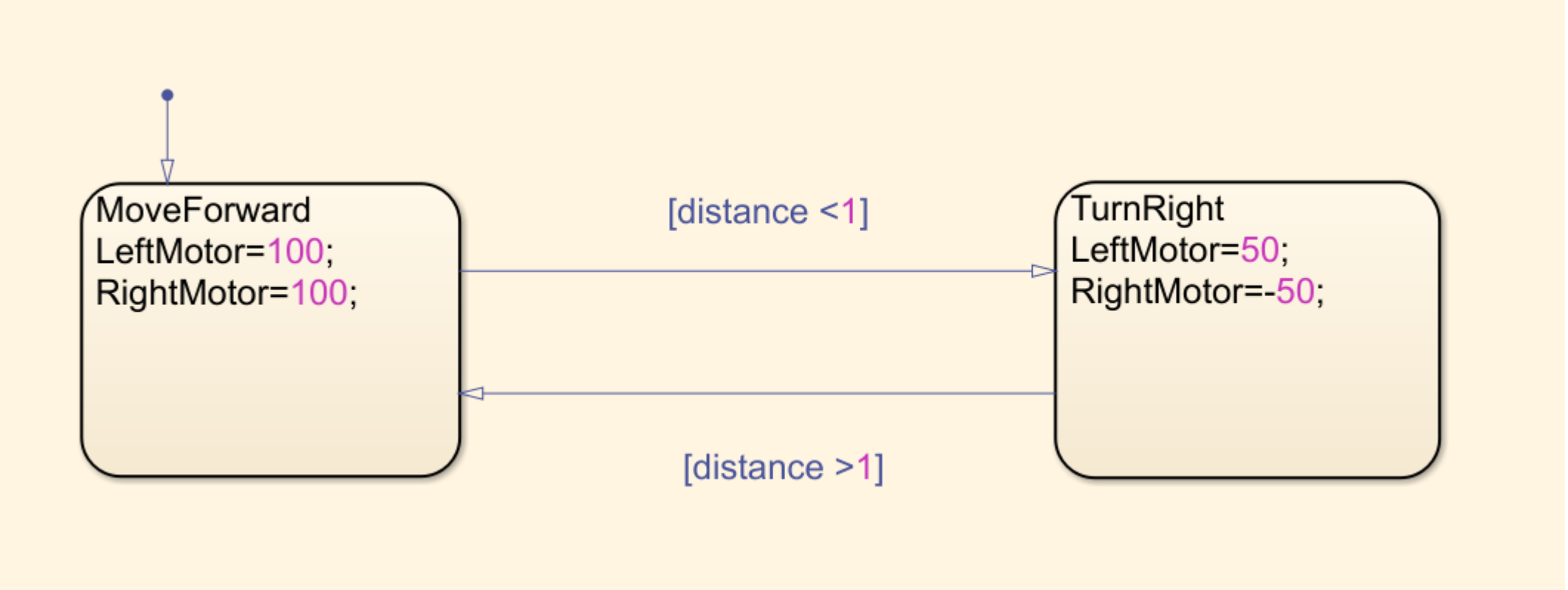
Introduction to Mobile Robotics with MATLAB and Simulink
Unit 7: Intro to Stateflow

By MathWorks Student Competition team

What is Stateflow?

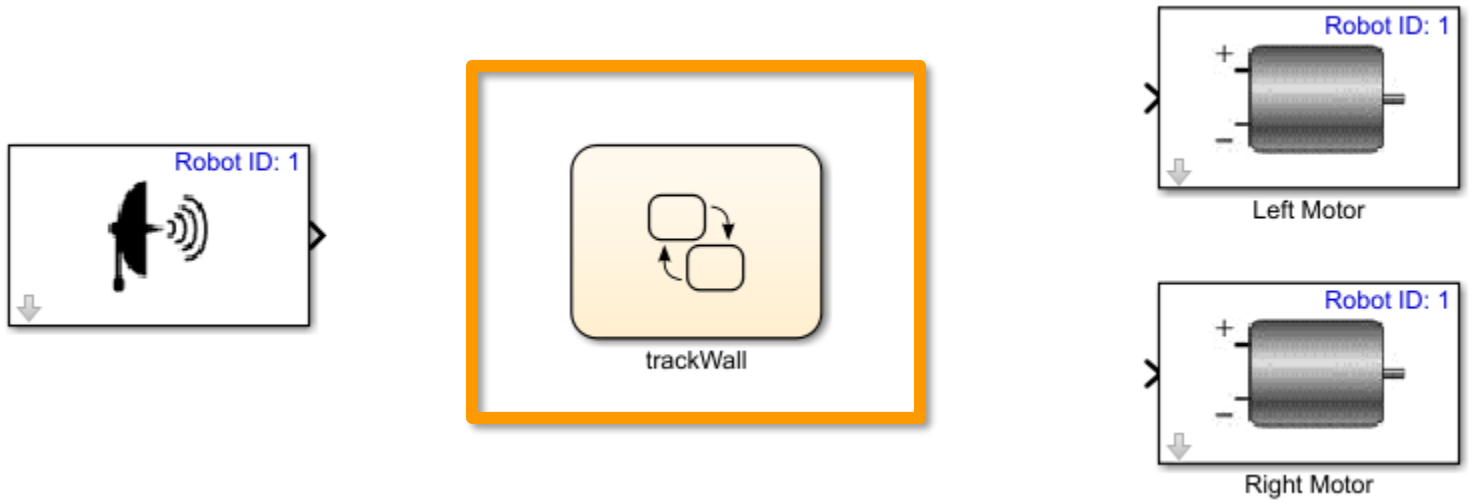
- A Simulink addon to visually and manage control logic using State Machines and Flowcharts
- *https://www.mathworks.com/videos/introduction-to-stateflow-for-student-competition-teams-1507636691946.html*



