MathModelica® System Designer™

External Functions

1 Introduction

While it is easy to write Modelica functions, it is sometimes convenient to call a subroutine written in C or FORTAN. This example shows how to use an external function written in C.

2 Chirp Function

A chirp signal is a sinusoid with a frequency that changes continuously over a certain band $\Omega : \omega_1 \le \omega \le \omega_2$ over a certain time period $0 \le t \le M$:

$$u(t) = A\cos(\omega_1 t + (\omega_2 - \omega_1)t^2/(2M))$$

The "instantaneous frequency" ω_i in this signal is obtained by differentiating the argument with respect to time *t*:

$$\omega_i = \omega_1 + \frac{t}{M}(\omega_2 - \omega_1)$$

And we see that it increases from ω_1 to ω_2. When applying the signal to a system it gives a good control over the exited frequency band, it is therefore often used for system identification. In this example we will define the chirp function in c and then use it as an external function in Modelica.

3 Modeling

We begin by creating a Modelica function called Chirp that will make an external call to a c function with the same name (for details on how to create models, please check multi domain example).

```
function Chirp
   input Modelica.SIunits.AngularVelocity w_start;
   input Modelica.SIunits.AngularVelocity w_end;
   input Real A;
   input Real M;
   input Real t;
   output Real u "output signal";
   external "C" annotation(Include="#include \"Chirp.c\"");
end Chirp;
```

The function has five input signals, one output, and a call to the external function Chirp.c. The declaration assumes that the function Chirp.c is declared with this five inputs and returns a double. If, for some reason, you wish to switch the order of the variables in calling the function this is possible by changing the declaration to, e.g.:

```
external "C" Chirp(t,A,M,w_start,w_end) annotation(Include="#include
\"Chirp.c\"");
```

However in this case we define a function that uses the same variables and in the same order:

```
double Chirp(double w1, double w2, double A, double M, double time)
{
   double res;
   res=A*cos(w1*time+(w2-w1)*time*time/(2*M));
   return res;
}
```

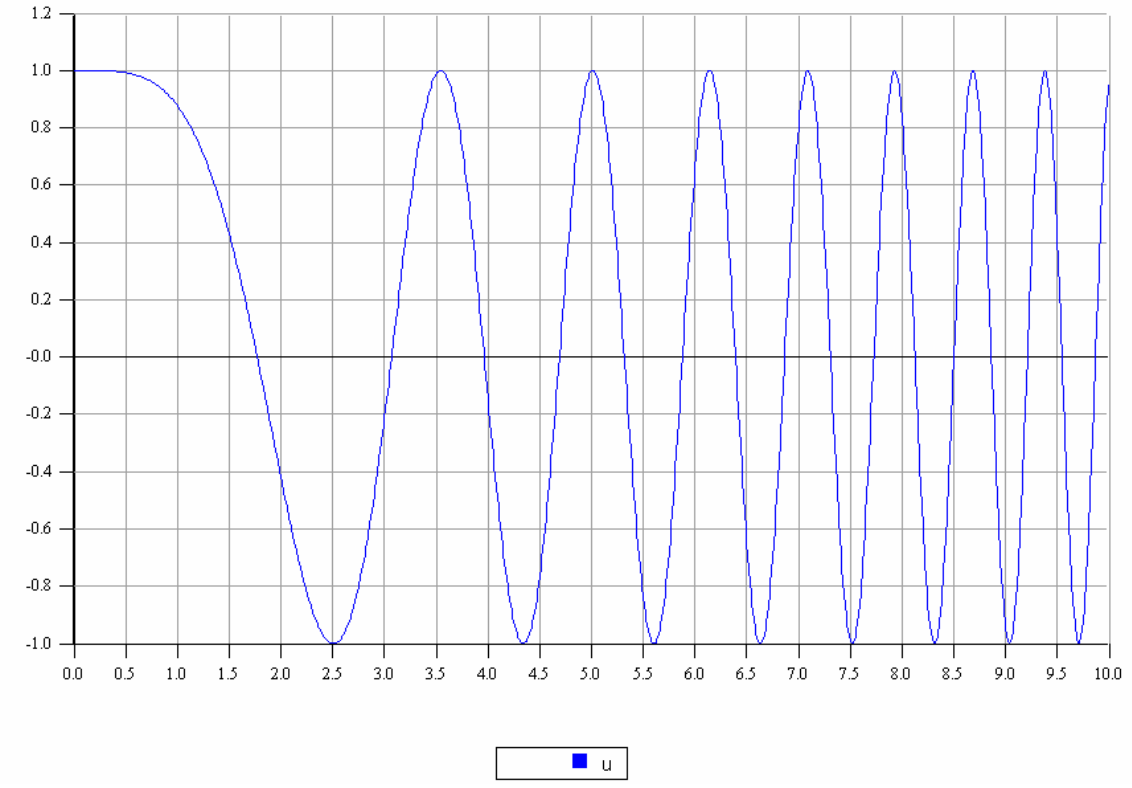
The function can be written in any editor, e.g. notepad, and stored with the name Chirp.c. In this case the c function should be stored in the same library as the Modelica function. However it can be placed in other places to, but the annotation should then be changed accordingly. For instance if you would like to place the function directly in the root of c:

```
external "C" annotation(Include="#include \"c:\\Chirp.c\"");
```

