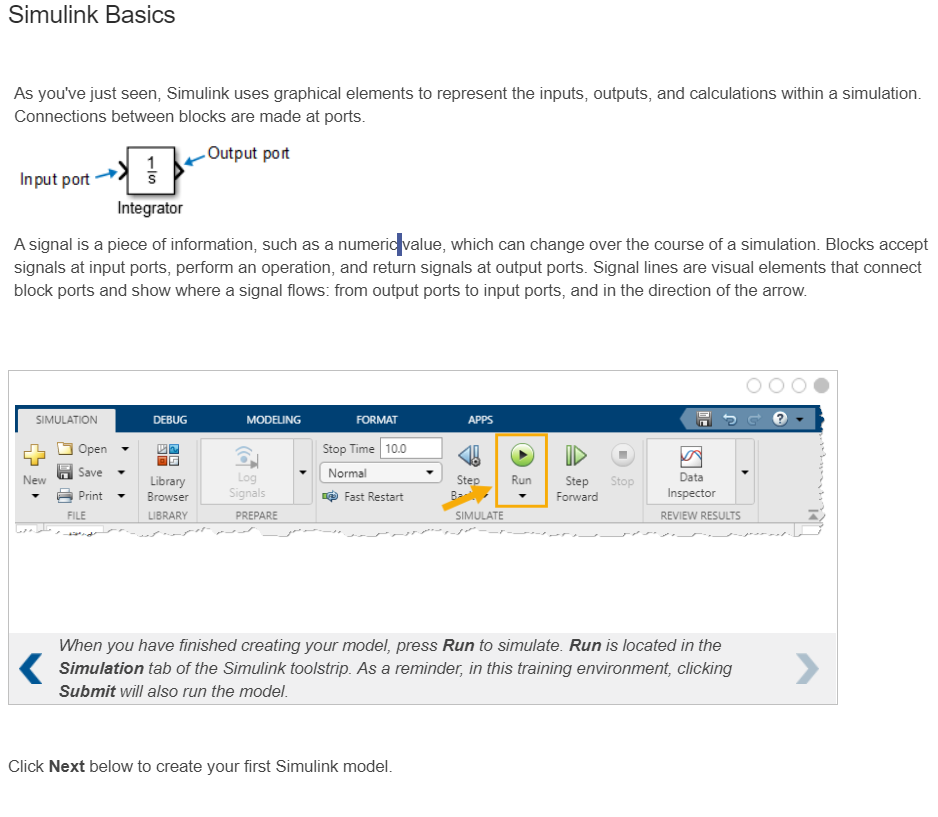
**MATLAB SIMULINK**

**Simulink** is a [MATLAB](https://en.wikipedia.org/wiki/MATLAB)-based graphical programming environment for modelling, simulating and analysing multidomain [dynamical systems](https://en.wikipedia.org/wiki/Dynamical_systems). Its primary interface is a [graphical block diagramming tool](https://en.wikipedia.org/wiki/Visual_modeling) and a customizable set of block [libraries](https://en.wikipedia.org/wiki/Library_(computer_science)). It offers tight integration with the rest of the [MATLAB](https://en.wikipedia.org/wiki/MATLAB) environment and can either drive MATLAB or be scripted from it. Simulink is widely used in [automatic control](https://en.wikipedia.org/wiki/Automatic_control) and [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing) for multidomain simulation and [model-based design](https://en.wikipedia.org/wiki/Model-based_design).

.

* Blocks can be copied from the Library Browser—an organized portfolio of Simulink blocks. For example, the Sources library contains blocks that represent input signals.
* Using a block-
* Each library is an independent collection of blocks Some libraries contain blocks that have the same or similar names. However, the functions of these blocks can be quite different When using the Quick Insert menu, the block's library is listed below the block name

.

