MathWorks
@MathWorks

Share the EXPO experience
#MATLABEXPO

Welcome to our workshop **Model Autonomous Navigation of a Mars Rover**

We'll get started soon.
In the meantime:

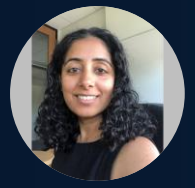
- Use the chat to share from where you are joining us today
- Use the Handouts tab to download instructions for accessing files

MATLAB EXPO

Model Autonomous Navigation of a Mars Rover

Harshal Upadhyay

Kritika Ramani



MathWorks®

MATLAB EXPO

Model Autonomous Navigation of a Mars Rover

Harshal Upadhyay

Chinmayi Lanka

MathWorks

MATLAB EXPO

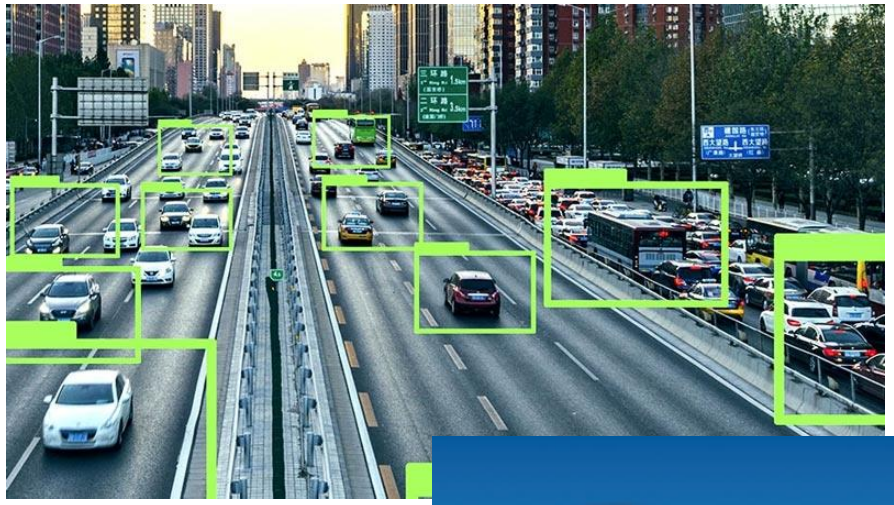
Model Autonomous Navigation of a Mars Rover

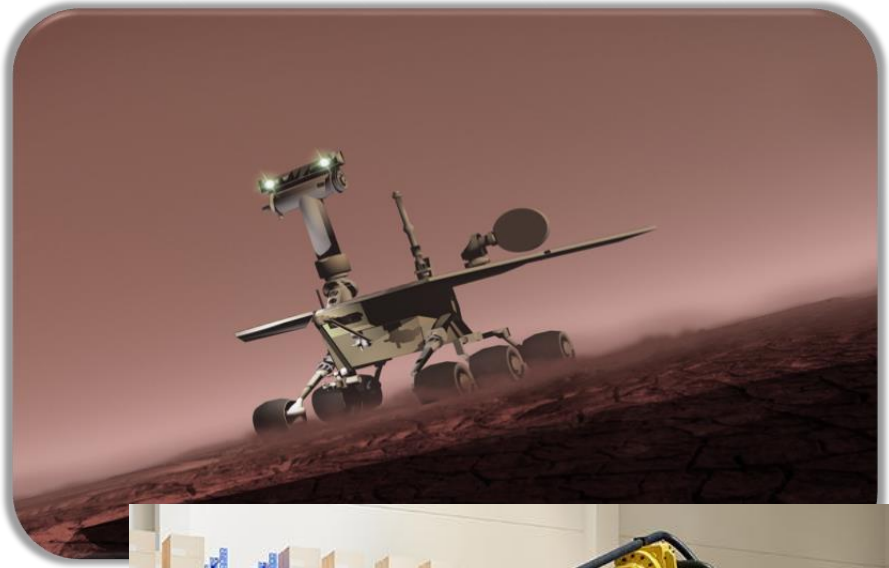
Chinmayi Lanka

Ameya Rajendra Godbole

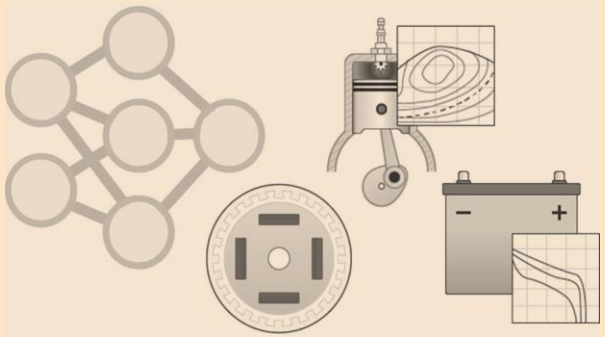
MathWorks

The rise of autonomous systems





Challenges



System Complexity

Decision Making

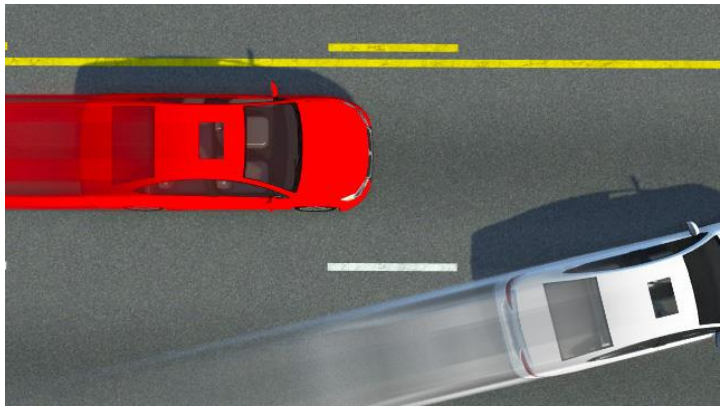
Safety and Reliability

Simulations can play a key role in developing and validating autonomous capabilities successfully

Expensive Hardware

Uncharted territories

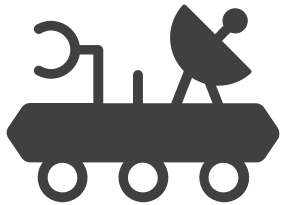
Complex Scenarios



The problem we'll be tackling in this workshop

Simulate navigation of an autonomous rover on a virtual Mars environment

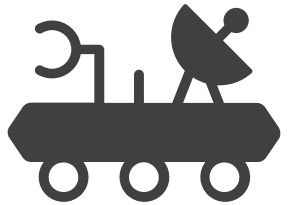
Before we start, let's prepare to participate in this workshop



Use your laptop and a web browser to run exercises

Download instructions using the "Handouts" Tab

A group of TAs will assist with questions



How to get assistance during the workshop

BigMarker
(**use Q&A**)