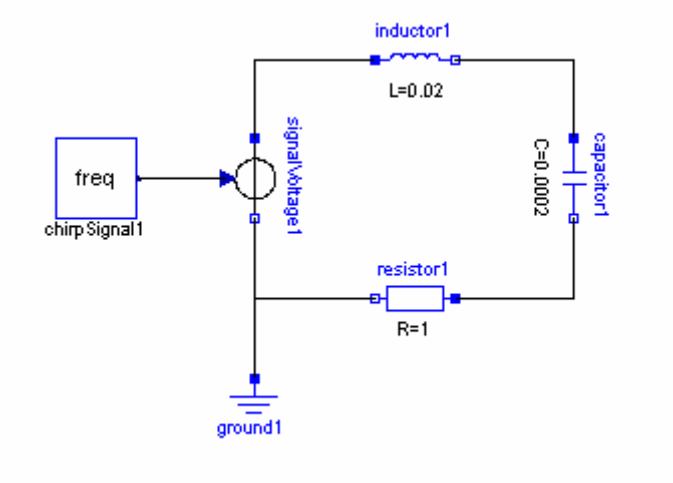
As soon as the c function is saved the Modelica function is ready to use. To do this we define a Modelica class and call the Chirp function in it.

```
class ChirpSignal
   Modelica.Blocks.Interfaces.RealOutput u;
   parameter Modelica.SIunits.AngularVelocity w_start=0;
   parameter Modelica.SIunits.AngularVelocity w_end=10;
   parameter Real A=1;
   parameter Real M=10;
equation
  u=Chirp(w_start, w_end, A, M, time);
end ChirpSignal;
```

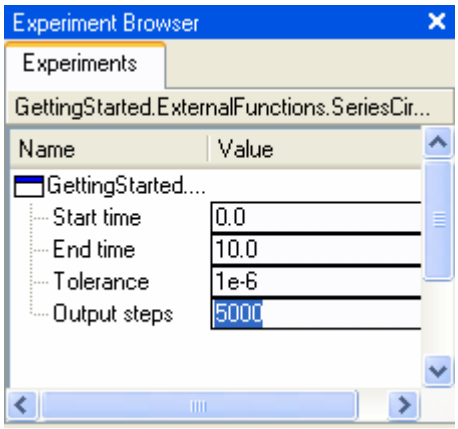
Note that we have set default parameters such that the signal should increase from 0 to 10 rad/s in 10 seconds, as shown by this simulation result:



