Traffic Light Model Description

Lecture 62, v02

A description of a model developed by The MathWorks

Purpose

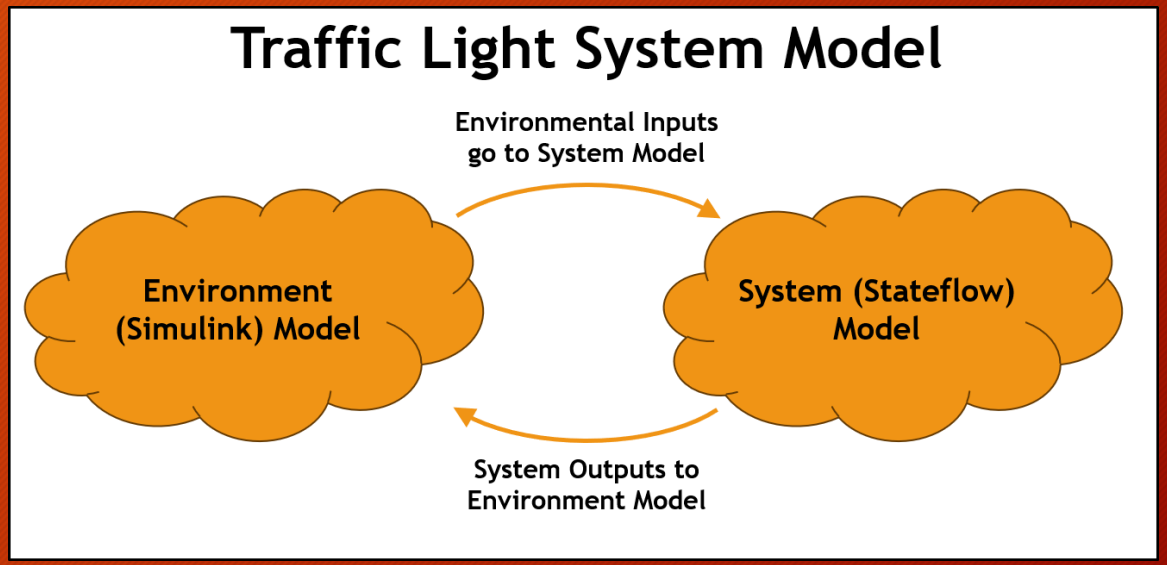
- This lecture describes the Traffic Light system model developed by The Mathworks
- The description of the model is conducted in three sections
  - A top level view of the major model elements (Section 1)
  - A mid-level description of the Traffic Controller Stateflow model (Section 2)
  - A detailed, low-level description of the piece parts that make up the model, and how they communicate with each other (Section 3)
- The vast majority of the material contained herein to describe this model is sourced from various pages on www.mathworks.com
- The copyright applied to this document refers to the arrangement of data collected from The MathWorks in a format that is appropriate to covering the subject in a 1- to 2-week educational module

