Component orientated modelling and symbolic manipulation

Written by Mike Dempsey, Claytex

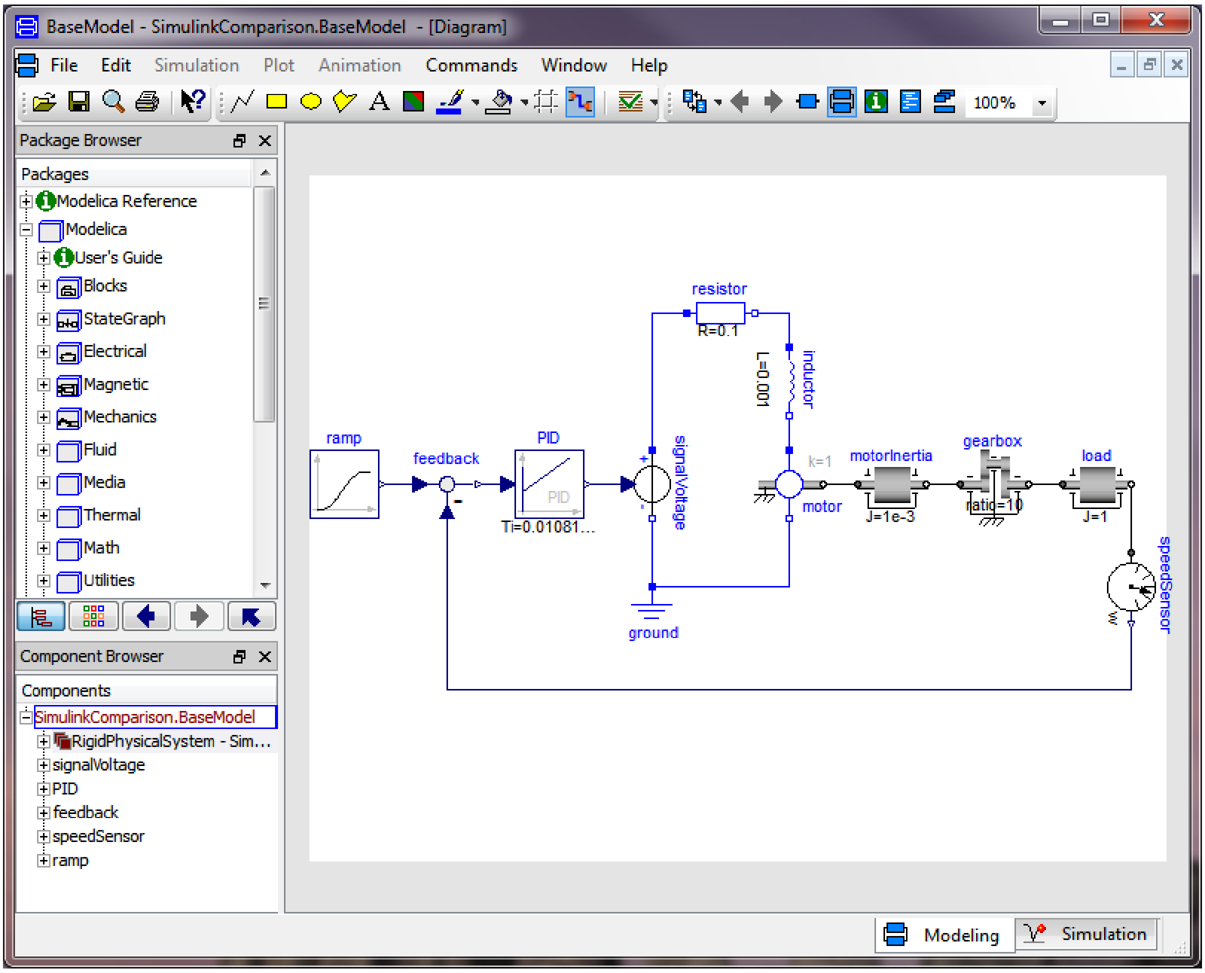
In this 2nd article in the series “Getting more from simulation” we explore how the component orientated, physical modelling approach used in Dymola simplifies model creation and accelerates the model development process.

