Chapter 6: **External Functions - Chirp Signal**

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.

6.1 Chirp function

A chirp signal is a sinusoid with a frequency that changes continuously over:

- a certain band:

$$\Omega : \omega_1 \le \omega \le \omega_2$$

- a certain time period:

$$0 \le t \le M$$

We will use the following signal:

$$u(t) = A\cos\left(\omega_1 t + (\omega_2 - \omega_1)\frac{t^2}{2M}\right)$$

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

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

We see that the instantaneous frequency increases from the lower bound of the frequency band to the higher. When applying the signal to a system it gives good control over the excited frequency band, and 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.

6.2 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, see any of the previous examples).

```
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 signal, and a call to the external function Chirp.c. The declaration assumes that the function Chirp.c is declared with these 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, for instance, the following:

```
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 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 text editor, and stored with the name Chirp.c. In this case the C function should be stored in the same library as the Modelica function. It can be placed in other places too, but the annotation should then be changed accordingly. For instance, if desired you can 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 block and call the Chirp function within it.

```
block 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 so that the signal will increase from 0 to 10 rad/s in 10 seconds, as shown by this simulation result:

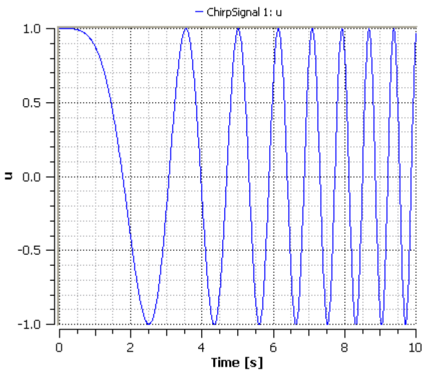

Figure 6-1: Plotting the chirp signal u of the IntroductoryExamples.ExternalFunctions.SeriesCircuit model.

The attentive reader will note that the variable u was declared as the predefined Modelica connector Modelica.Blocks.Interfaces.RealOutput, which is used in most block models in the Modelica Blocks library. As a result, 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.

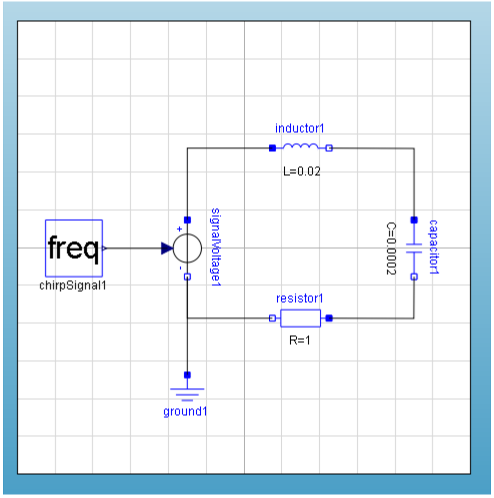

Figure 6-2: The diagram view of the IntroductoryExamples.ExternalFunctions.SeriesCircuit model.

Note that the default parameters of the electrical components have been changed according to the figure above. We have also changed the parameters of the chirp to sweep from 0 to 1000 rad/s.

Next we simulate and study the current.

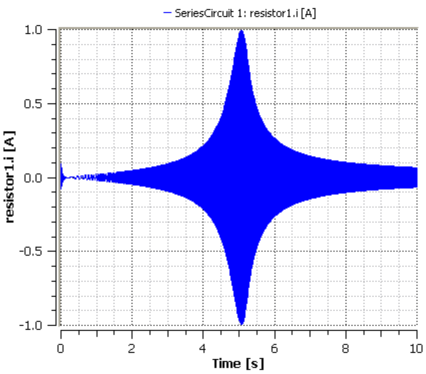

Figure 6-3: Plotting the current i of the IntroductoryExamples.ExternalFunctions.SeriesCircuit model.

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

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

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

$$\omega_0 = \frac{1}{\sqrt{LC}} = \frac{1}{\sqrt{0.02*0.002}} = 500 \text{ rad/s}$$

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.

6.2.1 Exercise

The chirp signal can easily be implemented as one single Modelica block without using a external function. This is left to the interested reader as an exercise.