Section 1

Traffic Light System Overview

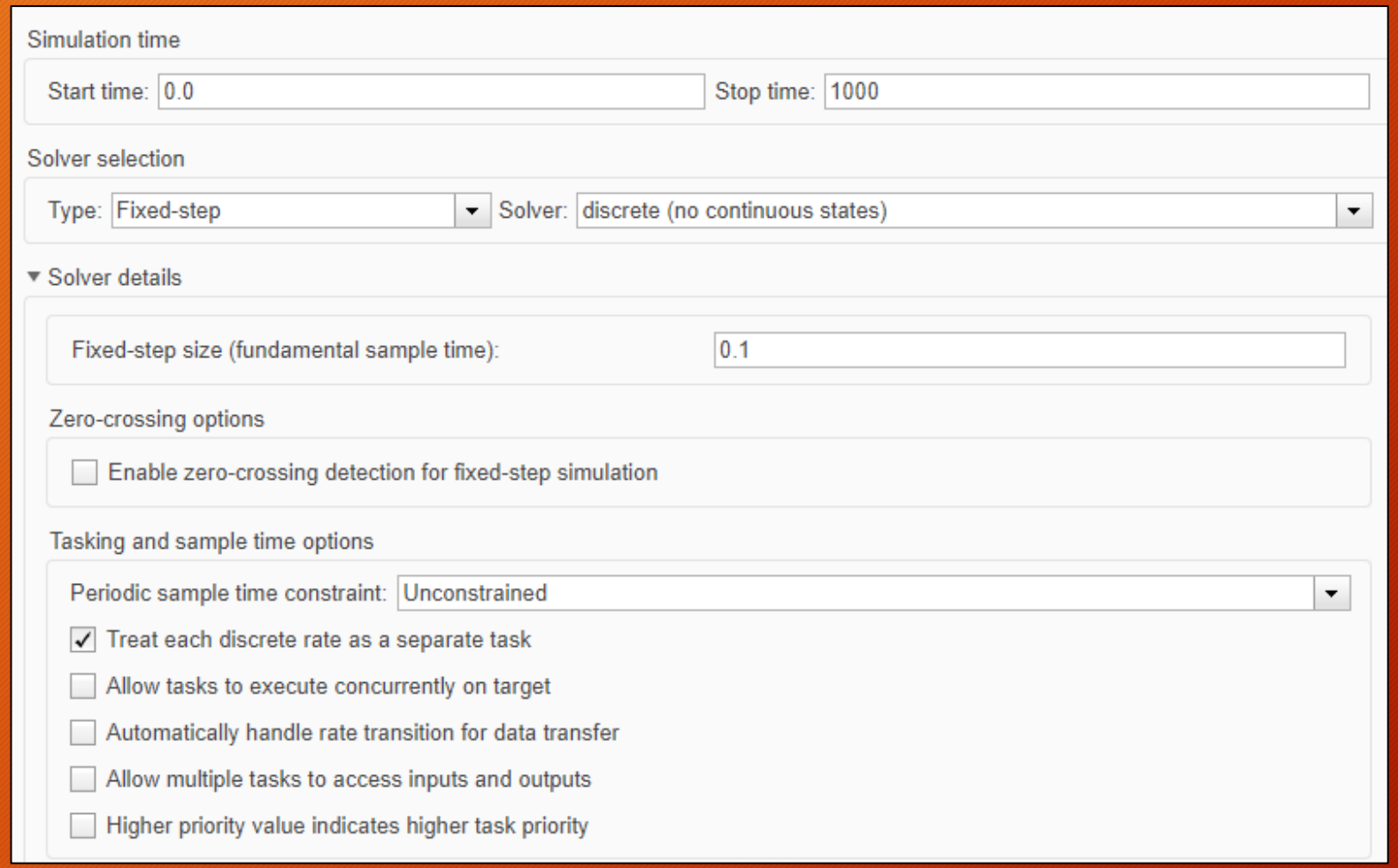
Simulink - Stateflow Association

- The Simulink side of the model serves as the model of the environment within which the System of Interest (SoI) operates
  - This includes the conditions of the traffic intersection being managed by the SoI
- The Stateflow side of the model serves as the model of the system behavioral logic
  - This includes the system that controls the state of th two semaphores directing traffic at the intersection
- An interface between the two exists in order for
  - Environmental conditions to be presented to the SoI
  - System responses and influences to be presented to the environment



Model Settings

- The Simulation Model Settings are as follows:
  - Found in: Modeling -> Model Settings -> Model Settings