Today we will be using **MATLAB Online**

https://tinyurl.com/ExpoMarsRover

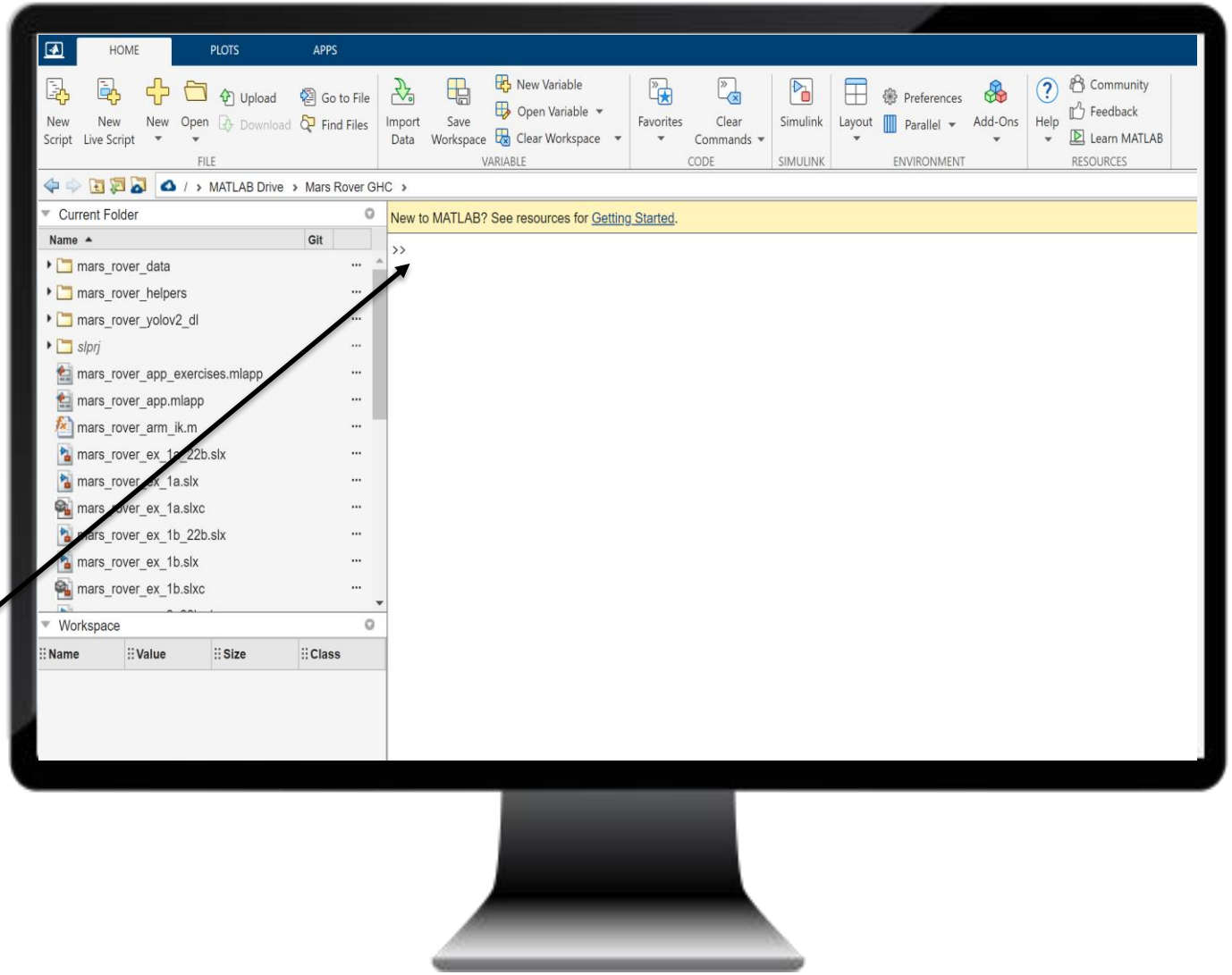
Open MATLAB Online

Once MATLAB Online is open,
maximize the browser window

We also strongly recommend

- **100% zoom for your browser**
- **Do not use Dark Mode in the browser**

```
>> mars_rover_startup
```