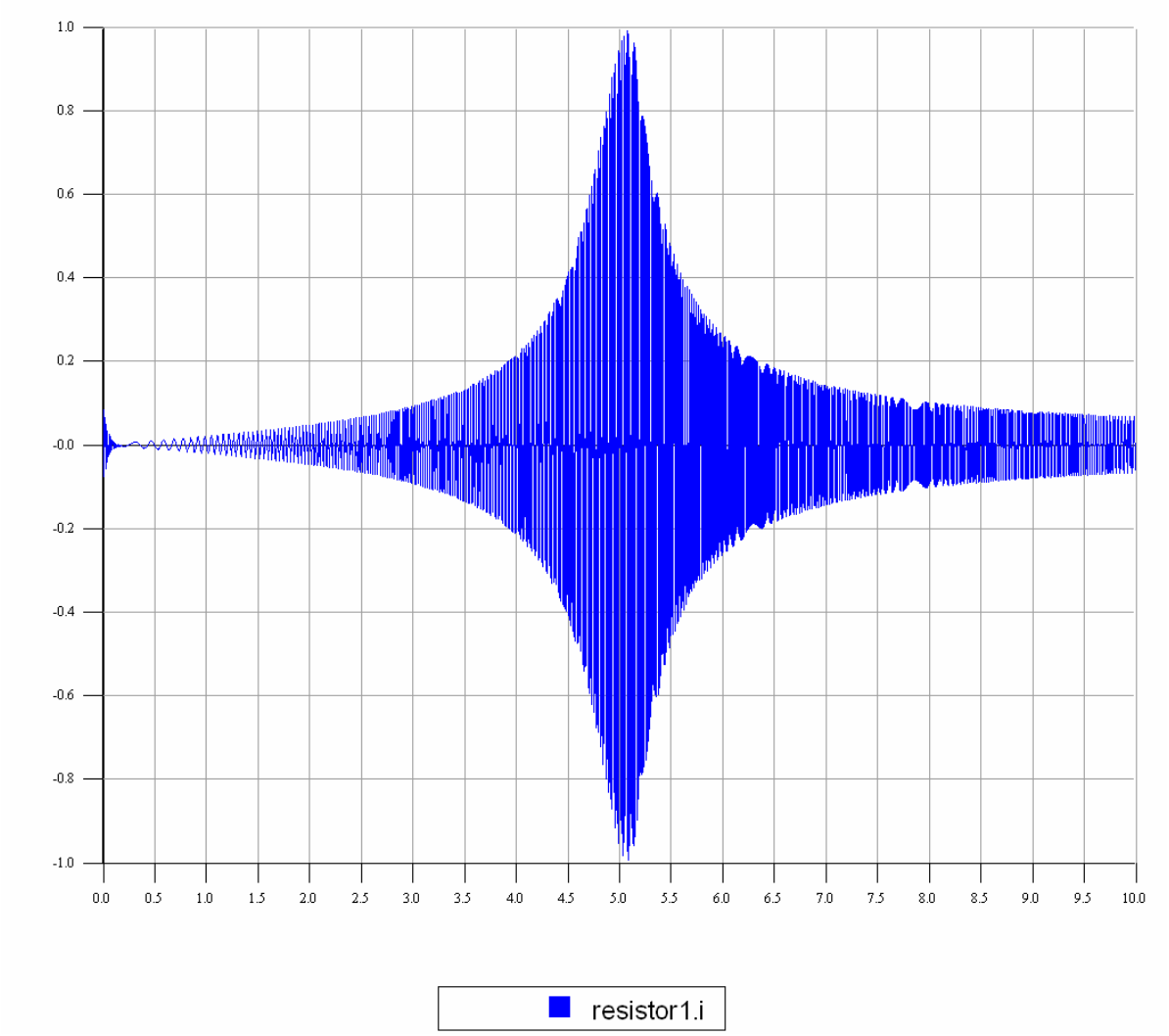
The attentive reader has noted that the variable u was declared as the predefined Modelica connector Modelica.Blocks.Interfaces.RealOutput, which is used in most block models in the Blocks library. As a consequence ChirpSignal can be used in other models as an input source. For instance we can use it to test the resonant frequency of the following electrical circuit (see the Simple Circuit example to learn how to make this model):



Note that the default parameters of the electrical components have been changed according to the figure. We have also changed the parameters of the chirp to sweep from 0 to 1000 rad/s. Before we perform the simulation we need to change the number of output steps, as the high frequency requires this to get a correct plot. We choose 5000 output steps:



After this we simulate and study the current:



As seen we get a top around 5 seconds, which corresponds to:

$$\omega_i = \omega_1 + \frac{t}{M}(\omega_2 - \omega_1) = 0 + \frac{5}{10}(1000 - 0) = 500 rad/s$$

This can also be verified by calculating the resonant frequency for the circuit analytically:

$$\omega_0 = \frac{1}{\sqrt{LC}} = \frac{1}{\sqrt{0.02 * 0.0002}} = 500 rad/s$$

Of course for more complicated systems it might be difficult to calculate the resonant frequency analytically, and in these cases a chirp signal can be very useful. The chirp signal can easily be implemented as one single Modelica block without using any external function. This is left to the interested reader as an exercise.