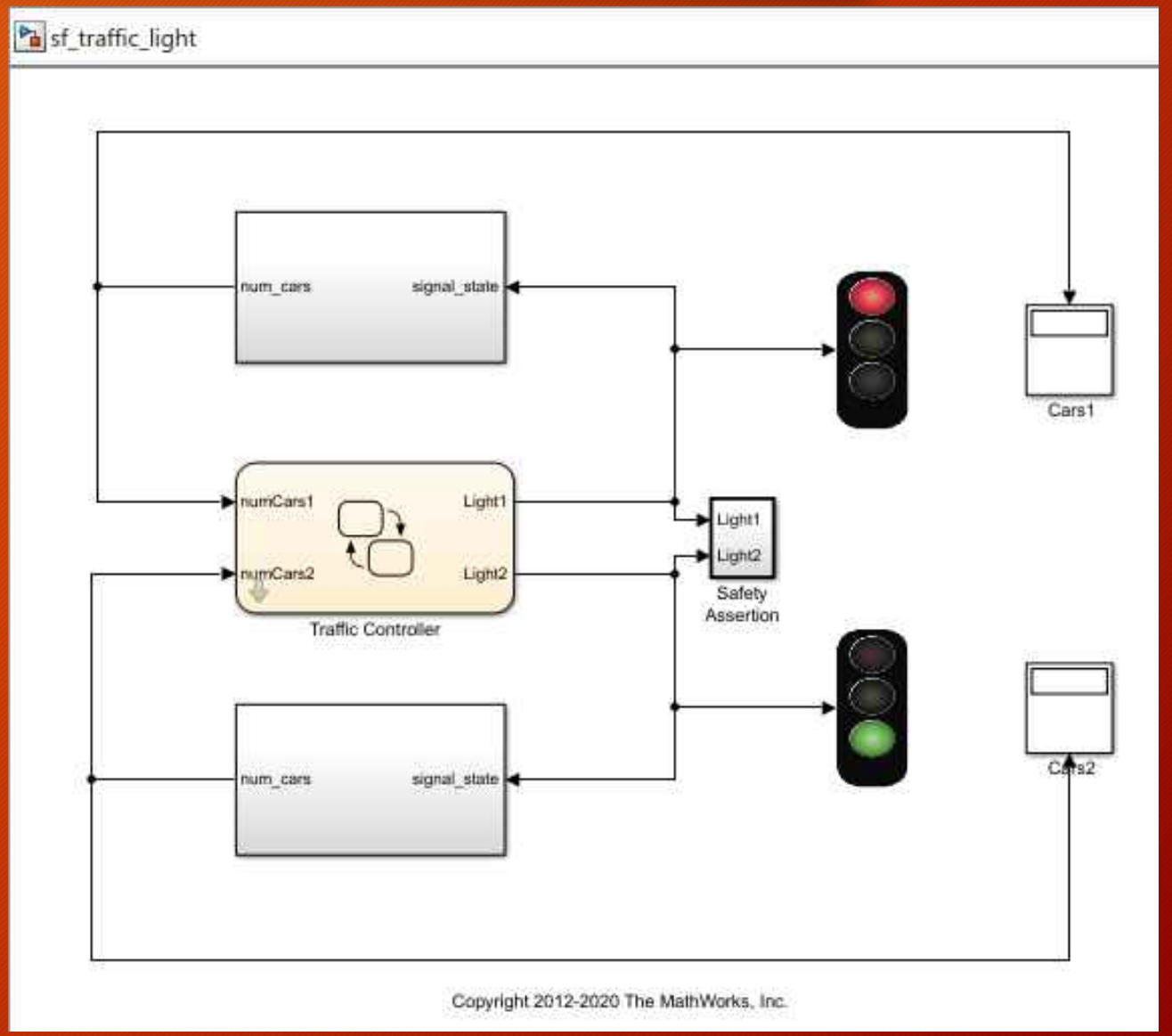


These settings indicate that the simulation runs with a fixed (non-changing) time step of 0.1 seconds per step

Traffic Light Model (sf_traffic_light.slx)

- This model represents a simple Traffic Light system at the intersection of two 1-way roads controlled by a Stateflow model of a Traffic Controller

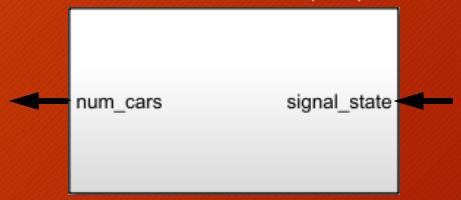
- Shown in the diagram on the right is the Simulink portion of the model
- It includes a Stateflow model that contains logic that controls the state transitions of two semaphores containing three colored lights (Red, Green, Yellow)
- The Stateflow model lies beneath a Statfelow “mask” shown to the right in beige fill



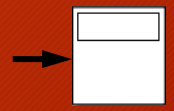
Simulink Model Component Descriptions

- Car Monitors (x 2)
  - Generates cars to occupy the intersection from each direction
  - Monitors the state of the Semaphore in each direction (Red, Yellow, or Green state)
- Car Counters (x 2)
  - Shows the number of cars at the intersection in each direction
- Traffic Controller
  - Stateflow model that executes two Statecharts, one for each direction
  - Based on the number of cars in the intersection, determines the state that each Semaphore should be in
  - Outputs Light 1 and Light 2 which controls which color of light the Semaphore must display
  - Supplies this Semaphore state data to the Car Monitors
- Traffic Semaphores (Lights) (x 2)
  - Implements the display of lights (Red, Yellow, or Green) depending on the signal provided by the Traffic Controller
- Safety Assertion
  - Ensures that two green lights cannot be ON at the same time

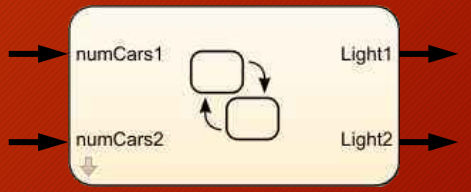
Car Monitor (x2)



Car Counter (x2)