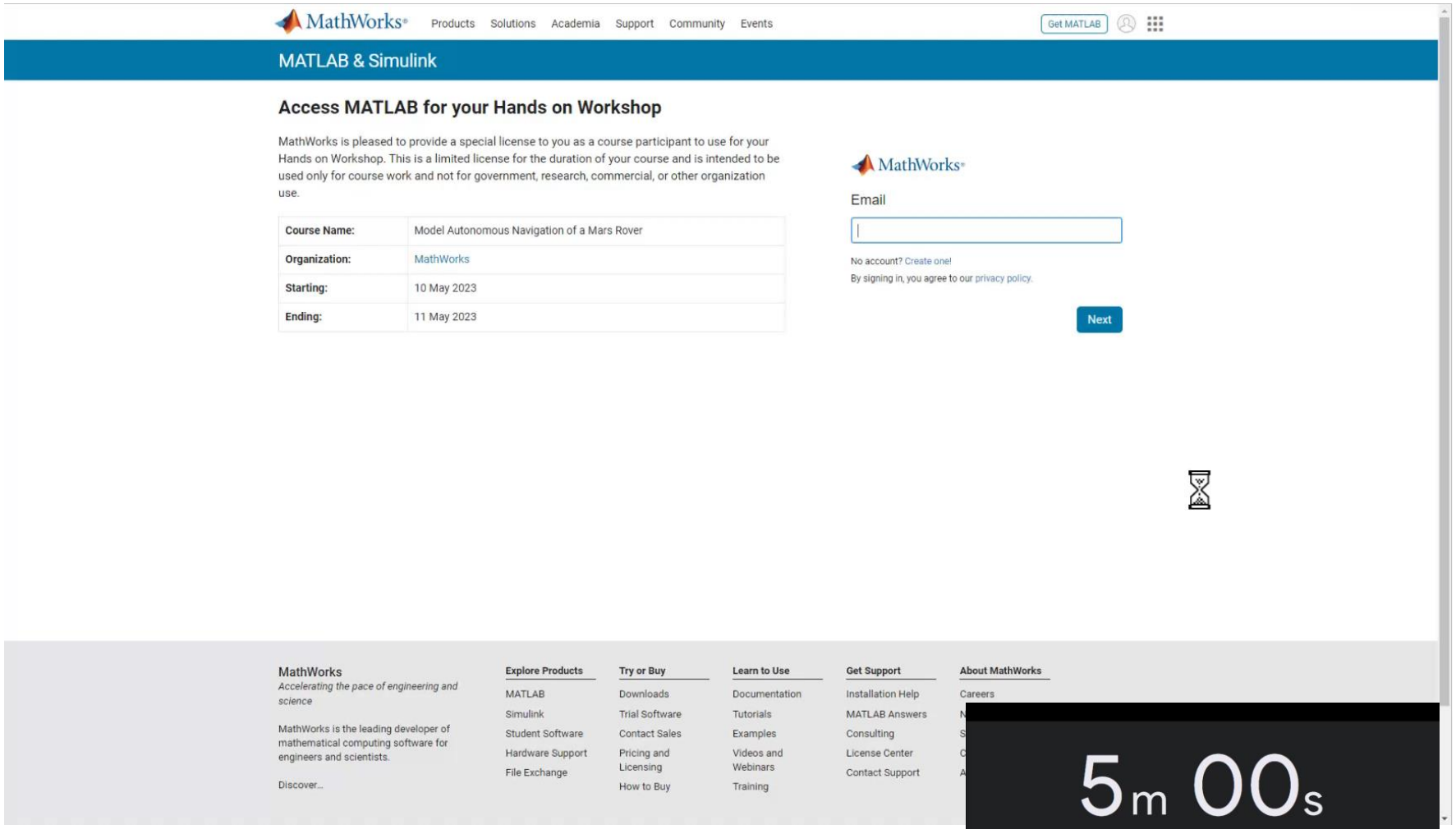


Get Access And Open Workshop exercises

MATLAB Online link

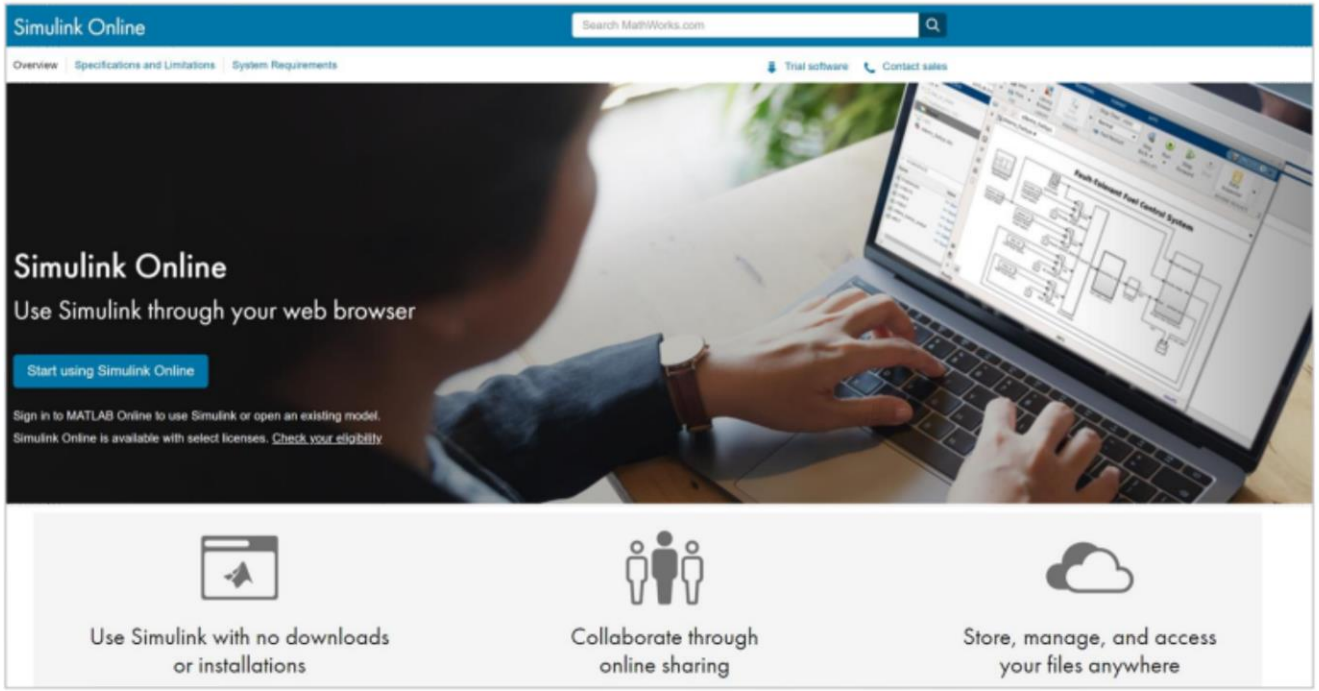
https://tinyurl.com/ExpoMarsRover

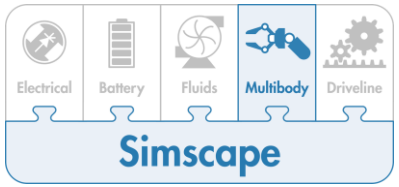
```
>> mars_rover_startup
```



Products used in this workshop

Today we will be using **Simulink Online**

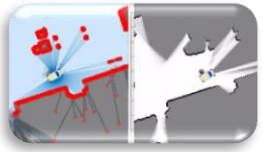




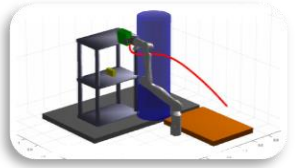
Simscape™

Deep Learning Toolbox™

Computer Vision Toolbox™

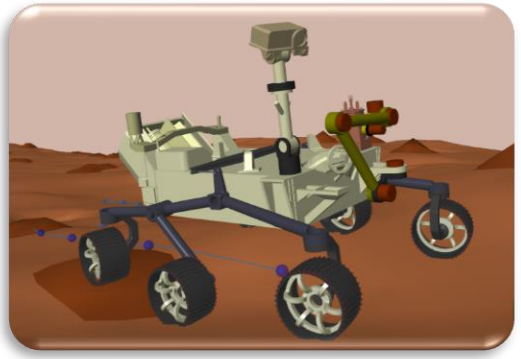


Navigation Toolbox™

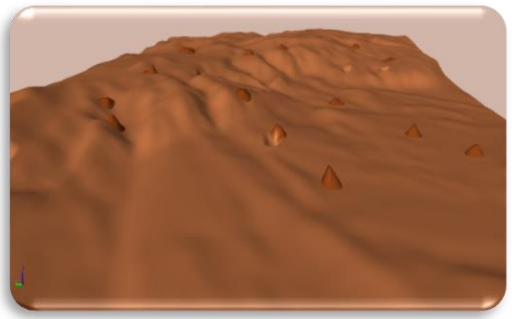


Robotics System Toolbox™

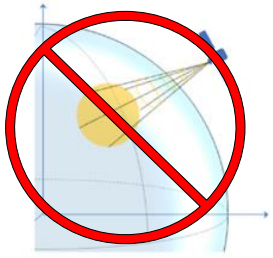
Key capabilities

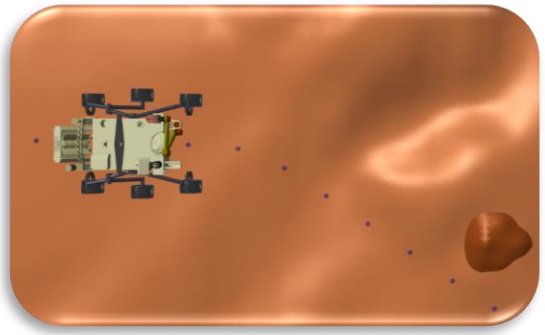


Model a rover

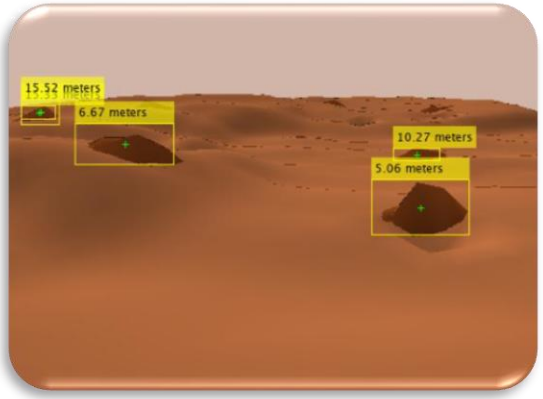


Virtual terrain

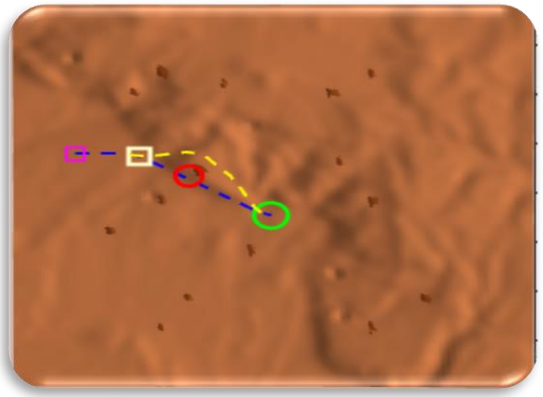




Position estimation

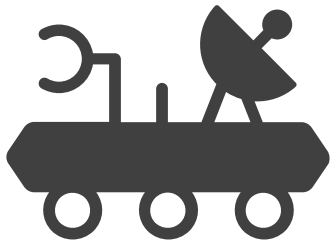


Hazard detection

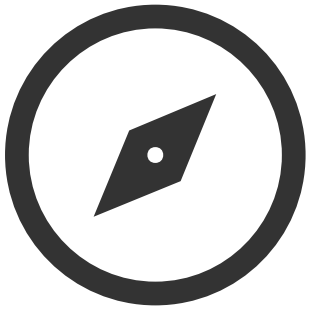


Path Planning

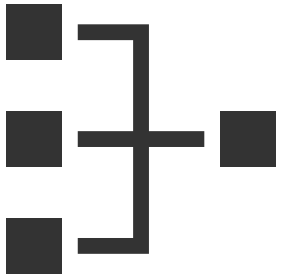
Our workshop today is split into multiple tasks