* + - Try using the Quick Insert menu to insert a Sine Wave block Be sure it is the same block as you added in Task 1, from the Simulink > Sources Library You can replace the existing black to verify that you are correct

**Signals-**

Simulink signals are mathematical, not physical, entities. The lines in a block diagram represent mathematical, not physical, relationships among blocks. Simulink signals do not travel along the lines that connect blocks in the same way that electrical signals travel along a wire

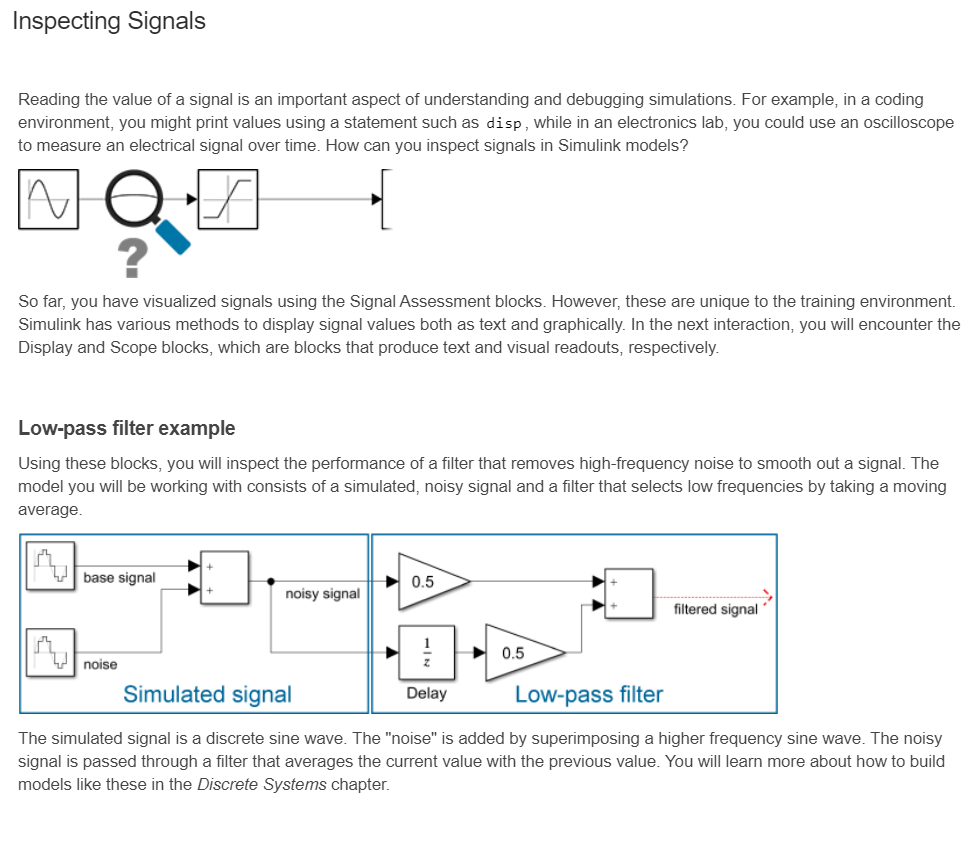
**Inspecting Signals**

Reading the value of a signal is an important aspect of understanding and debugging simulations. For example, in a coding environment, you might print values using a statement such as disp, while in an electronics lab, you could use an oscilloscope to measure an electrical signal over time. How can you inspect signals in Simulink models?