Traffic Controller

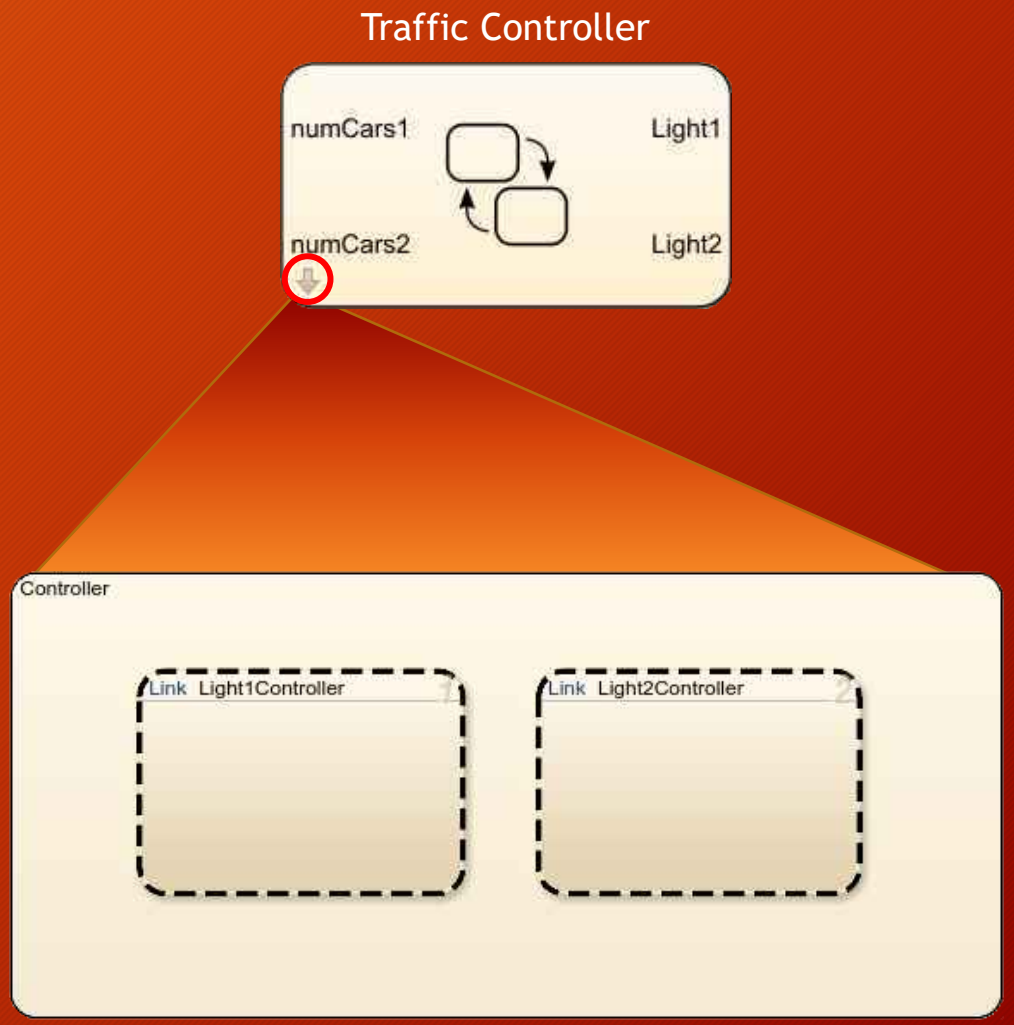


Semaphore (x2)

Safety Assertion

Beneath Traffic Controller “Mask”

- The Stateflow chart Traffic Controller manages two traffic controllers in parallel
- Each controller determines the phase of the downstream traffic light based on
  - Traffic congestion at the intersection
  - An input from Simulink
  - Parameters on the chart’s “mask”
- Beneath the Stateflow “mask” in the Simulink model (the Traffic Controller) lies the Controller
  - This is accessed by clicking on the small arrow at the lower left of the Traffic Controller
- The Controller contains two Subcharts, one for each traffic direction
  - Light1Controller
  - Light2Controller