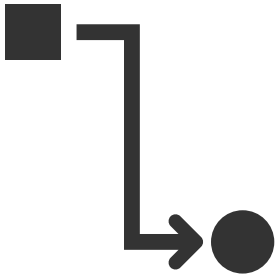
Who am I?
Rover Model



Where am I?
Calibration

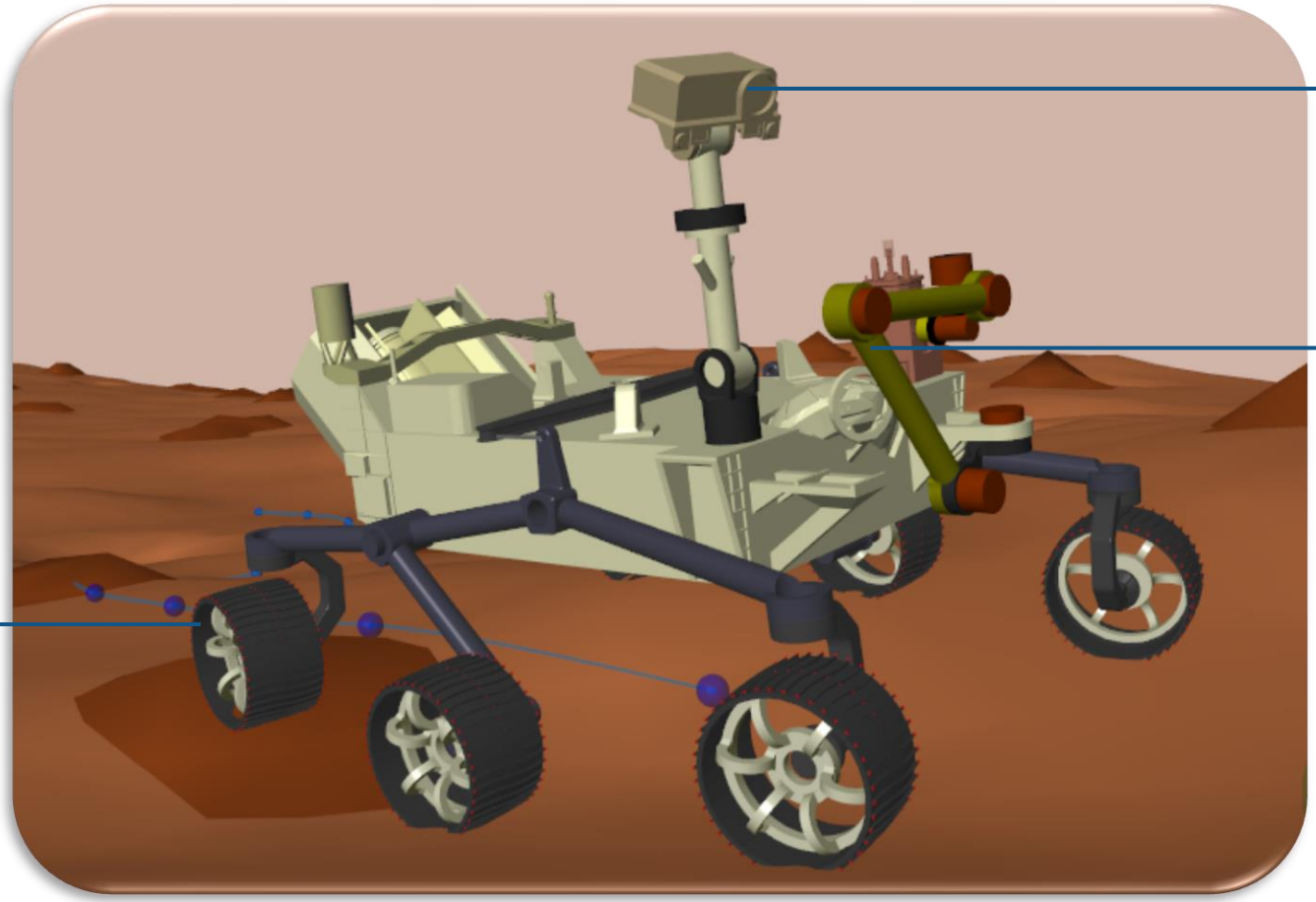


What do I see?
Object Detection

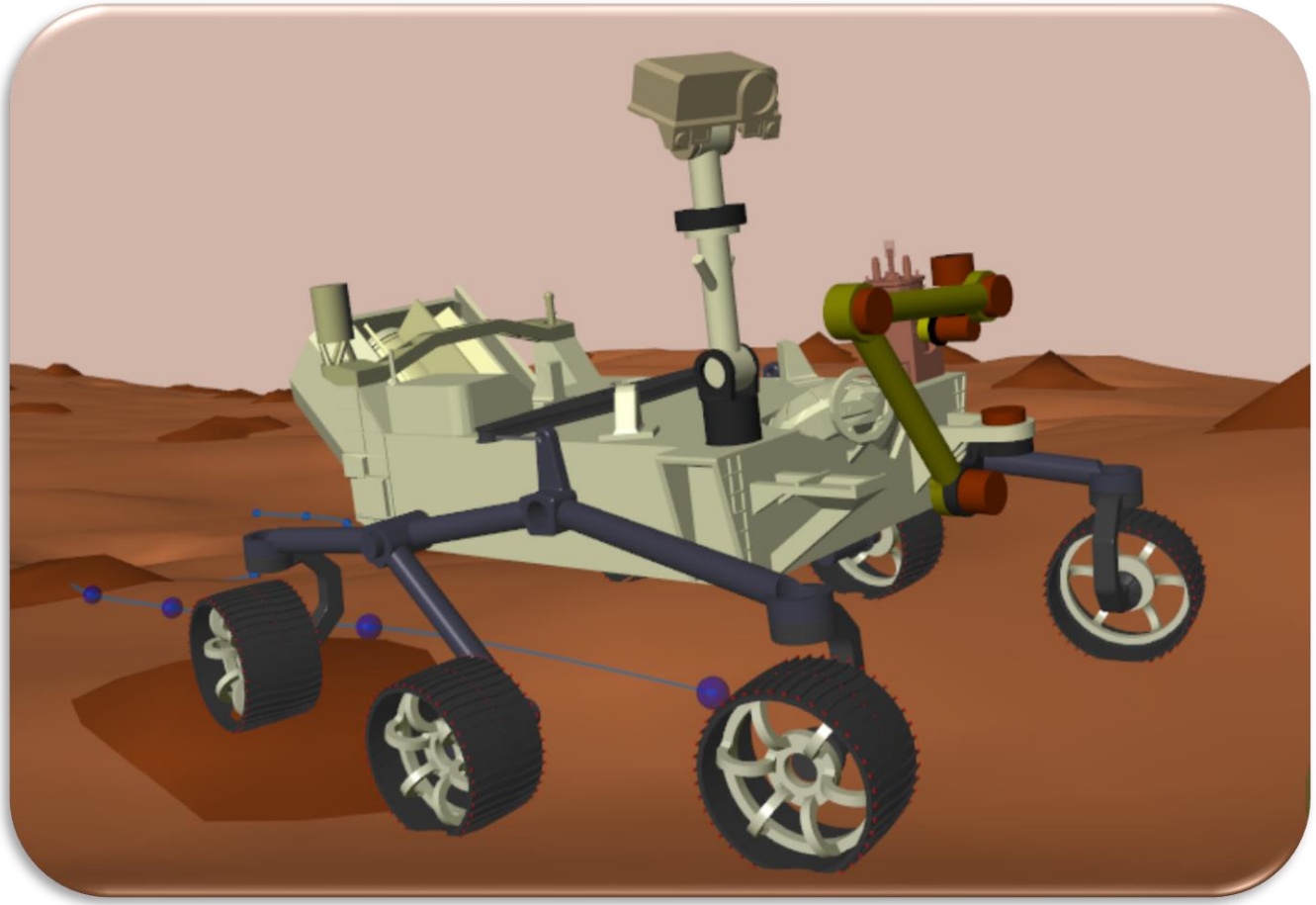


Where can I go?
Navigation

Multiple components are used to construct a physical model



Multiple components are used to construct a physical model



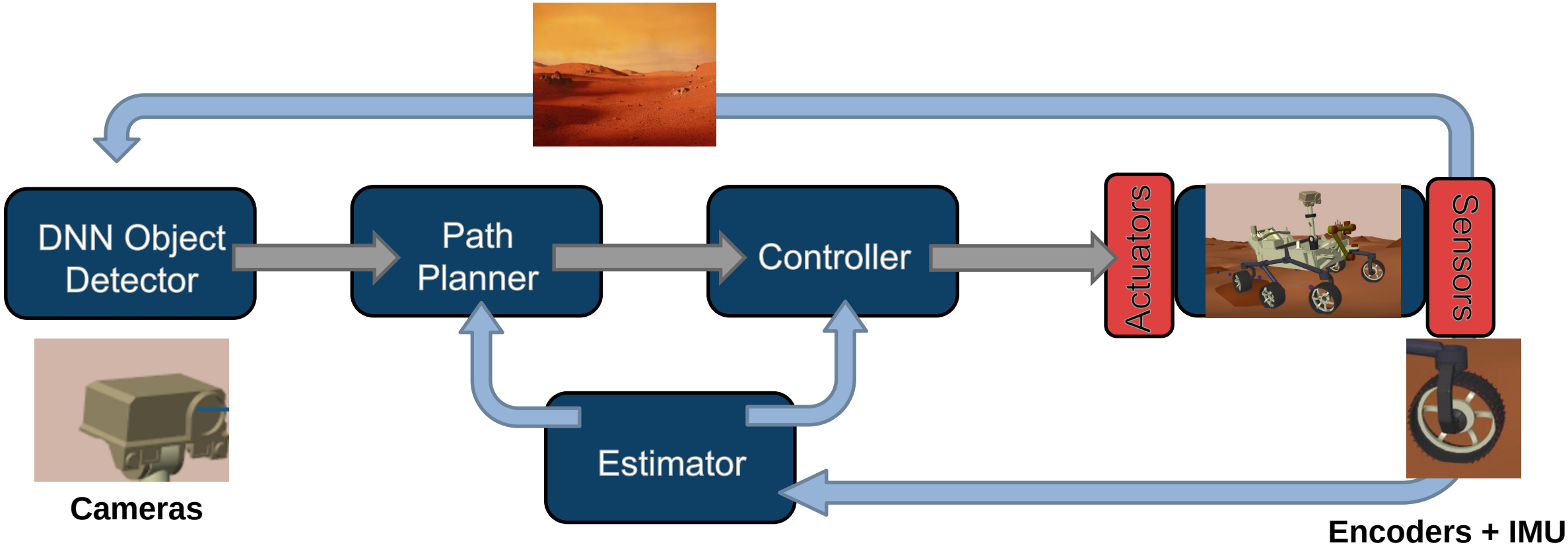
Chassis, Steering

Ground contact forces

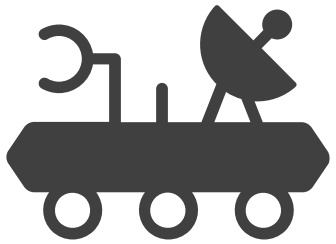
Max velocity (~0.1 mph)

Mapping Environment

Components of an autonomous rover



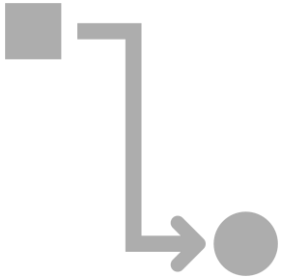
Our workshop today is split into multiple tasks