What a component orientated modelling approach means in practice is that each object in the model diagram represents a real physical part in the system (e.g. a pipe, resistor or gearbox). The Dymola modelling environment frees the engineer to focus on what’s important (i.e. designing and developing the overall system and specifying the individual components) rather than being forced into wasting time developing and debugging an abstract block diagram representing one interpretation of the system. With Dymola there is no requirement for the user to waste time manually rearranging equations, it’s all handled within the next-generation symbolic manipulation engine which comes as standard.

Model creation in Dymola is performed using a drag-and-drop approach. Component models from all engineering domains are available in libraries (shown on the left hand side in the screenshot below) and by simply dragging and dropping the required components in to the model diagram the schematic of the system can be created.

The system consists of an electric motor connected to a 1D rotational inertia and a load inertia via a gearbox. The voltage supplied to the motor is determined using a PID controller that aims to match the rotational speed of the load inertia to the set-point defined in the step function.

Contd……



Components from different engineering domains have different types of connectors that represent the appropriate physical connection between the components. For example, the connection between the motorInertia and gearbox is a 1D rotational mechanical connection and defines the relationship between the angle of rotation of the two components and the torque acting on them. The connections are acausal which means there is no definition of the signal flow in the connection.

To set the component parameters, simply double click on the part and a dialog box showing the parameters appears.

One the model schematic has been created and the parameters specified we can simulate the model. The first step in the simulation process is for Dymola to transform the model diagram in to efficient simulation code which it does through a process known as symbolic manipulation. To appreciate what symbolic manipulation does, we first need to explain how a component model is defined in Dymola.

Dymola uses the Modelica modelling language to define components. Modelica is an open standard modelling language designed for component orientated modelling and is developed and maintained by the Modelica Association. In the above system model we have used 1D rotational inertias (called motorInertia and load in the diagram) and the Modelica definition of this component is shown below. Modelica is a fully object orientated modelling language and you can find out more at http://www.claytex.com/dymola/modelica-modelling-language/

```
model Inertia
  extends Interfaces.Rigid;
  parameter SI.Inertia J=1 "Moment of Inertia";
  SI.AngularVelocity w "Angular velocity";
  SI.AngularAcceleration a "Angular acceleration";
equation
  w = der(phi);
  a = der(w);
  flange_a.tau + flange_b.tau = J * a;
end Inertia;
```

The most important feature of Modelica for this discussion is to notice how the model equations are actually written. The final equation implements Newtons 2nd Law in an instantly recognisable form, i.e. the sum of the torques acting on the object is equal to its inertia times acceleration. Modelica supports equations of the form expression = expression rather than the traditional programming approach that only supports equations of the form variable = expression.

To transform the system model from Modelica code in to efficient simulation code, Dymola uses symbolic manipulation. Symbolic manipulation applies computer algebra techniques to automatically rearrange the model equations in to the required solution. The animation below explains the basic steps of this process:

http://www.claytex.com/wp-content/uploads/2010/05/symbolics_commented.swf

Contd….

In reality, symbolic manipulation is able to deliver more than simply rearranging the model equations in to the appropriate solution. It also simplifies the resulting system of equations through the application of advanced mathematical techniques to handle linear and non-linear systems of equations. Using our example system model, symbolic manipulation manipulates the equations as follows:

Original model

Total number of equations:	86
Sizes of linear systems of equations:	4

Translated model (equations to be calculated at each time step)

State variables:	7
Time-varying variables:	23
Sizes of linear systems of equations:	0

Symbolic manipulation has been able to reduce the size of the model from 86 equations to just 30 equations that need to be calculated at every time step and in whilst achieving this reduction it has also been able to eliminate the linear system of equations. This reduction in the number of equations to be calculated at each time step will deliver improvements in simulation performance.

In the next article in this series we will look at how the component orientated modelling approach compares to the traditional block diagram modelling approach.

Contact

Mike Dempsey

www.claytex.com

Tel: 01926 885900