Section 2

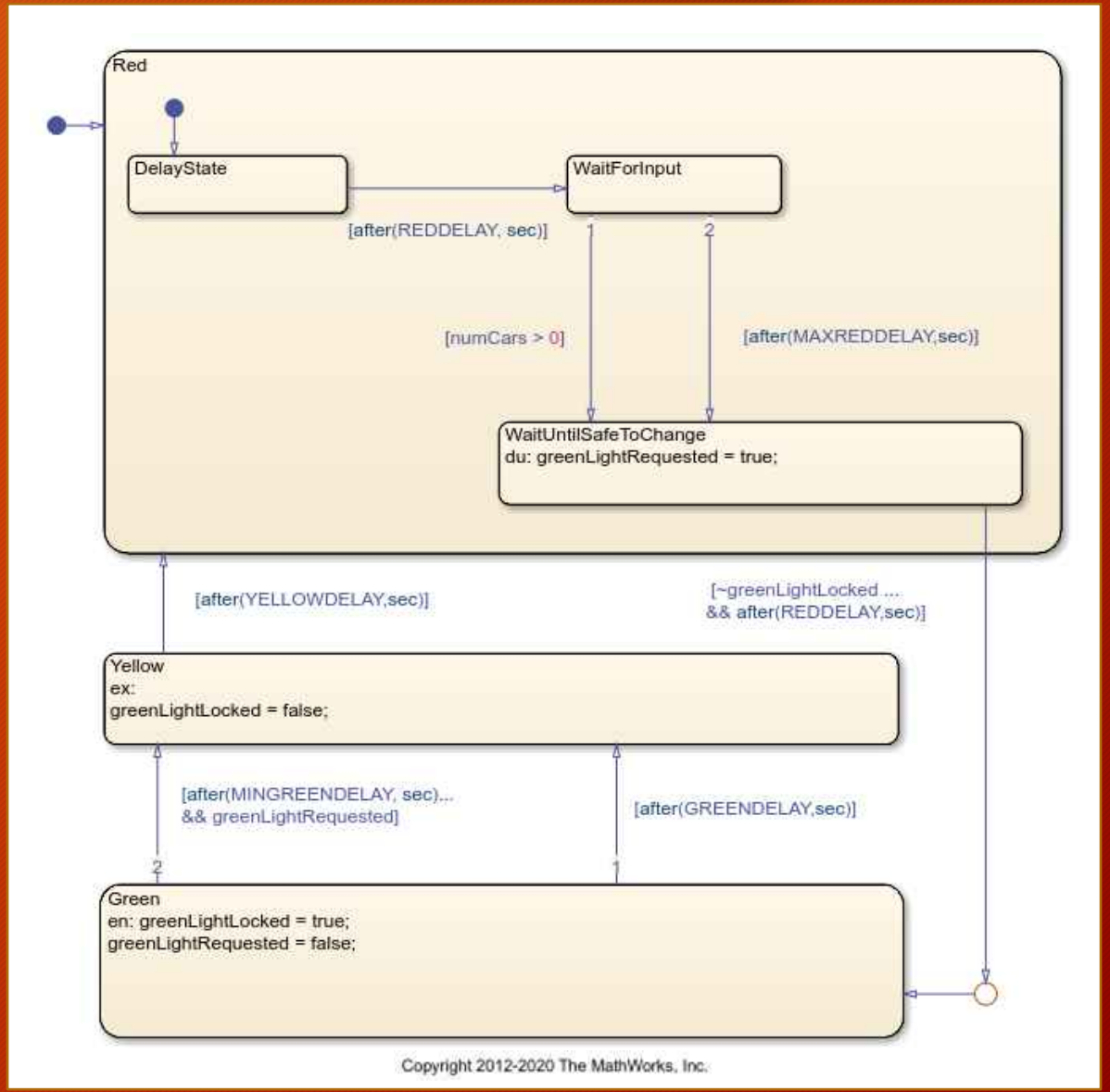
Traffic Controller Stateflow Model

Stateflow Model of Traffic Light State Behavior

- This Stateflow model for Light1Controller and Light2Controller are identical
- The two subcharts have an identical hierarchy consisting of three child states: Red, Yellow, and Green
- The output data Light1 and Light2 (not shown in this view) correspond to the active child states in the subcharts
- These output signals:
  - Determine the phase (light color) of the Semaphores
  - Contribute to the generation of cars waiting at each light
  - Drive the Safety Assertion subsystem verifying that the two traffic lights are never simultaneously green

When on Statechart is guarding certain transitions, it is checking the conditions of the other Statechart

Example, in going from Red to Green:
[~greenLightLocked] is determining whether the other Statechart has gone Red, since the data store memory (flag) greenLightLocked is cleared on exit from the Yellow state



Light Controller Parameters Definition

- This table defines the meanings of the parameters used in the two Light Controllers

Parameter	Preset Value	Definition
REDDELAY	6 seconds	Length of time before the controller begins to check for cars at the intersection Also, minimum length of time before the traffic light can turn green after the controller requests a green light
MAXREDDELAY	360 seconds	Maximum length of time that the controller checks for cars before requesting a green light
GREENDELAY	180 seconds	Maximum length of time that the traffic light remains green
MINGREENDELAY	120 seconds	Minimum length of time that the traffic light remains green
YELLOWDELAY	15 seconds	Length of time that the traffic light remains yellow