Adding a Chart to a Simulink Model

- Lets make a Stateflow chart that makes the robot track the wall as in the previous example

1. Open the model **“trackWall_SF_start”**
2. Rename the chart to “trackWall”



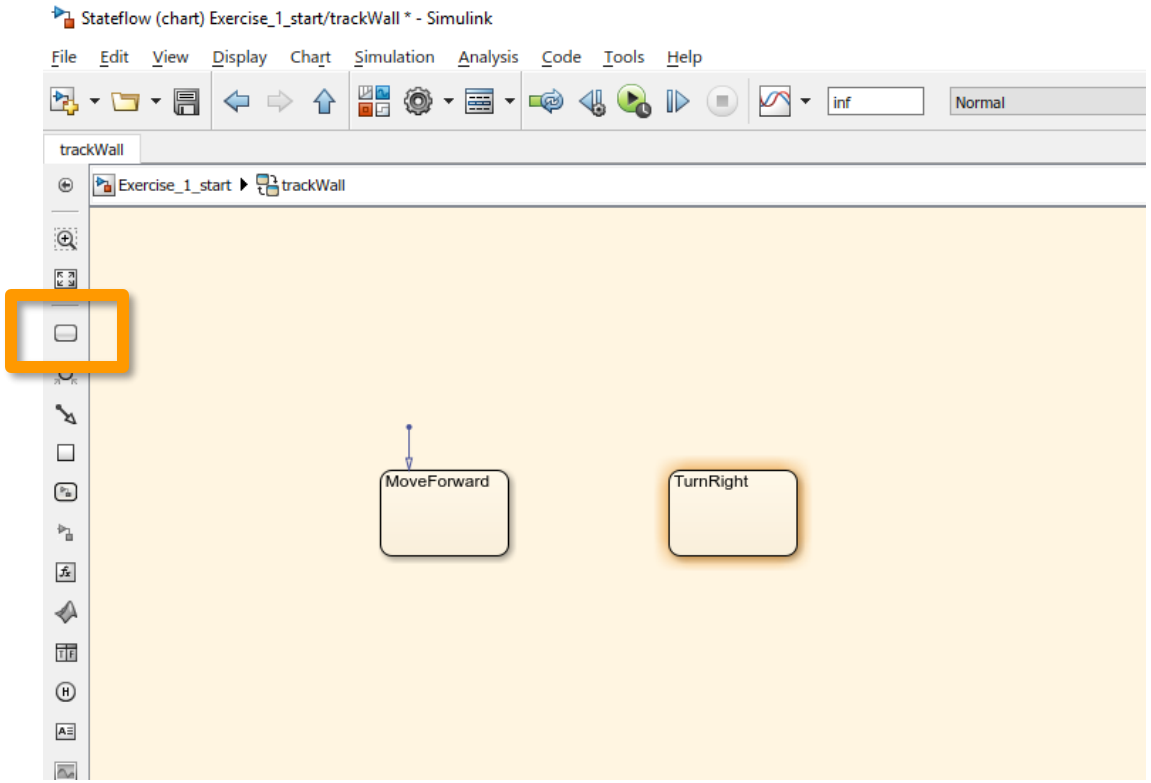
- Chart blocks are also available through the Simulink Library Browser

Adding States

1. Double-click on the chart to open its contents

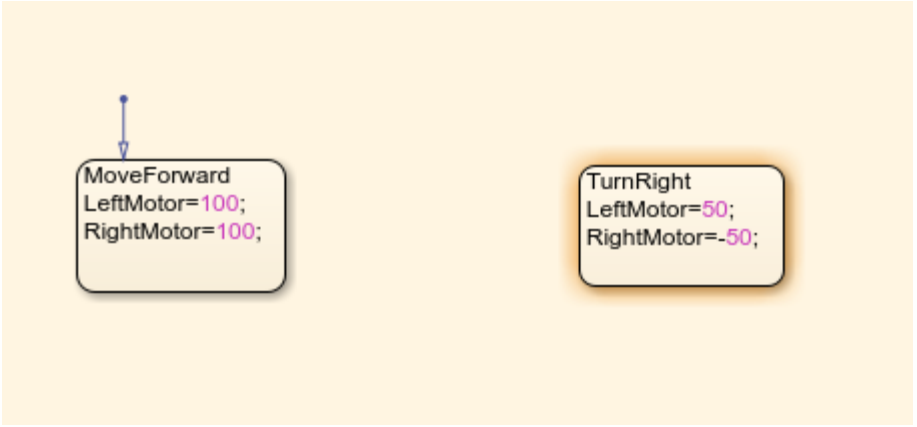
2. Drag two states from the toolbar on the left into the canvas