Who am I?
Rover Model

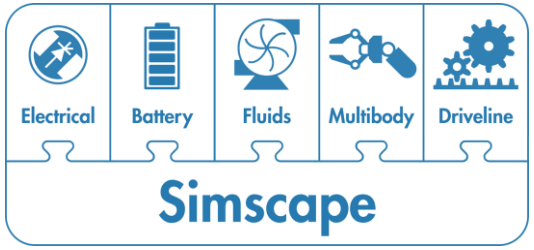
Where am I?
Calibration

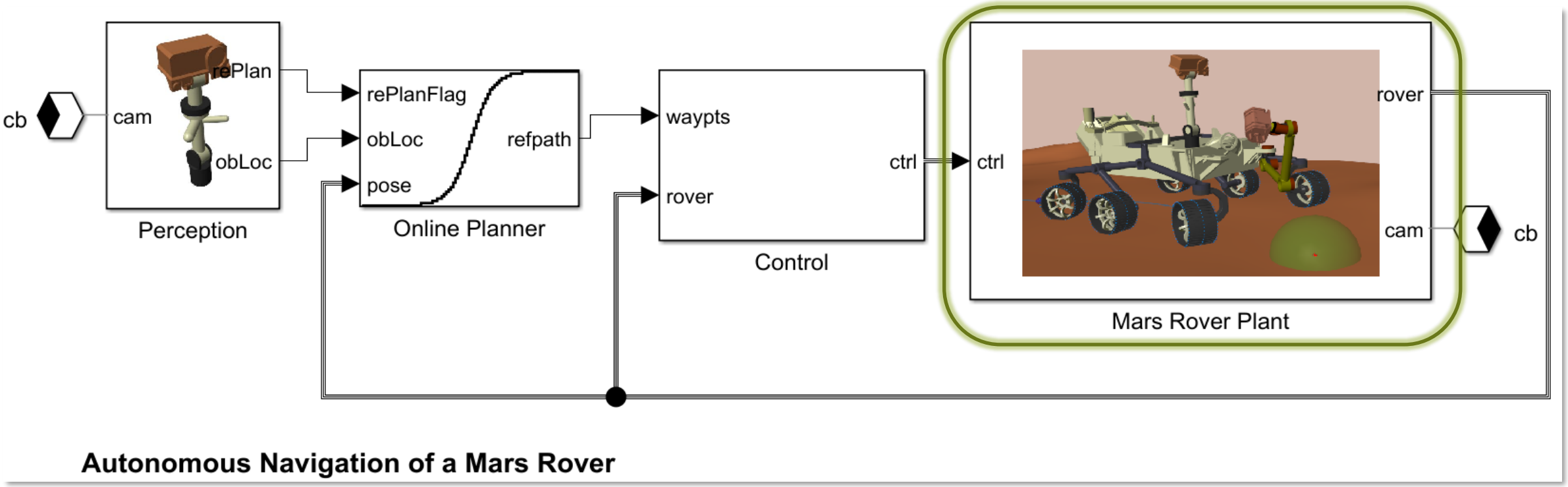
What do I see?
Object Detection



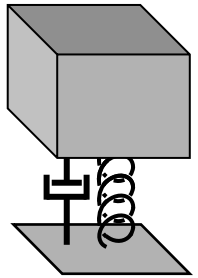
Where can I go?
Navigation

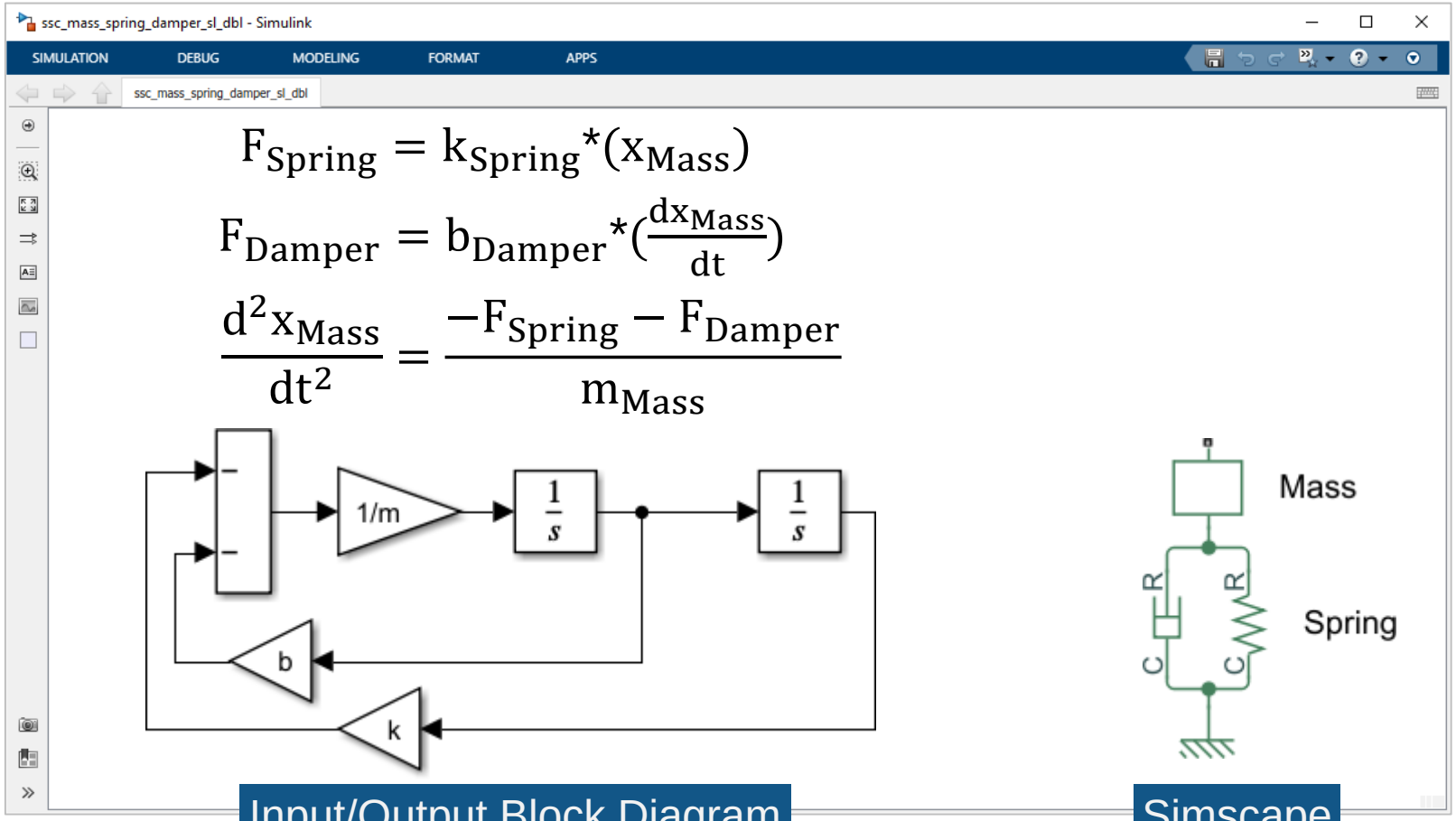
We used Simscape to build the multibody model of our Mars rover