Behavior of Traffic Controller Subcharts

- Each traffic controller cycles through its child states, from Red to Green to Yellow and back to Red
  - The output signals Light1 and Light2 indicate which state is active at any given time
- Red State
  - When the Red state becomes active, the traffic light cycle begins
  - After a short delay, the controller checks for cars waiting at the intersection
  - If it detects at least one car, or if a fixed length of time elapses, then the controller requests a green light by setting greenLightRequest to true
  - After making the request, the controller remains in the Red state for a short length of time until it detects that the other traffic signal is red [~greenLightLocked]
  - The controller then makes the transition to Green
- Green State
  - When the Green state becomes active, the controller cancels its green light request by setting greenLightRequest to false
  - The controller sets greenLightLocked to true, preventing the other traffic signal from turning green
  - After some time, the controller checks for a green light request from the other controller [greenLightRequested]
  - If it receives a request, or if a fixed length of time elapses, then the controller transitions to the Yellow state
- Yellow State
  - Before transitioning to the Red state, the controller remains in the Yellow state for a fixed amount of time
  - When the Yellow state becomes inactive, the controller sets greenLightLocked to false, indicating that the other traffic light can safely turn green
  - The traffic light cycle then begins again

Behavior of Light Controllers

- This table details the activity during the first 300 seconds of the simulation

Time	Description	Light 1	Light2	greenLightRequested	greenLightLocked
t = 0	At the start of the simulation, both traffic lights are red	Red	Red	false	false
t = 6	After 6 seconds (REDDELAY), there are cars waiting in both streets Both traffic lights request a green light by setting greenLightRequested = true	Red	Red	true	false
t = 12	After another 6 seconds (REDDELAY): · Light 1 turns green, setting greenLightLocked = true and greenLightRequested = false · Light 2 requests a green light by setting greenLightRequested = true	Green	Red	false, then true	true
t = 132	After 120 seconds (MINGREENDELAY): · Light 1 turns yellow	Yellow	Red	true	true

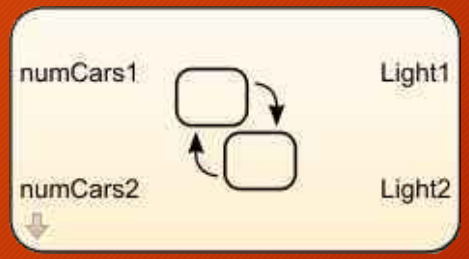
Behavior of Light Controllers (continued)

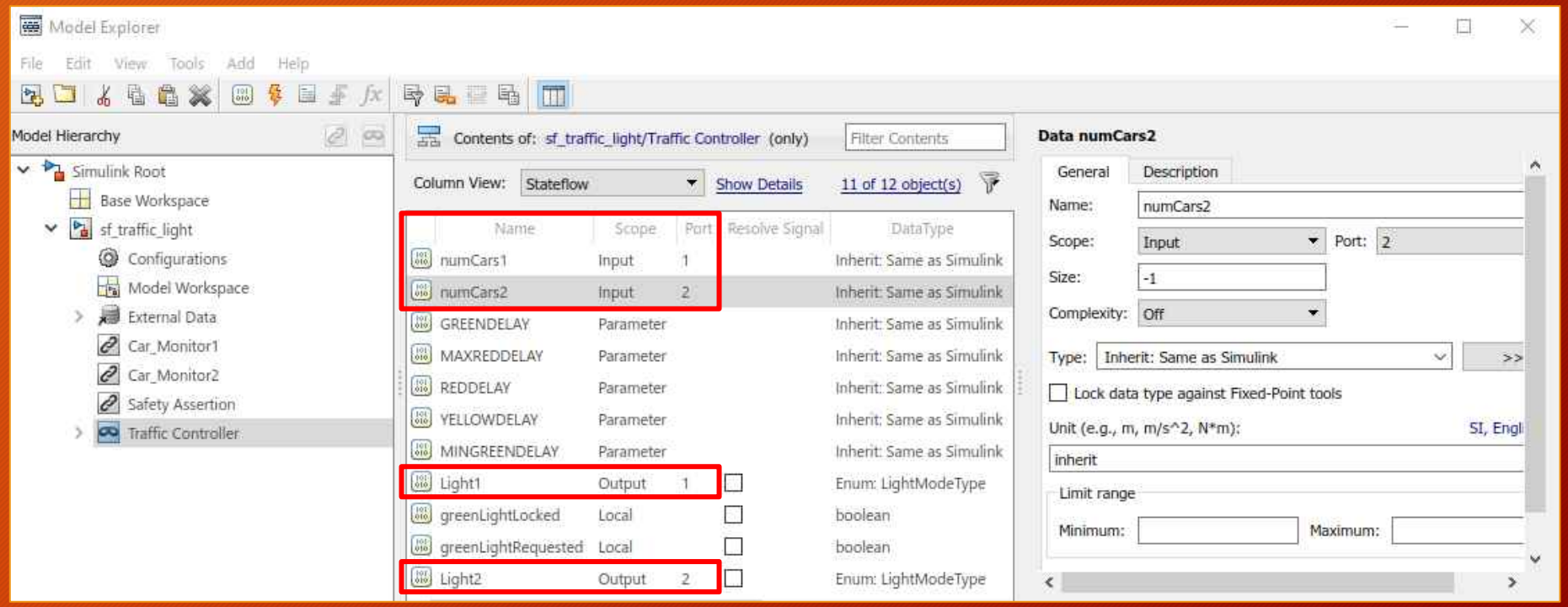
Time	Description	Light 1	Light2	greenLightRequested	greenLightLocked
t = 147	After 15 seconds (YELLOWDELAY): · Light 1 turns red, setting greenLightLocked = false · Light 2 turns green, setting greenLightLocked = true and greenLightRequested = false	Red	Green	false	false, then true
t = 153	After 6 seconds (REDDELAY), Light 1 requests a green light by setting greenLightRequested = true	Red	Green	true	true
t = 267	Light 2 turns yellow 120 seconds (MINGREENDELAY) after turning green	Red	Yellow	true	true
t = 282	After 15 seconds (YELLOWDELAY): · Light 2 turns red, setting greenLightLocked = false · Light 1 turns green, setting greenLightLocked = true and greenLightRequested = false	Green	Red	false	false, then true
t = 288	After 6 seconds (REDDELAY), Light 2 requests a green light by setting greenLightRequested = true	Green	Red	true	true