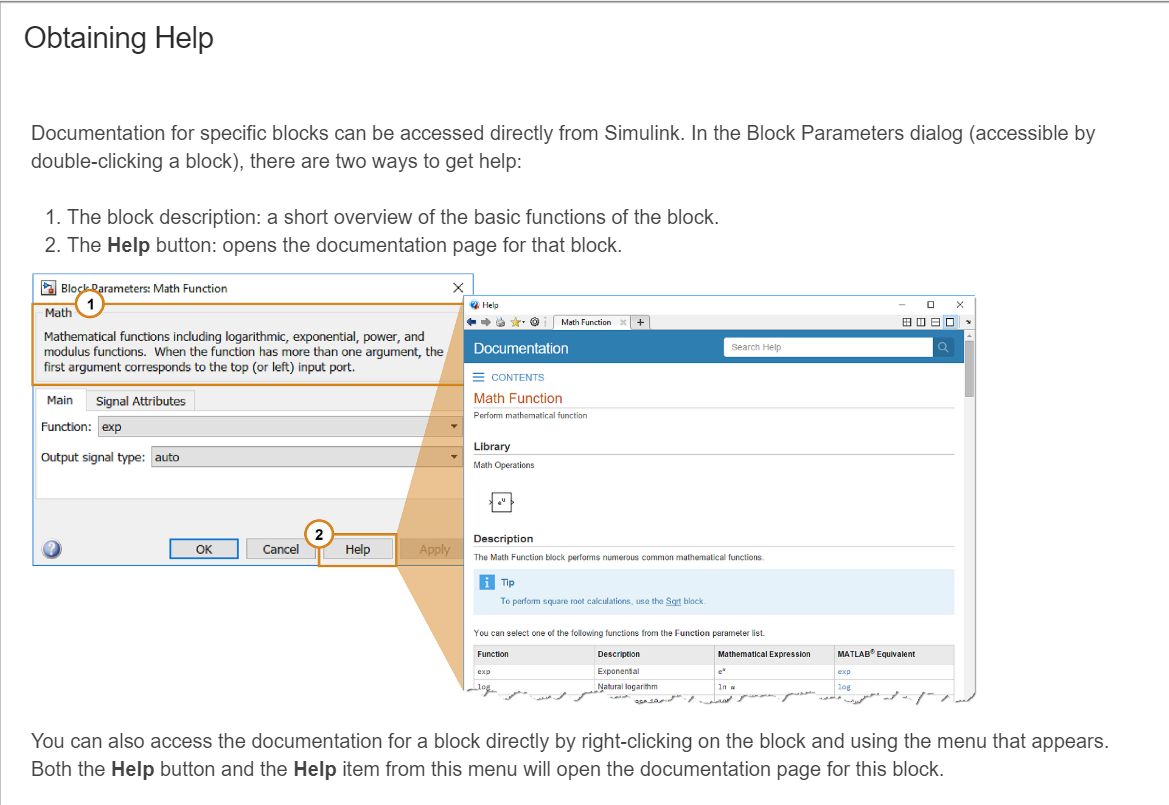
So far, you have visualized signals using the Signal Assessment blocks. However, these are unique to the training environment Simulink has various methods to display signal values both as xt and graphically. In the next interaction, you will encounter the Display and Scope blocks, which are blocks that produce text and visual readouts, respectively.

**Low-pass filter example**

Using these blocks, you will inspect the performance of a filter that removes high-frequency noise to smooth out a signal. The model you will be working with consists of a simulated, noisy signal and a filter that selects low frequencies by taking a moving average