3. Rename the states “MoveForward” and “TurnRight”



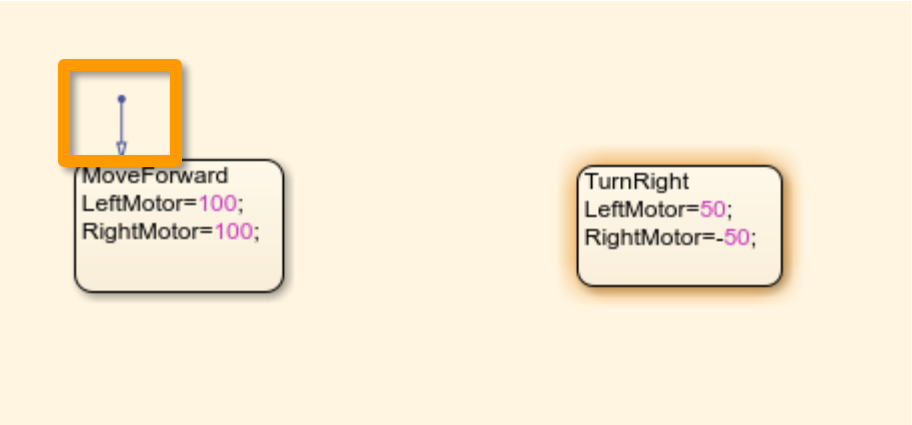
Adding Code

- Each state represents a block of code that will execute when it is active
- Add code to set the motor speeds within each state



Transitions

- Transitions are lines that connect the different operating states and represent the conditions necessary to move between states
- The line that was added to the first state added to the chart is called a default transition.
- A default transition identifies the starting state of the chart

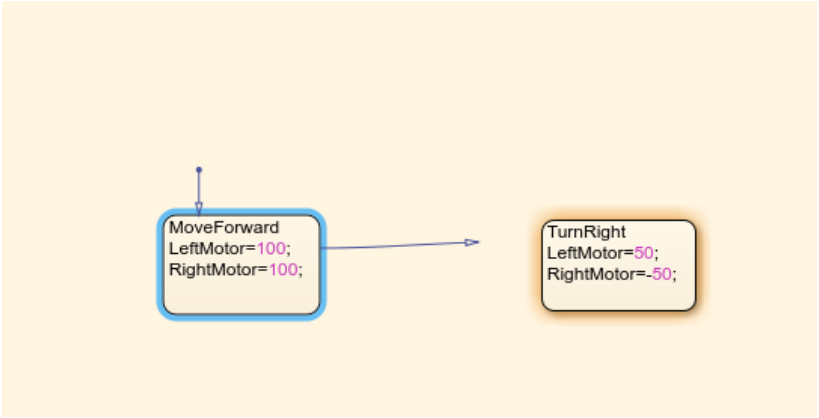


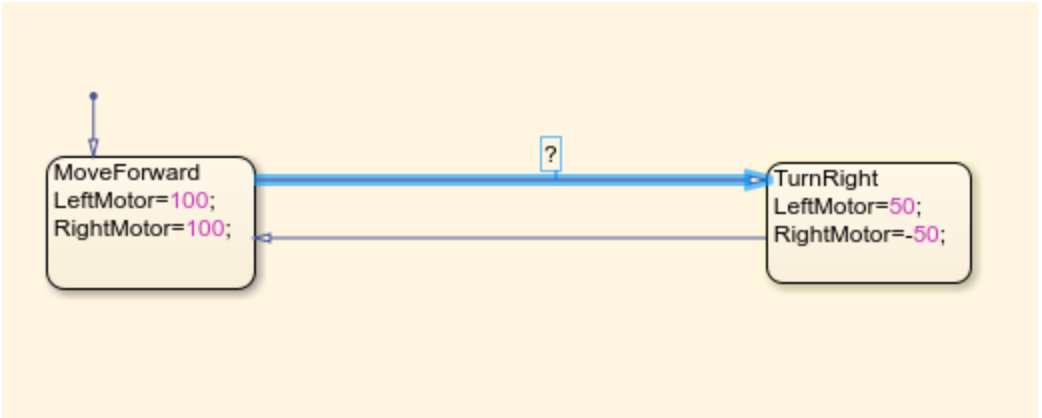
Implementing Transitions

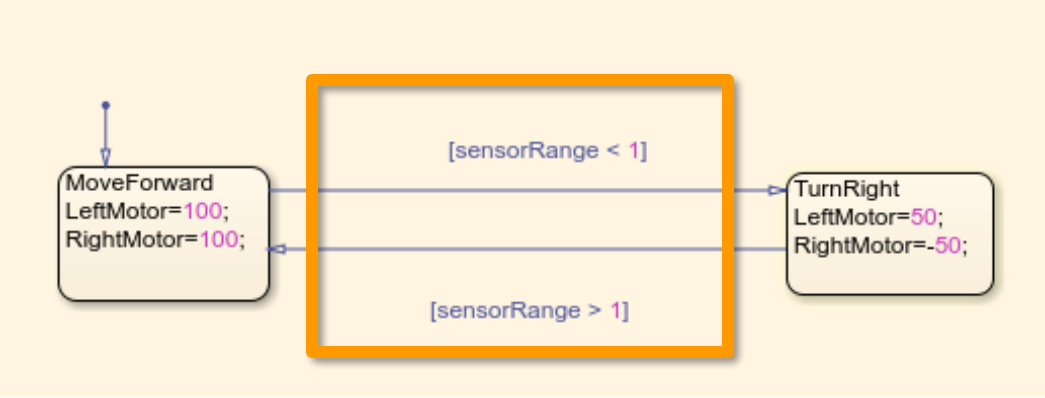
1. Connect the two states by clicking on the edge of a state and dragging a transition to another state