The cycle repeats until the simulation ends at t = 1000 seconds

Simulink-Stateflow Interface

- Simulink communicates with Stateflow by exchanging data items at ports on the Stateflow “mask” (shown to the right)
- In this example, Simulink provides the number of cars waiting at an intersection to use as an input to the Stateflow model
- Stateflow provides two outputs to control the light selection of the two Semaphores that control traffic in each of the two directions



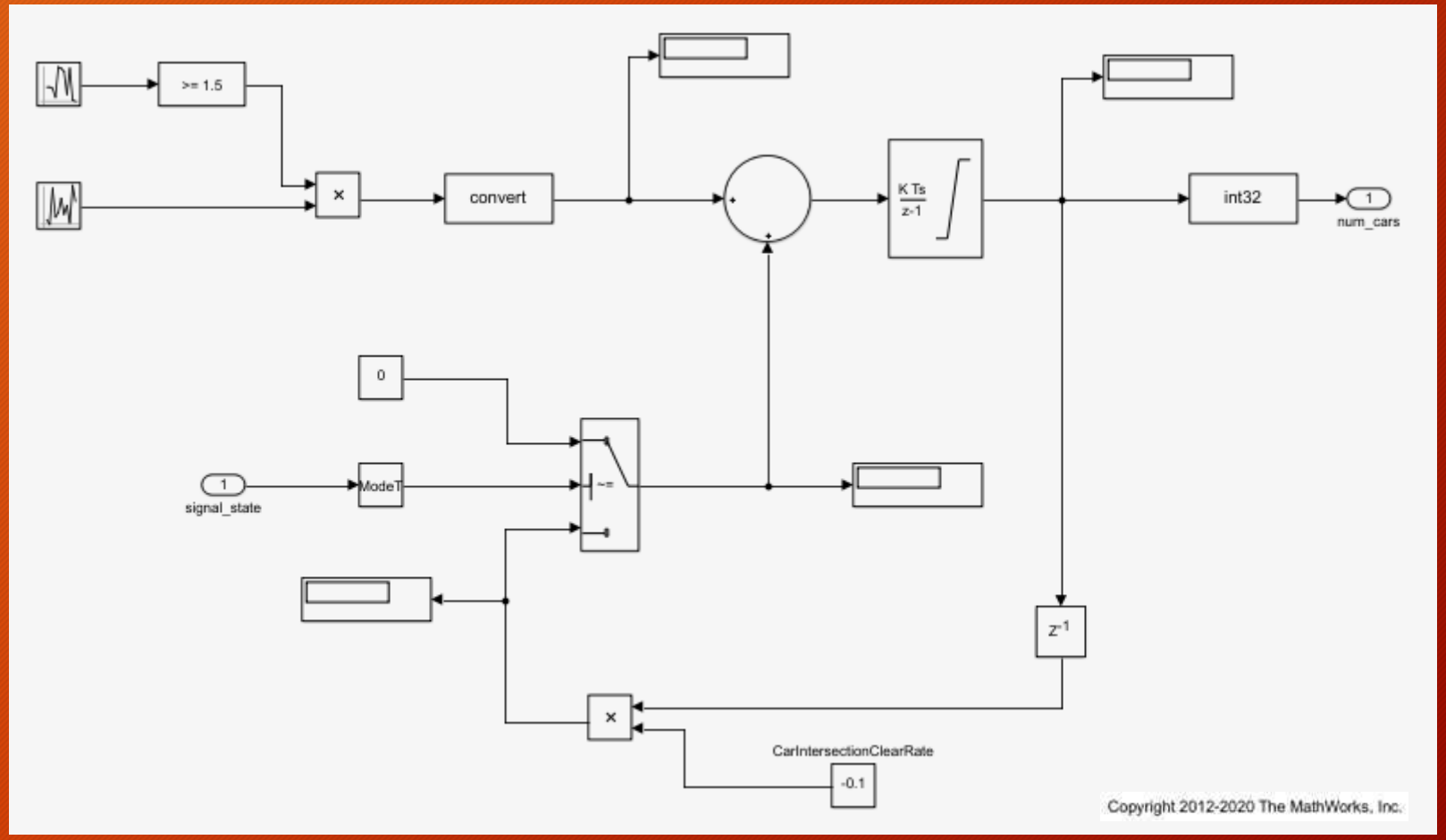


Section 3

Detailed Simulink Model Component Descriptions

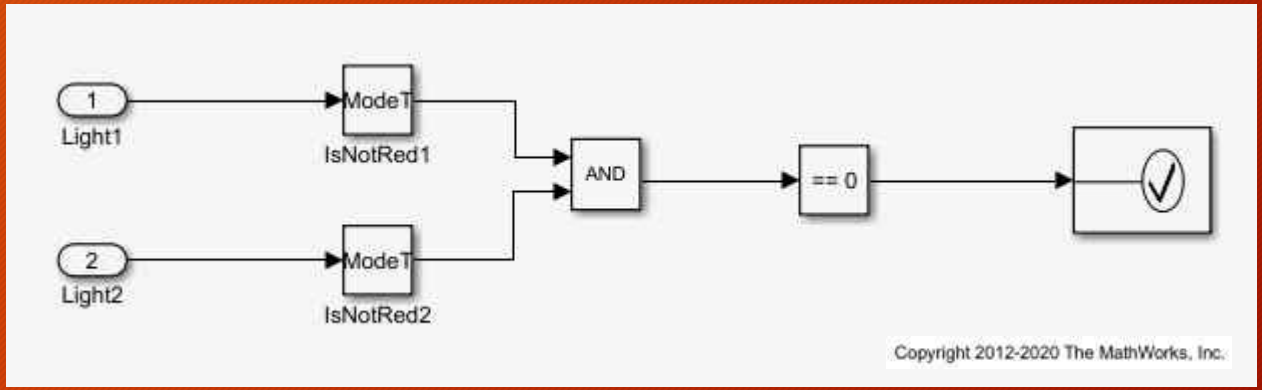
Simulink Model Car Monitor Component

- Detailed analysis of this component TBD



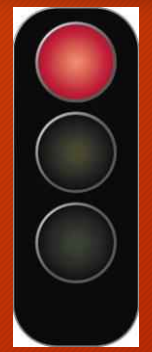
Simulink Model Safety Assertion Component

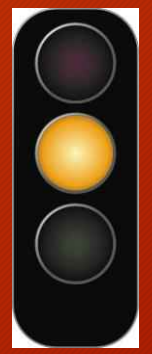
- Detailed analysis of this component TBD

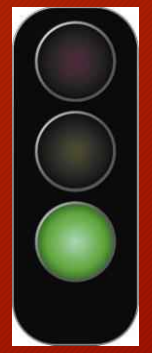


Simulink Semaphore Component

- Detailed analysis of this component TBD
- A MATLAB s-function controls the selection of these three .jpg images to display based on the outputs Light1 and Light2 from the Stateflow model







References

- Source of information from The MathWorks
  - https://www.mathworks.com/help/stateflow/gs/active-state-data.html
  - https://www.mathworks.com/help/stateflow/ug/modeling-an-intersection-of-two-1-way-streets-using-stateflow.html
- Road intersection images adapted from
  - https://www.proprofs.com/quiz-school/story.php?title=oduymjey0z1j