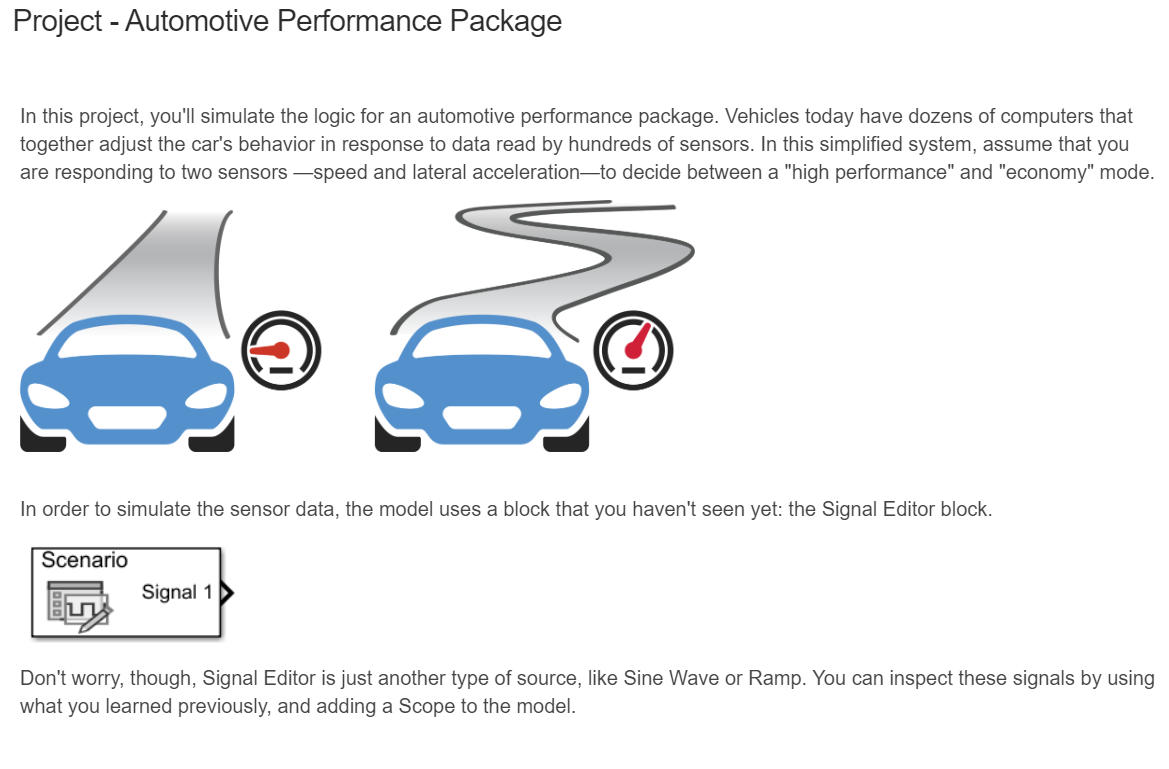
****

**Obtaining help**



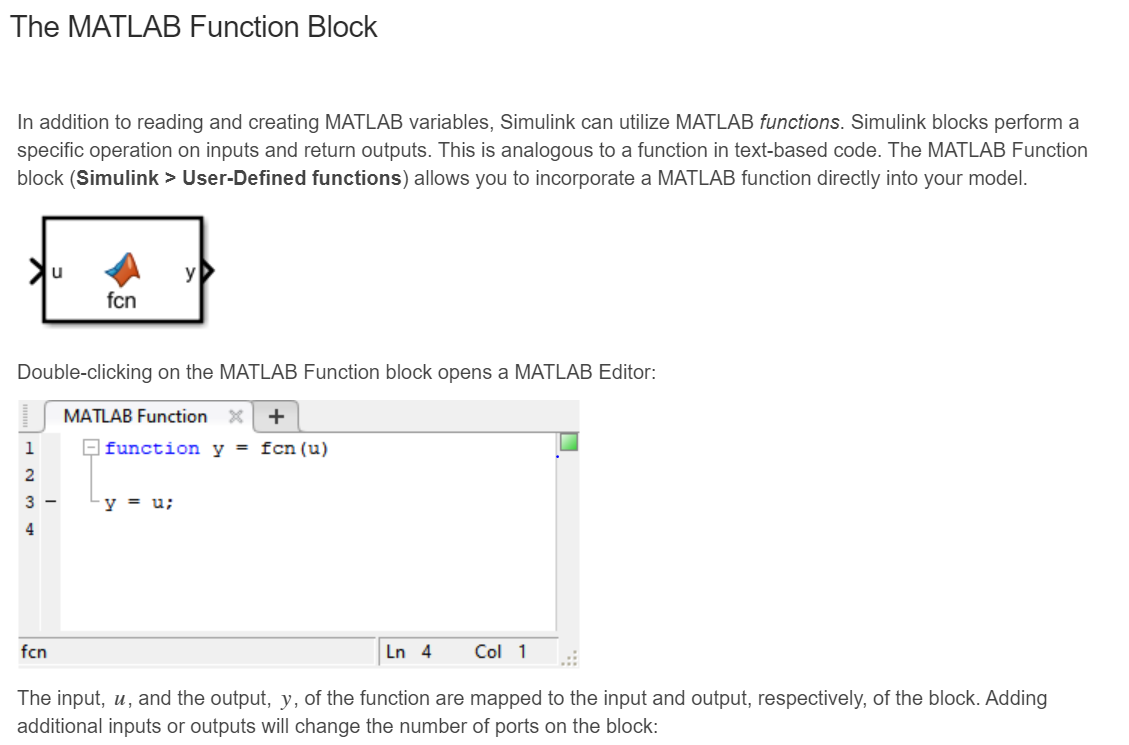
**Project- Automotive performance package**