2. Add conditions to the transitions by clicking on them and enclosing statements on brackets [ ]

3. Add some statements to check the signal from the sensor. Use a variable called **“sensorRange”**



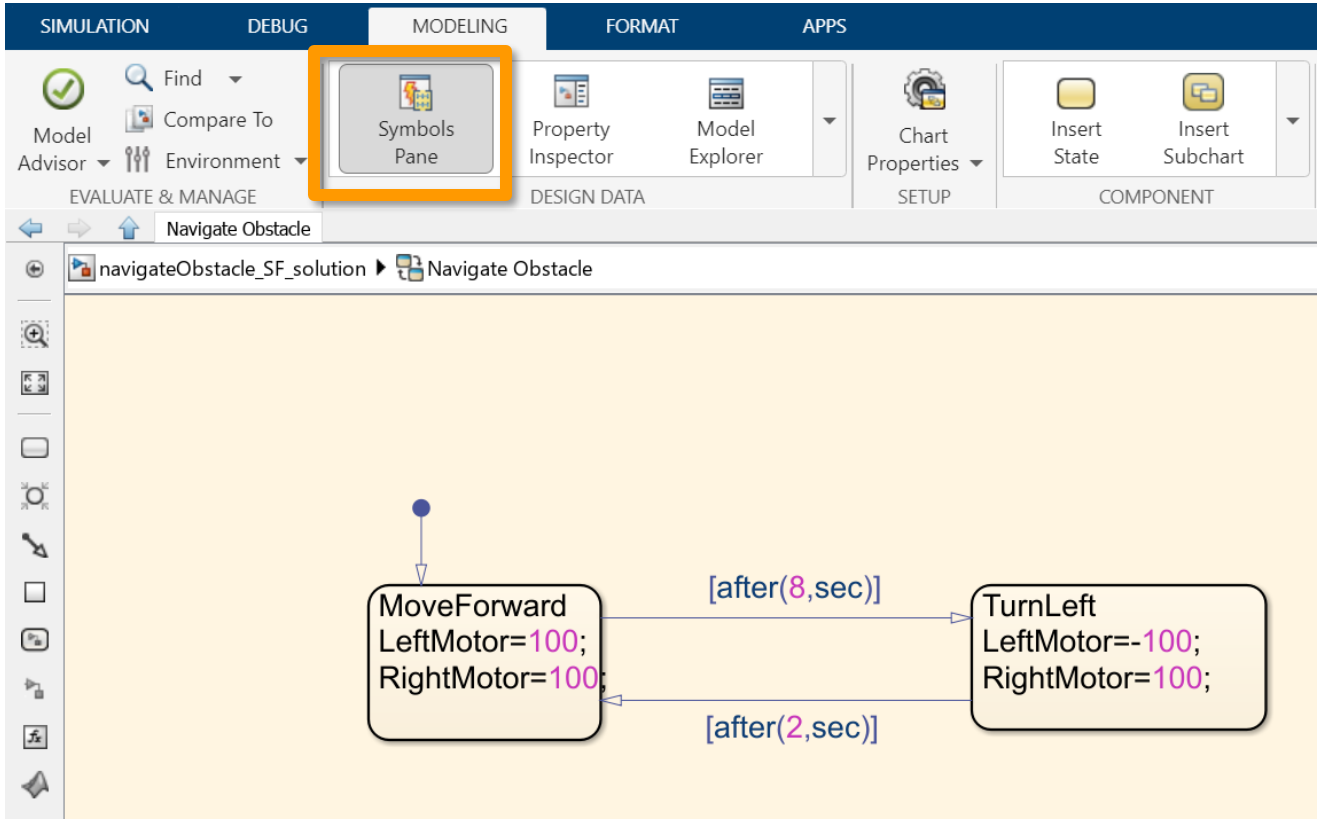




Setting Input/Output Variables

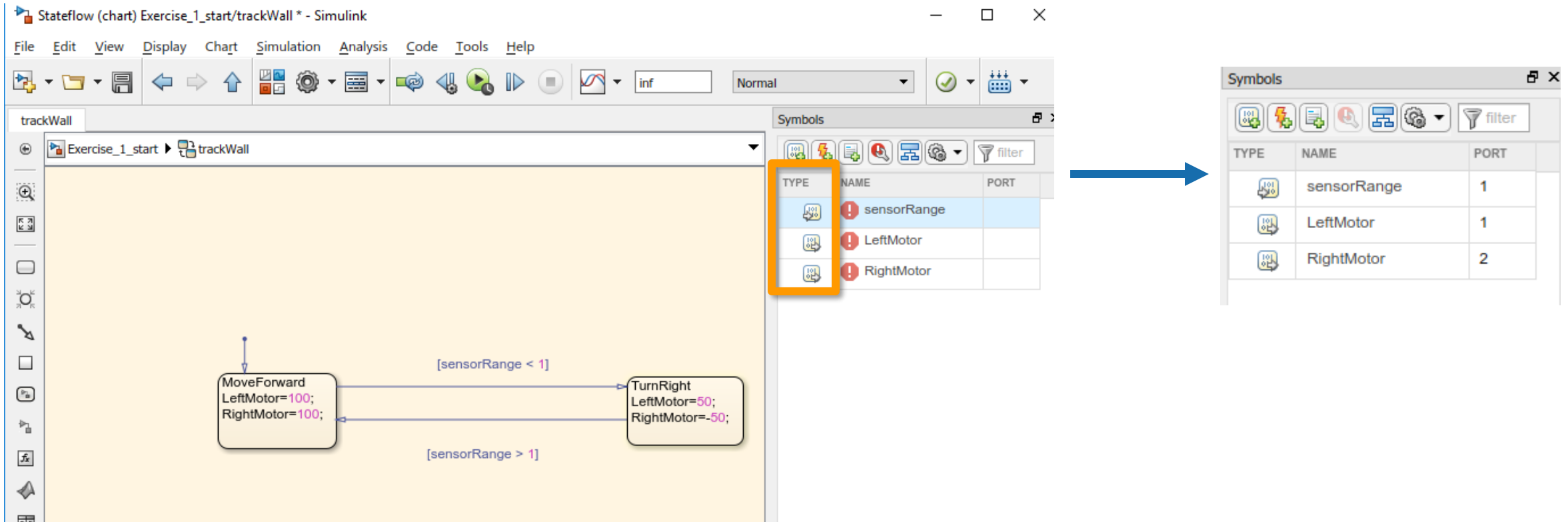
- To create Inputs and Outputs for the chart

