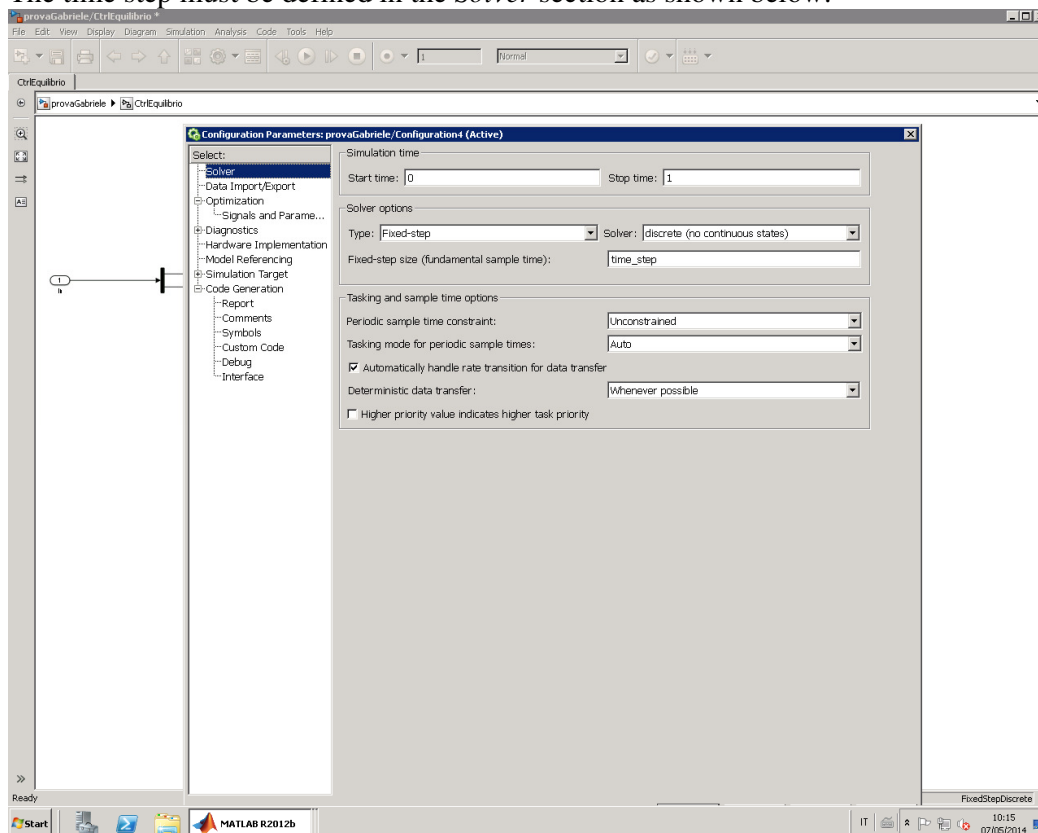


The *Code Generation* section defines among others the System target file and the Language. Select **grt.tlc** and **C++**, respectively.

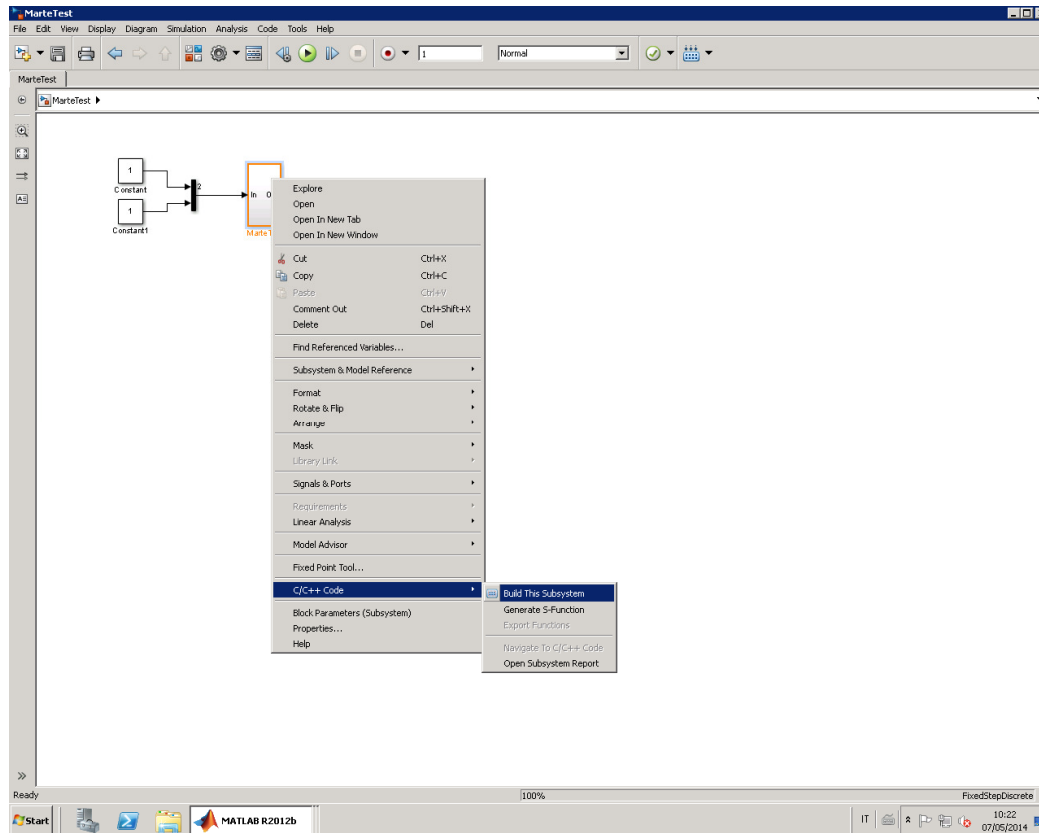
In order to allow the parameters defined in the block to be configurable in the produced code, it is necessary to tick and configure *Inline parameters* in the *Signals and Parameters* section as shown below. Define all the parameters with **SimulinkGlobal(Auto)** Storage class.