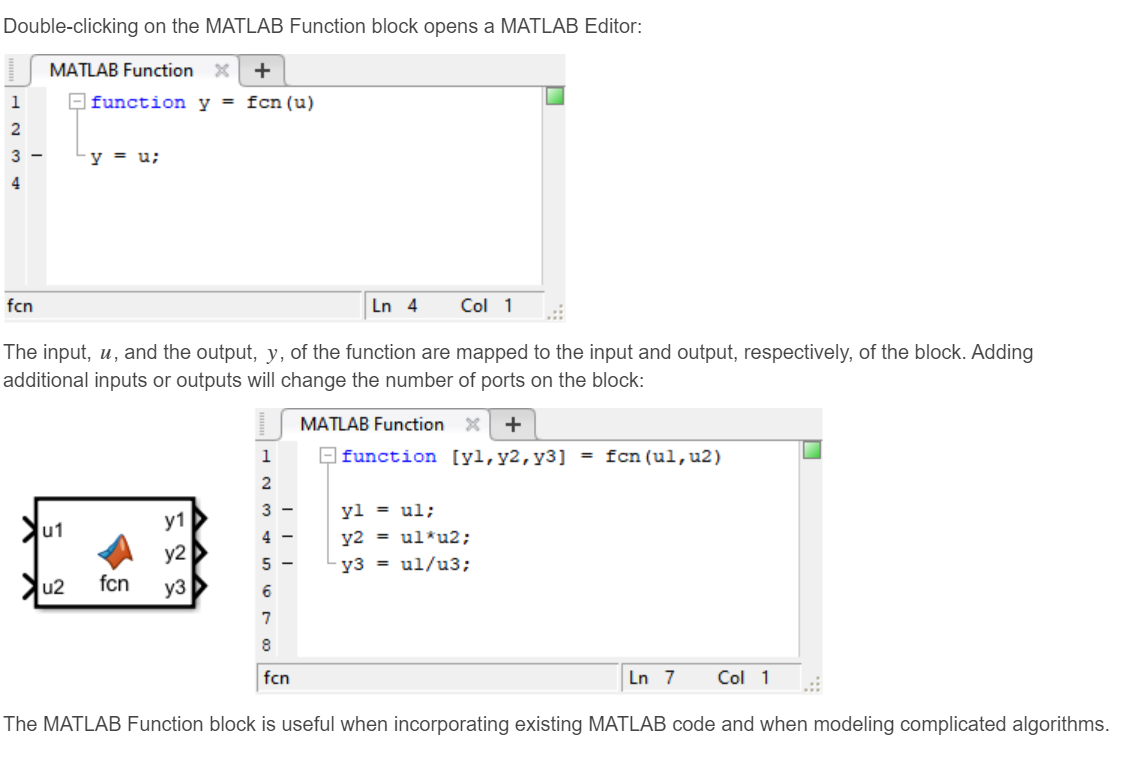
In this project, you simulate the logic for an automotive performance package Vehicles today have dozens of computers that together adjust the car's behaving response to data read by hundreds of sensors. In this simplified system, assume that you are responding to two sensors and lateral acceleration to decide between a "high performance" and "economy" mode