1. Go to the “Modeling” tab
2. Select “Symbols Pane”



Setting Input/Output Variables

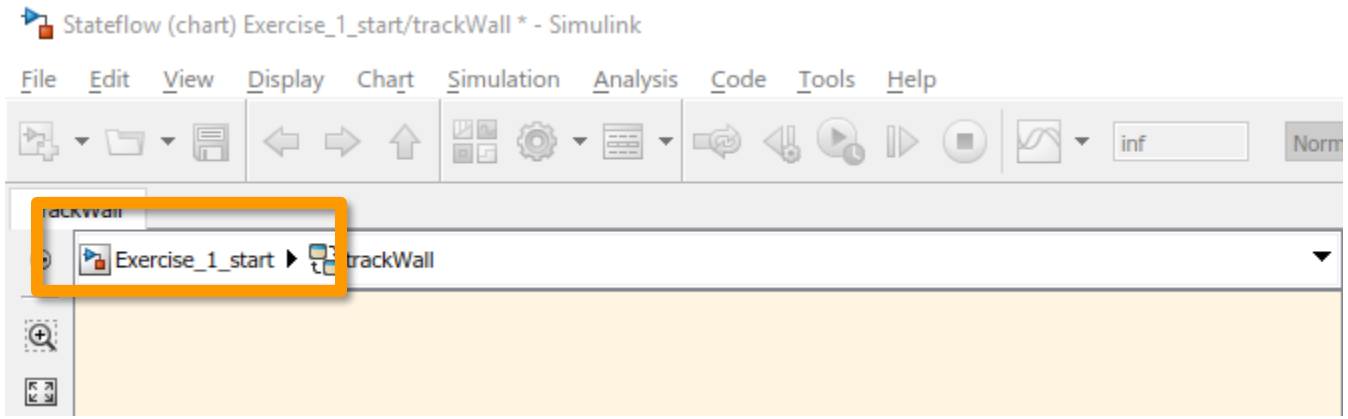
- The variables you are currently using in the chart will appear in the symbols pane
- Click the “Type” icons
- Specify whether these variables are Input Data or Output Data.
- The warning signs next to them should go away.