Simscape makes modeling physical systems easy

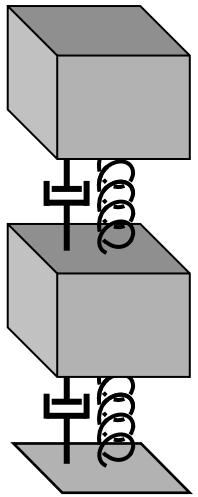


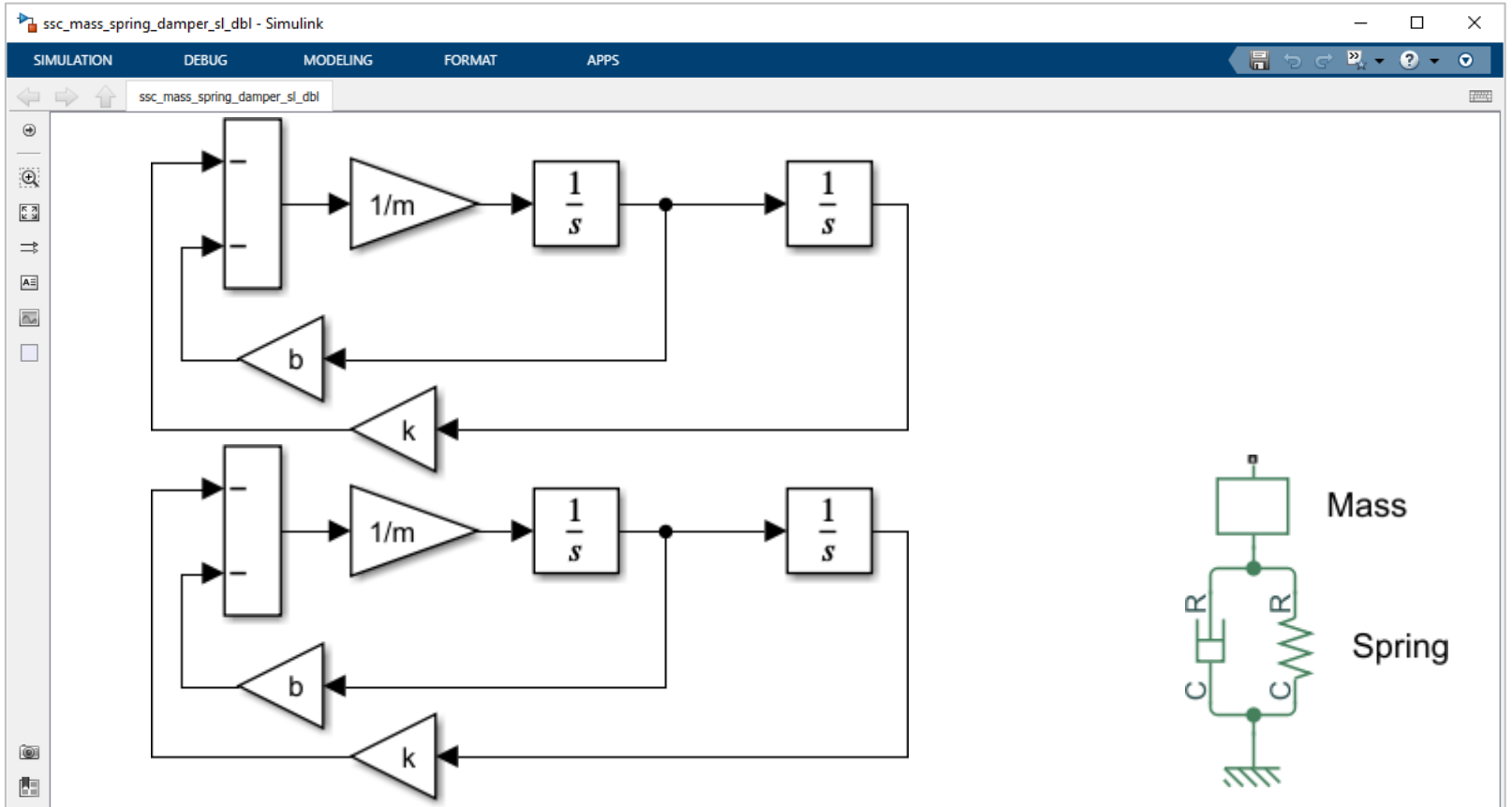


Input/Output Block Diagram

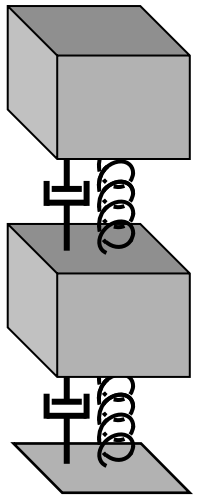
Simscape

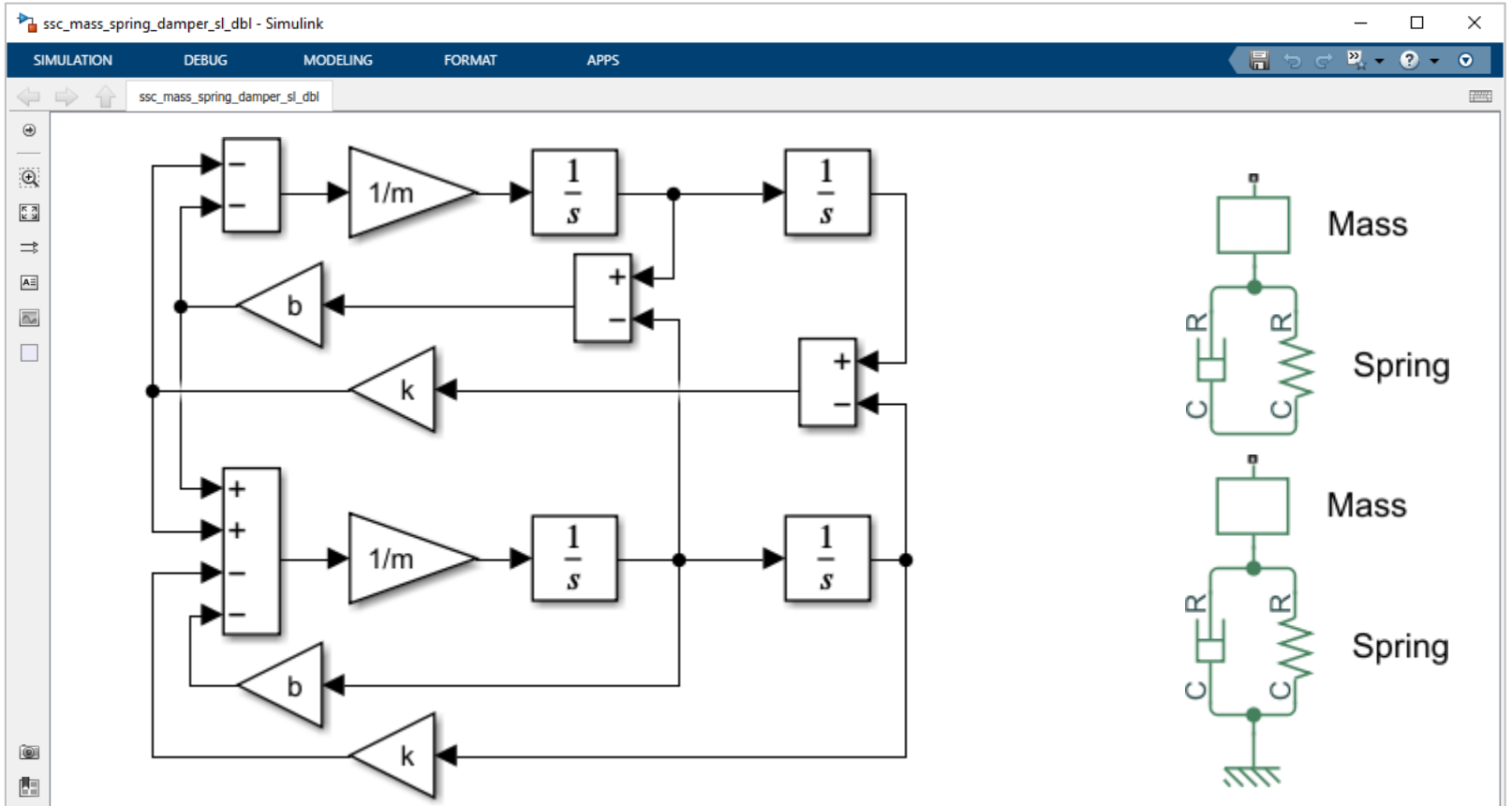
Simscape makes modeling physical systems easy