**The MATLAB Function Block**

In addition to reading and creating MATLAB variables. Simulink can utilize MATLAB functions Simulink blocks perform a specific operation on inputs and retum outputs. This is analogous to a function in text-based code. The MATLAB Function block (Simulink> User-Defined functions) allows you to incorporate a MATLAB function directly into your model



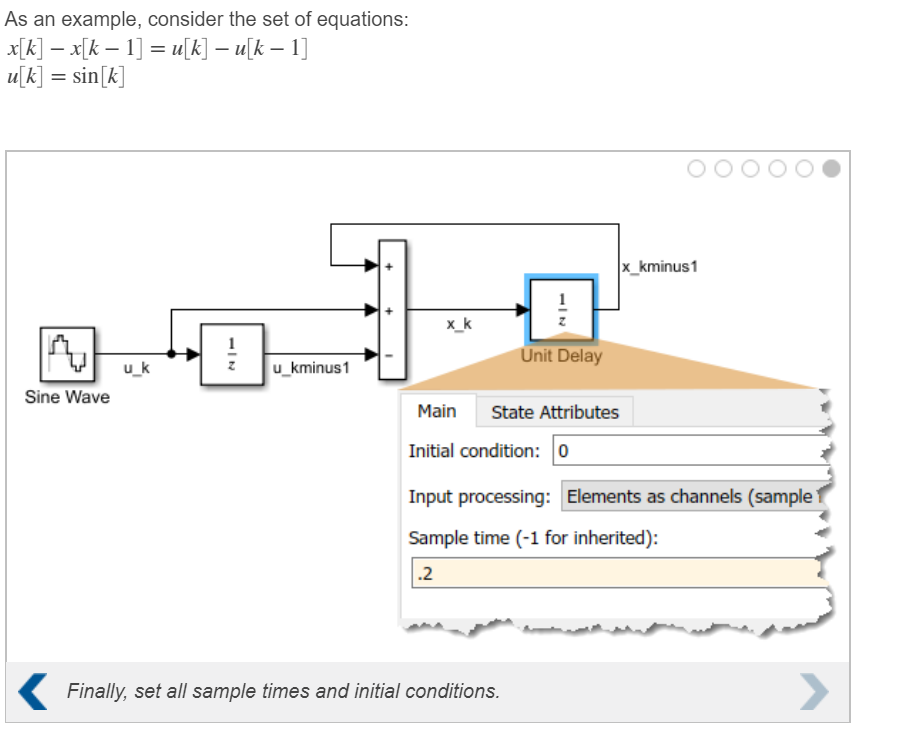


**Modelling Differential equations**

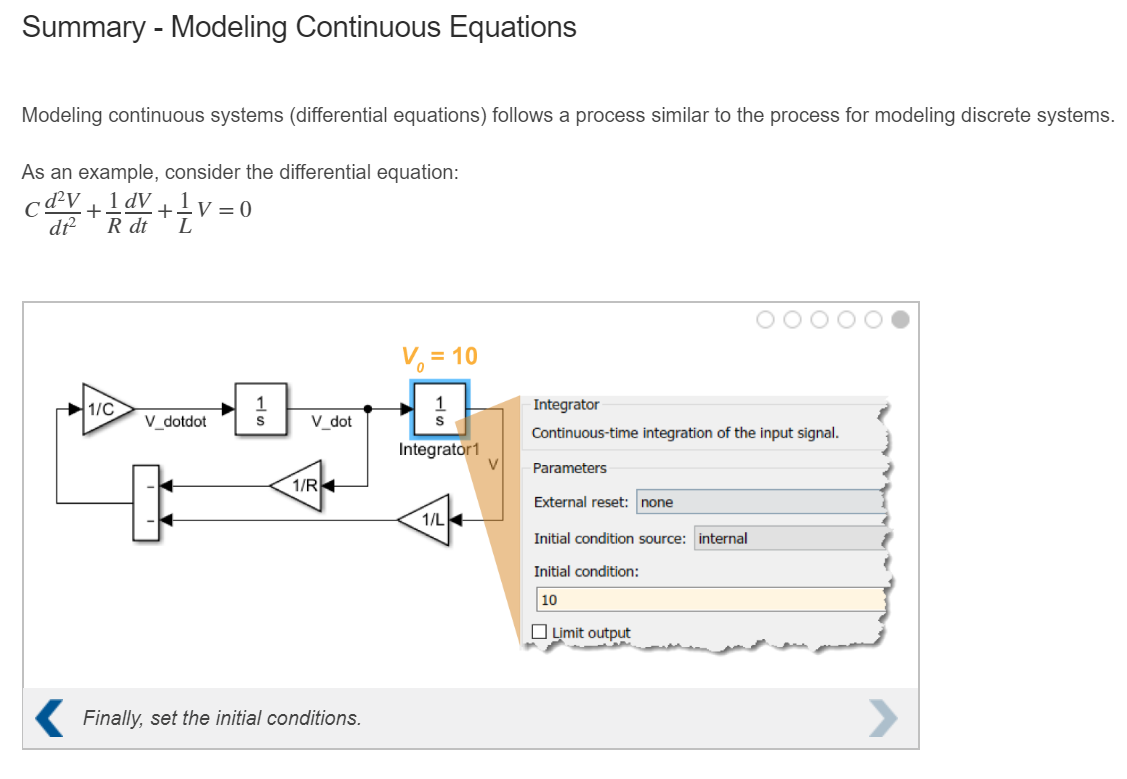
To model difference equations, use the following procedure. This applies to both single equations as well as systems of equations.

As an example, consider the set of equations: x[k] - x[k - 1] = u[k] - u[k - 1] u[k] = sin[k]

x[k] = x[k - 1] + u[k] - u[k - 1] u[k] = sin[k**]**



**Modelling Continuous equations**

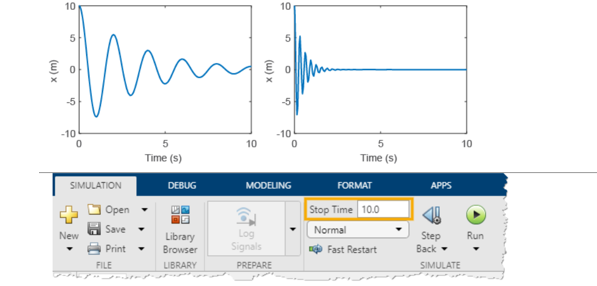


**Simulation Time and Step Size**

When Simulink runs a simulation, it numerically solves the dynamic equations represented in the model. Characteristics of dynamic systems can vary widely; for example, they can be discrete, continuous, fast, and slow. To account for this, there are many types of numerical solvers.