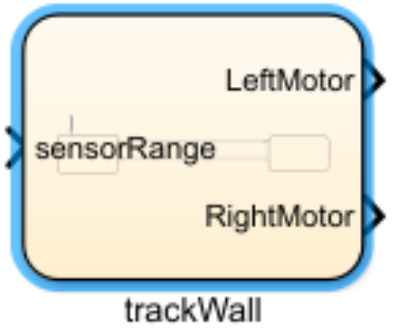


Implementing Wall Tracking

1. Click the title bar below the toolbar to go back to the Simulink model

2. Verify the chart has the specified Inputs and Outputs

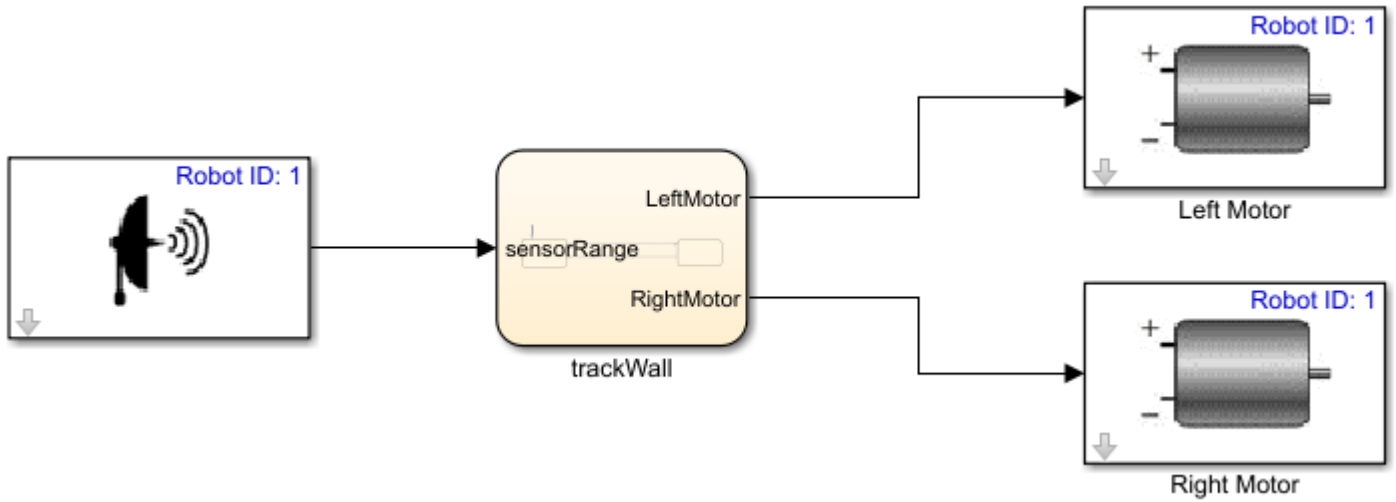




Implementing Wall Tracking

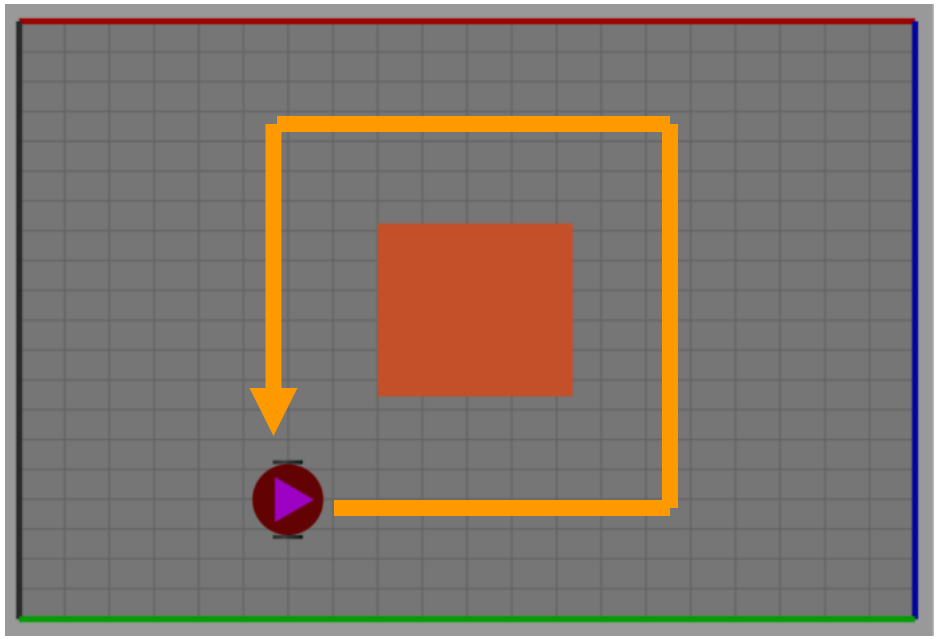
- That's it !

1. Connect the chart to the sensor and motor blocks
2. Run the model
3. Verify the algorithm works correctly ☺



Exercise 2: Navigating an Obstacle

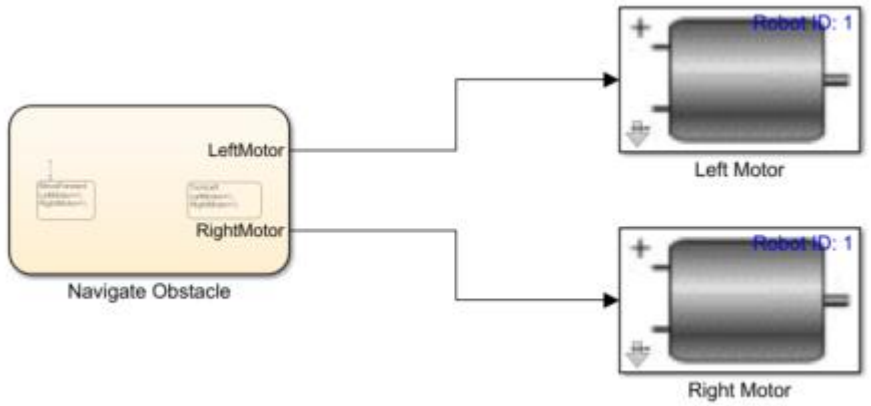
- Program a sequence of events that will make the robot navigate around an obstacle
- Create a Stateflow chart that moves the robot forward and then make it turn over a period of time to navigate around the obstacle in the field