Simscape makes modeling physical systems easy

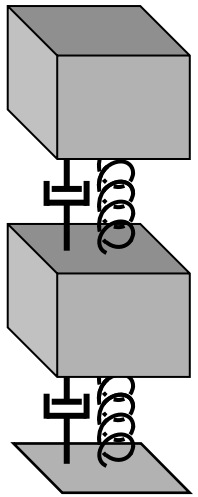


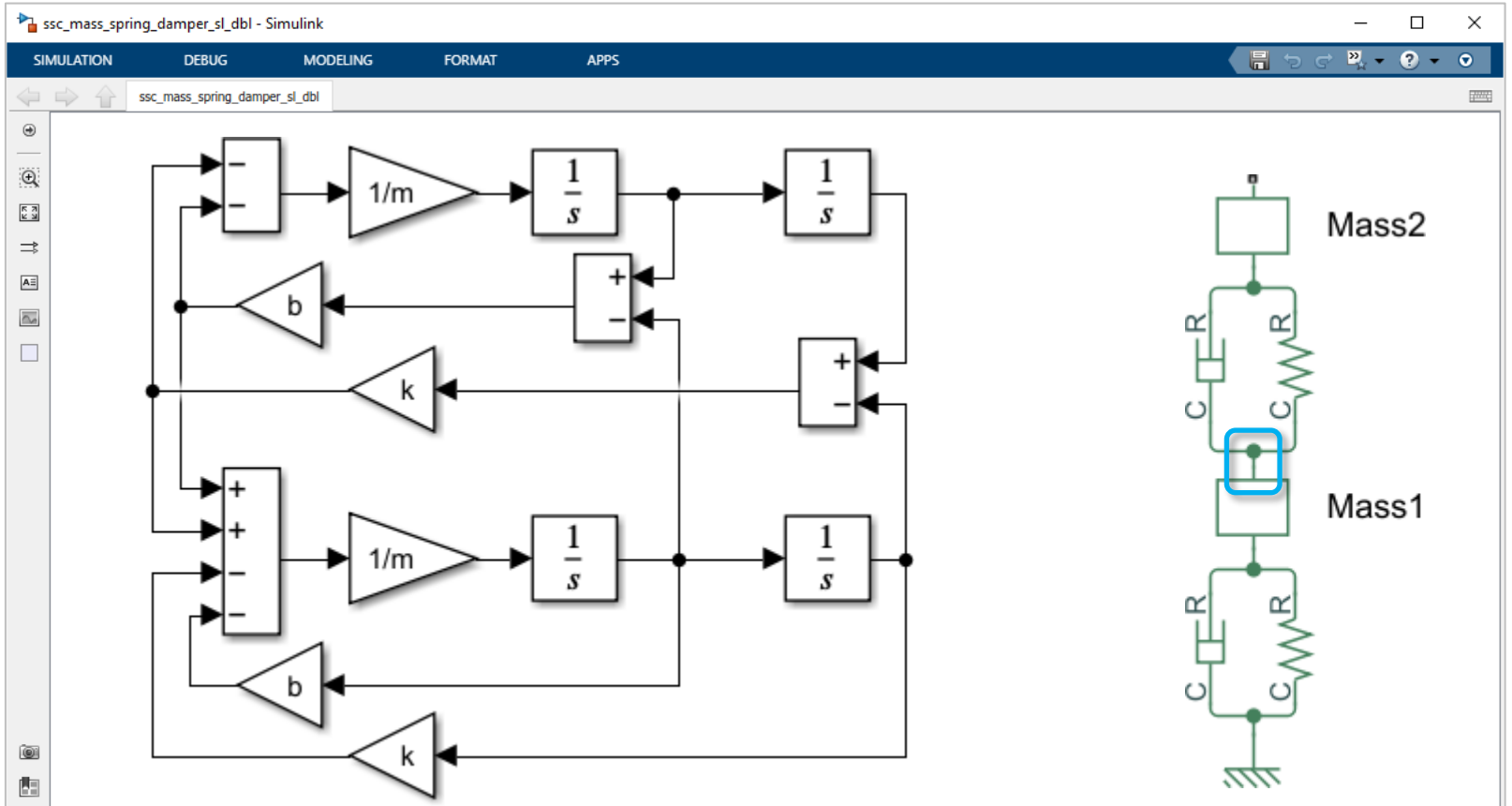


Input/Output Block Diagram

Simscape

Simscape makes modeling physical systems easy



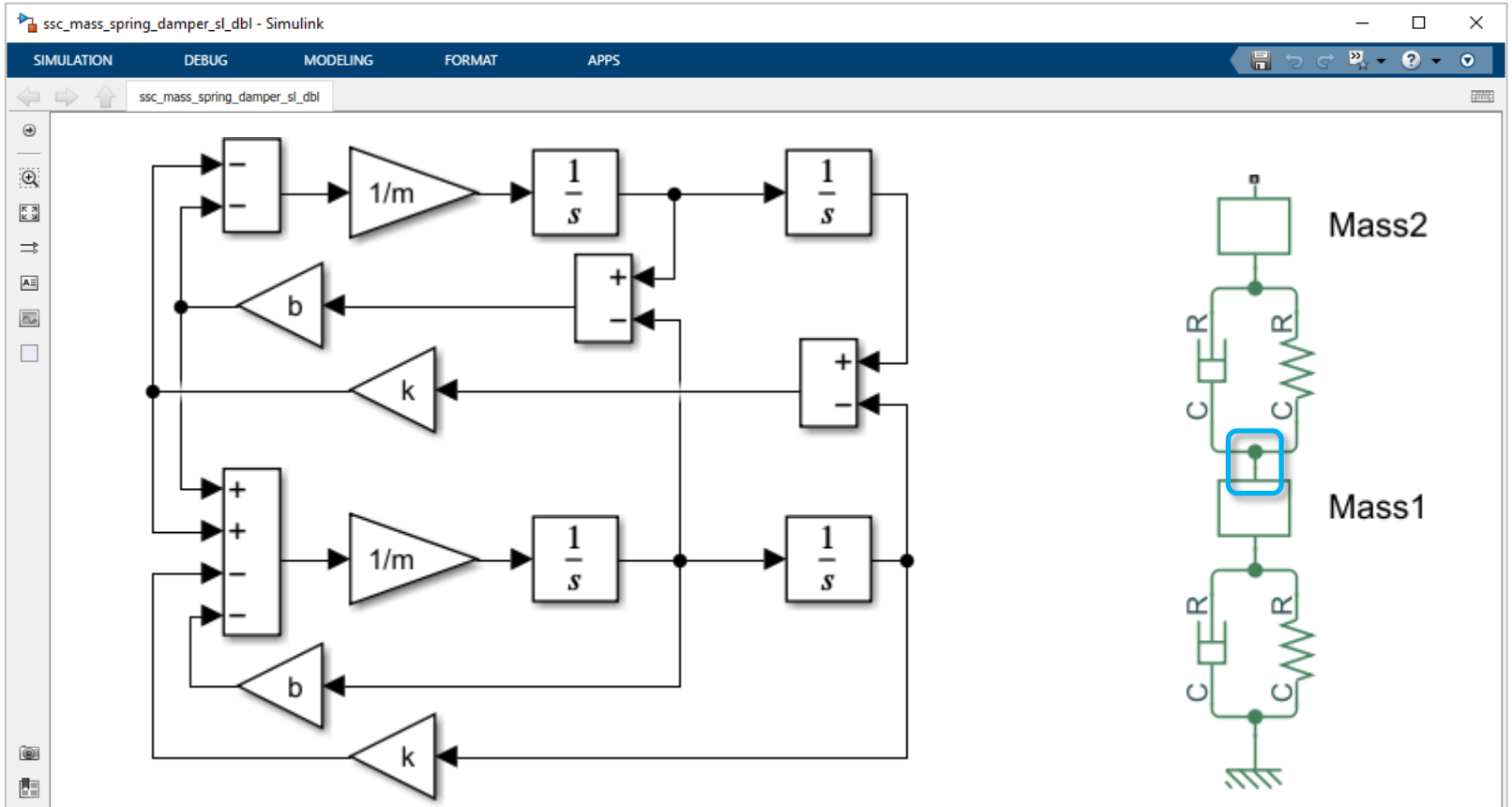


Input/Output Block Diagram

Simscape

Simscape makes modeling physical systems easy

- Simply connect the components you need
- The more complex the system, the more value you get from Simscape
- Resulting model is intuitive, easy to modify, and easy for others to understand

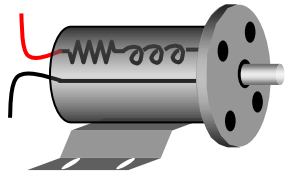


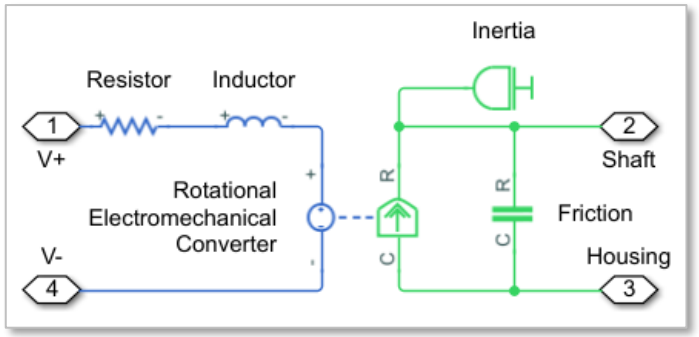
Input/Output Block Diagram

Simscape

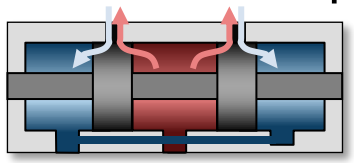
Modeling method applies to many domains

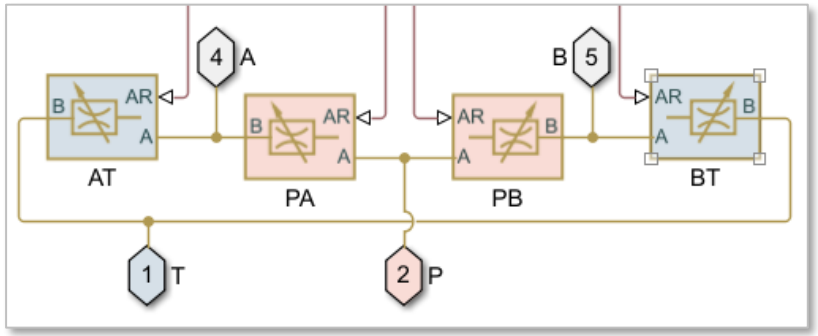
Electromechanical Systems
such as motors