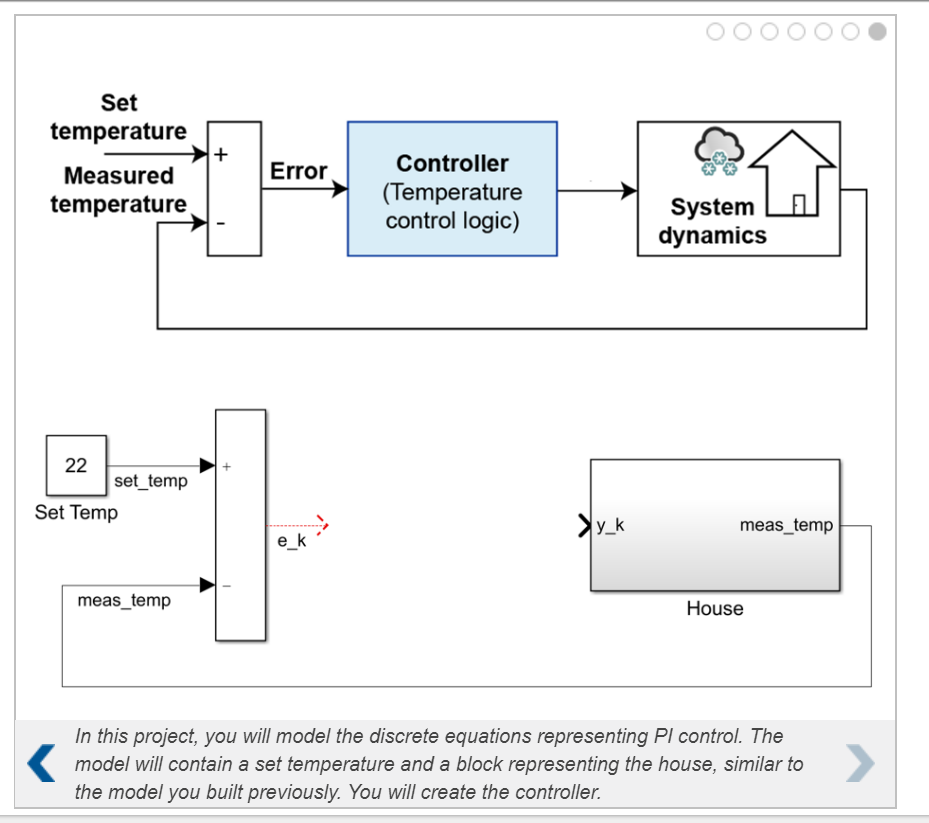
**Simulation duration**

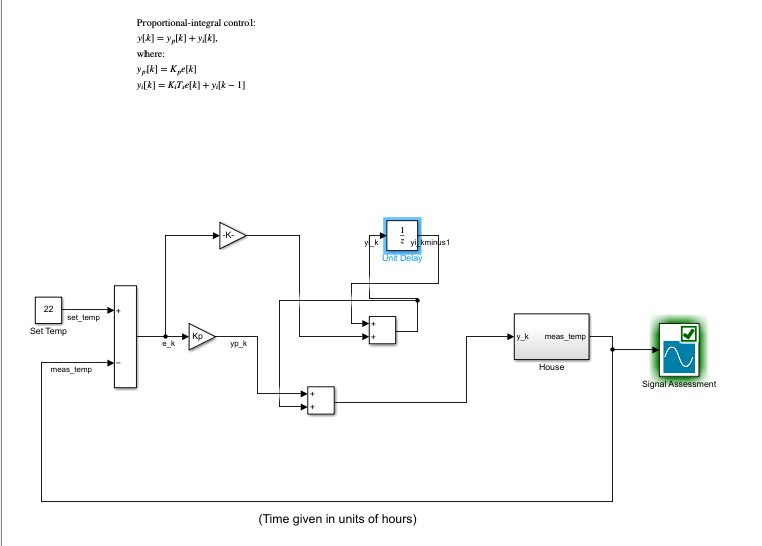
Simulink knows how to assess the properties of your system and automatically select a good solver for you. However, the auto-solver cannot determine how long your simulation should run. For example, the two systems below have different dynamics. To accurately capture the interesting behavior, the slow system (left) should be run for more time, and the fast system (right) for less.



**Project - Thermostat Model**

In Chapter 10, you modelled a first order cooling equation. That equation could represent the temperature dynamics of a house. To keep the house comfortable, you might want to control the temperature, using some strategy based on the current and desired temperatures. This is called a *control system*, and is exactly what a thermostat is! Since most controllers are digital, they are often modelled as discrete systems.





**Project - Peregrine Falcon Dive**

When diving for prey, the Peregrine falcon can reach speeds up to 350 km/h. As the falcon dives, the two forces acting on it are gravity and air drag. The equation of motion for this is:

*m˙v*=(*ρCdAv*2−*mg)/2*

As it approaches the ground, however, it changes the angle of its wings, increasing drag, to rapidly decelerate to a safe landing speed. This can be modelled by a change in the area, *A*, and drag coefficient, *Cd* .In this project, you will model this behaviour in Simulink in two parts. First, you will model the equation of motion for a given set of drag parameters. In the second part, you will model the wings opening to slow the bird down.