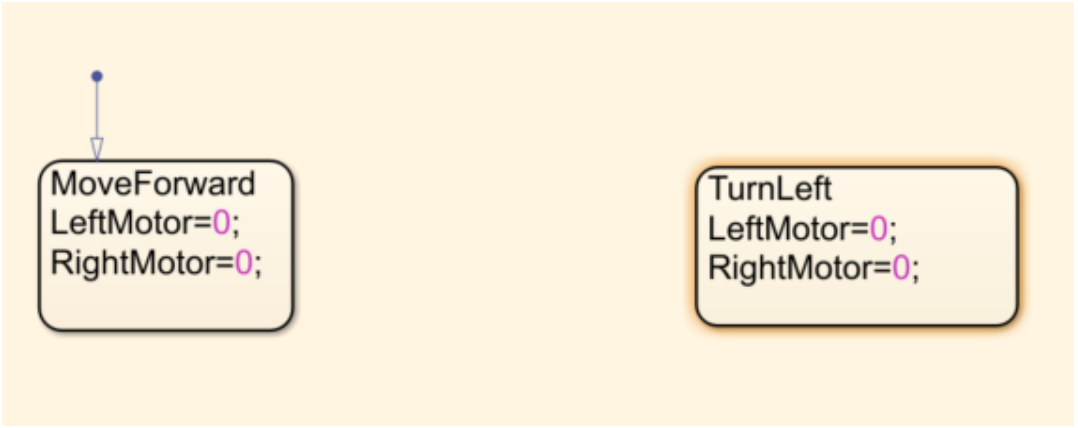


Exercise 2: Avoiding an Obstacle

- Transition between states based on time passed by using the **“after”** condition in a transition
  - **Example:** [after(3,sec)] Will wait 3 seconds before transitioning to the next state

1. Open the model **“navigateObstacle_SF_start.slx”**
2. Set motor values
3. Add transitions with “after” conditions
4. Run model
5. Adjust conditions so the robot navigates around the obstacle