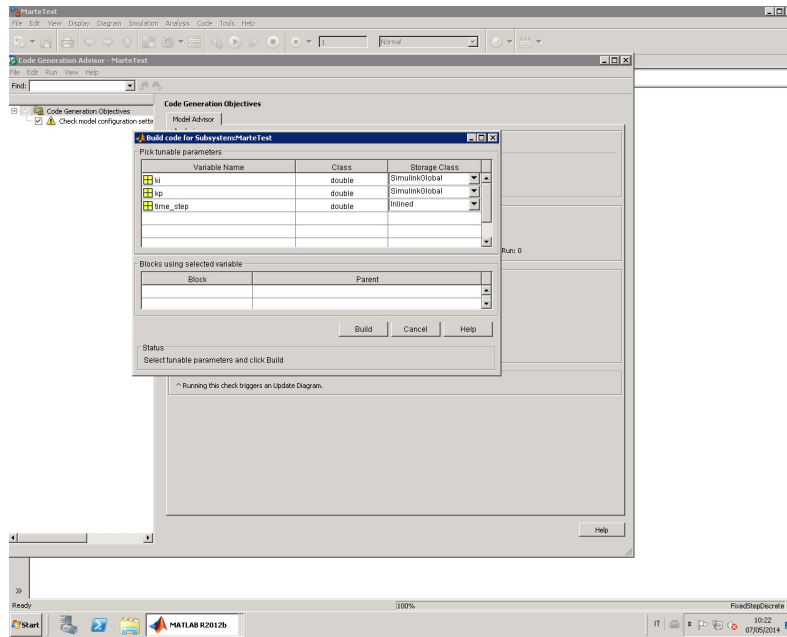
The time step must be defined in the *Solver* section as shown below:



Once configured, Simulink code generation can be started as shown below:



If the code generation succeeds, the window below is displayed, reporting also a summary of the parameters defined for the block. Observe that parameter **time\_step** will be reported as inline despite it has been defined as SimulinkGlobal like the other parameters. The reason is due to the fact that this parameter is used to specify the time step in the computation and therefore is hard-coded in the generated program by Simulink.



At the end of the Simulink code generation phase the directory named **<BlockName>0\_grt\_rtw** has been produced (MarteTest0\_grt\_rtw in this example, the number 0 is added by Simulink). This directory must be copied in the MARTE distribution directory tree, namely in

### **<MARTE Root>/Interfaces/SimulinkInterface**

The code generated by Simulink cannot be directly embedded in MARTE, but a further transformation step is required to embed it into a MARTE GAM. This operation is carried out by the Simulink2GAM.py python program which must be invoked within SimulinkInterface directory as follows:

**python Simulink2GAM.py <BlockName>**

The program will parse the source files produced by Simulink and will populate the (possibly created) directory

### **<MARTE Root>/GAMs/<BlockName>GAM**

with all the files (source, include and Makefile) required to generate the GAM library with the usual command (within the GAM directory)

**make -f Makefile.linux**

(linux is currently only supported)

The new GAM can then be declared in a MARTE configuration file. The following is a sample declaration of the block in a configuration file

.....

```

+SimulinkModule = {
    Class = MarteTestGAM
    kp = 1
    ki = 1
    InputSignals = {
        InSig1 = {
            SignalName = "WaveRefs(0)"
            SignalType = float
        }
        InSig2 = {
            SignalName = "WaveRefs(1)"
            SignalType = float
        }
    }
    OutputSignals = {
        OutSig1 = {
            SignalName = proportional
            SignalType = float
        }
        OutSig2 = {
            SignalName = integral
            SignalType = float
        }
    }
}
}
.....

```