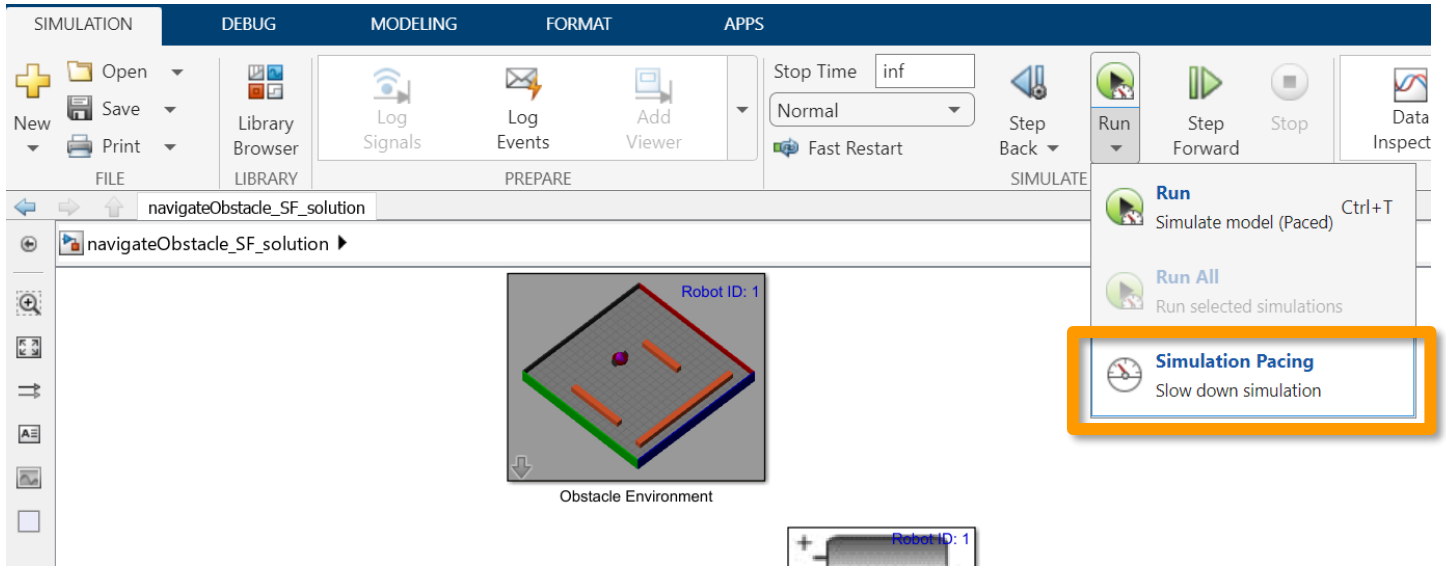


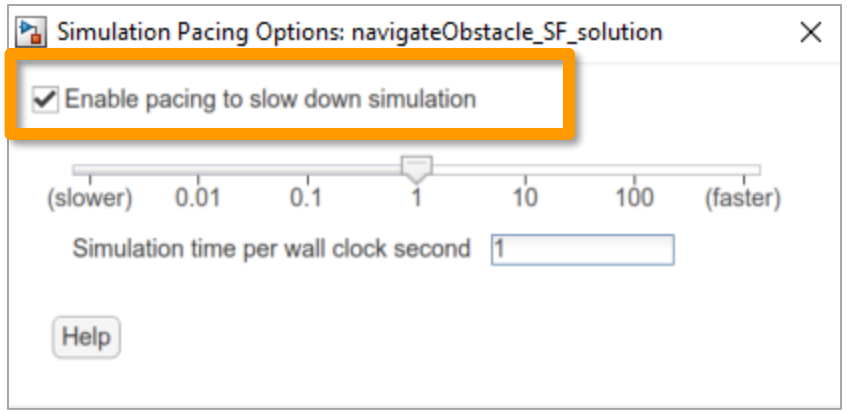


Simulation Pacing

- Make the simulation run faster by removing Simulation pacing

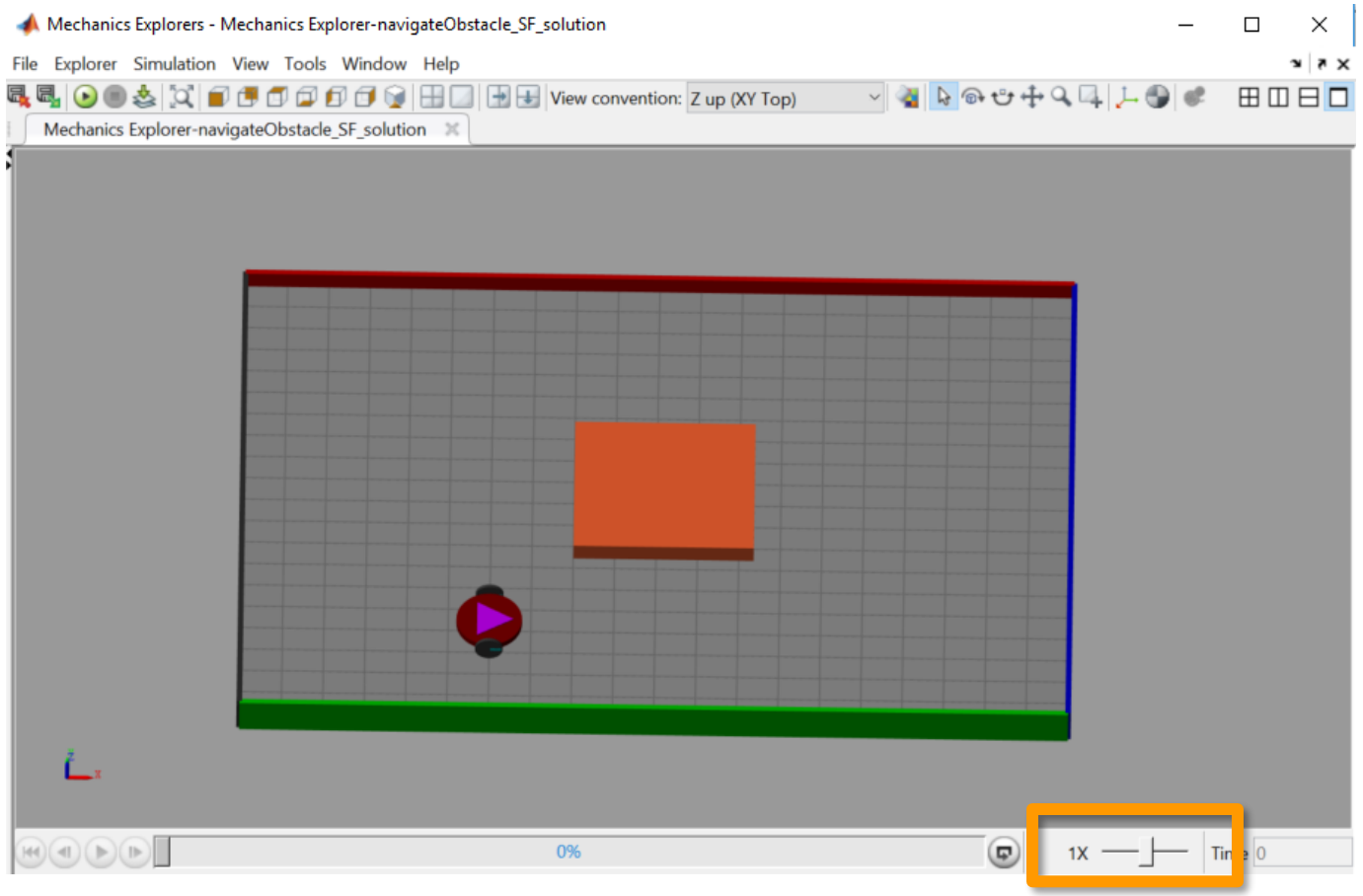
1. In the Simulation tab, select “Run”
2. Select “Simulation Pacing”
3. Disable the checkbox to slow down the simulation





Change Playback Speed of Virtual Environment

- Change the playback speed of the virtual environment using the Playback Speed Slider
- This will help with faster troubleshooting of algorithms



End of Unit 7: Intro to Stateflow

- Congrats !

- Here are some learning outcomes from this unit:
  - How to setup Stateflow charts
  - How to implement states to control robot behavior
  - How to implement time-based (Temporal) logic to automate robots
  - How to speed up simulations for faster troubleshooting