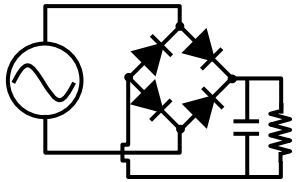


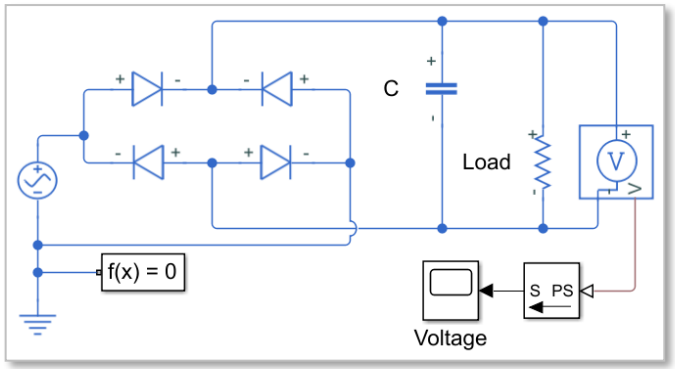
Fluid Systems
such as valves and pumps



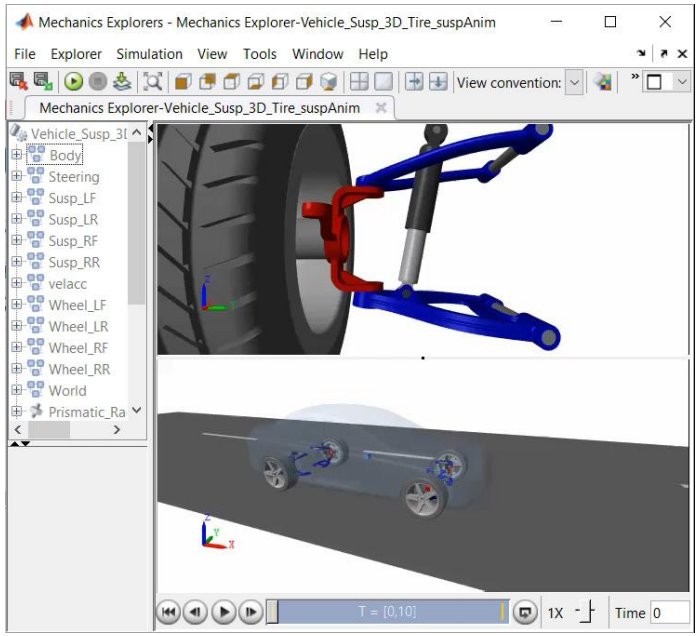


Electronic Systems
such as power converters

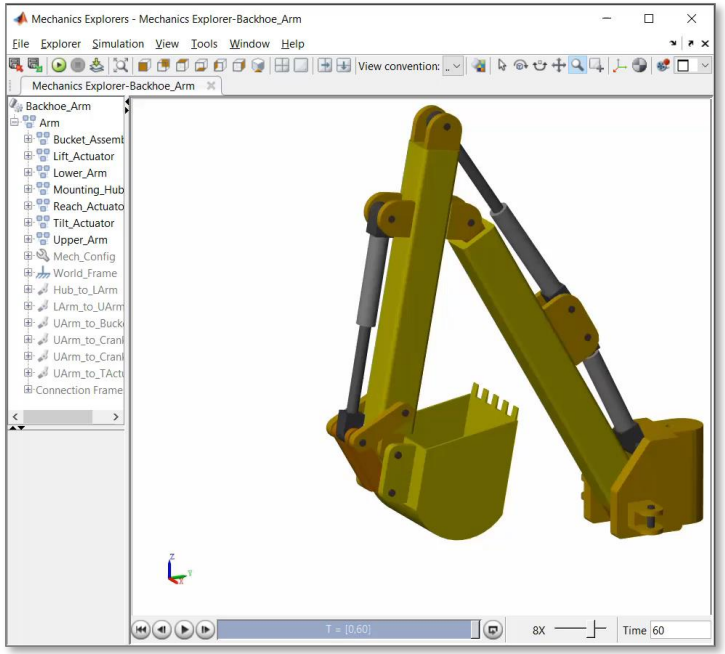




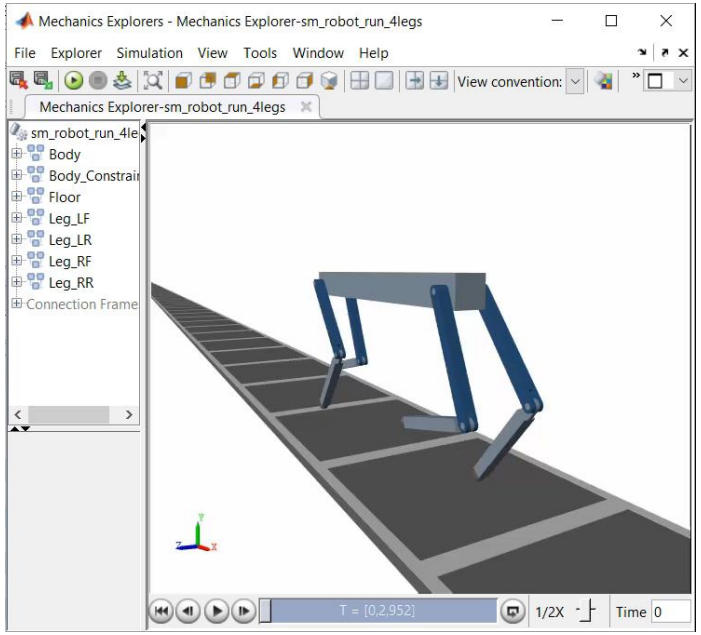
Modeling method scales to complex systems



Automotive Systems
such as vehicle dynamics
and powertrain

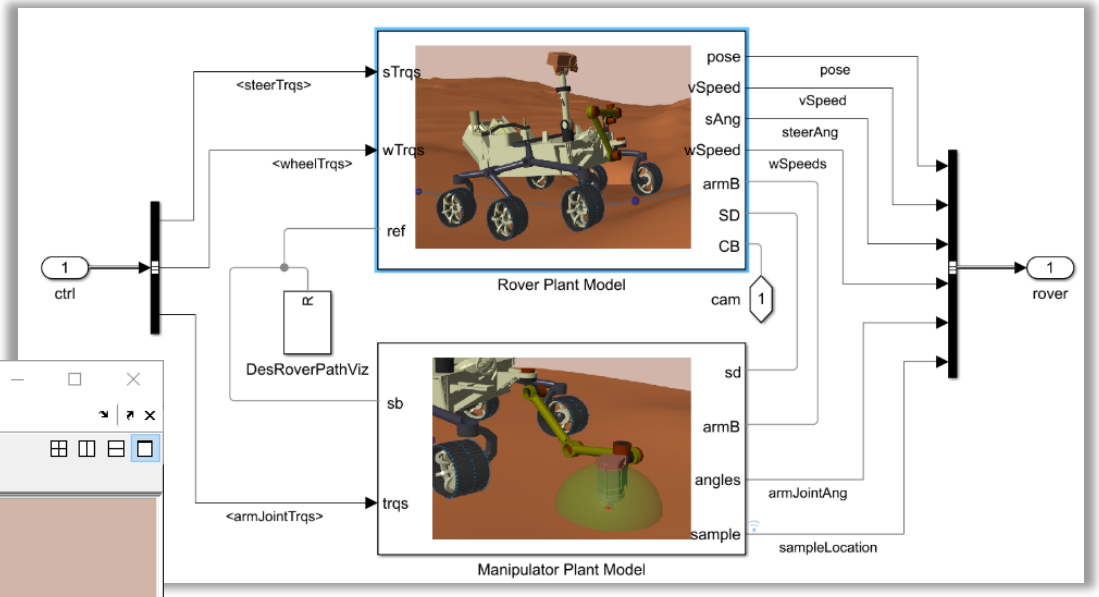


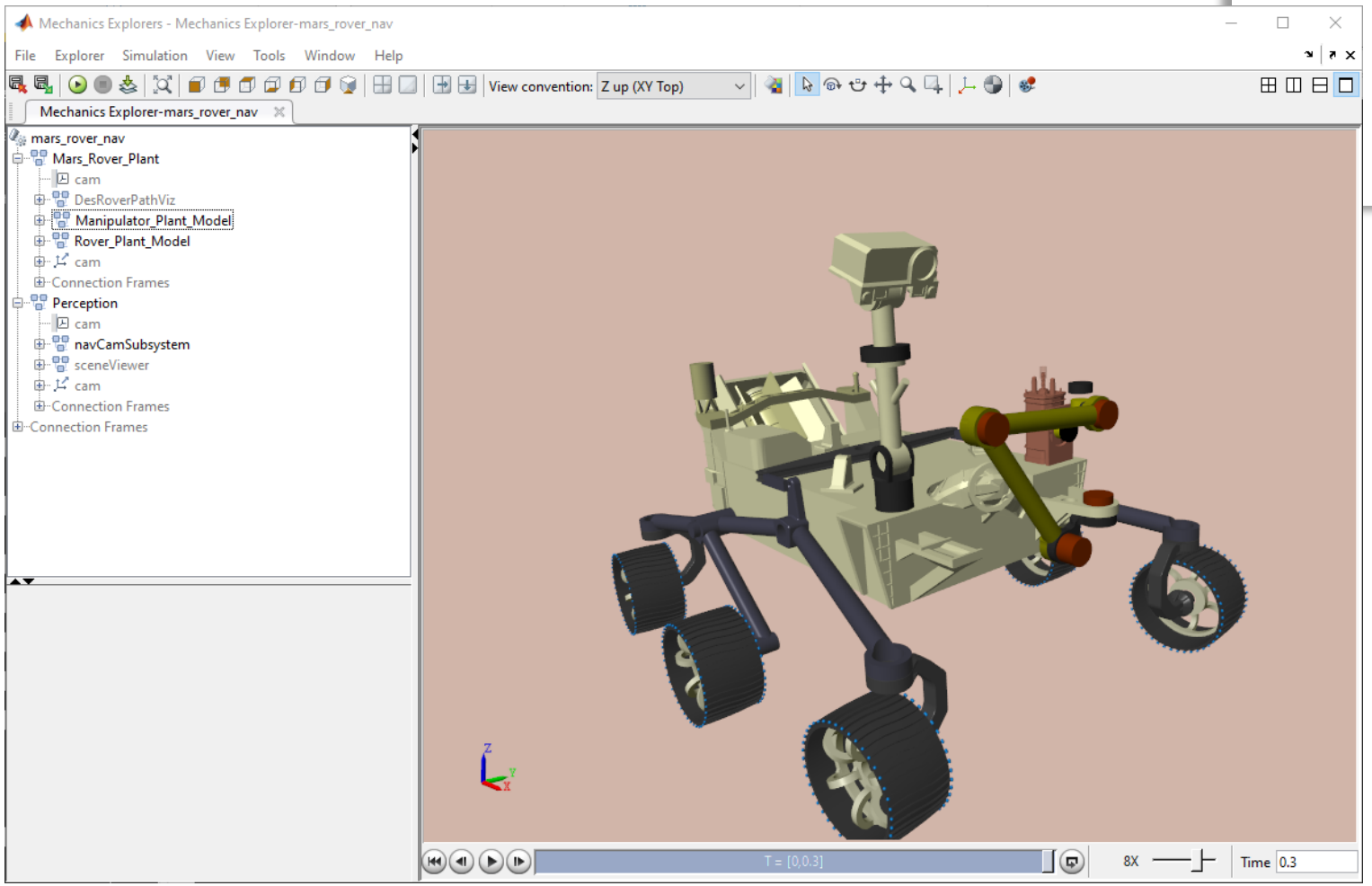
Industrial Equipment
such as backhoes



Robotic Systems
such as mobile robotics
and manipulators

Modeling the Rover





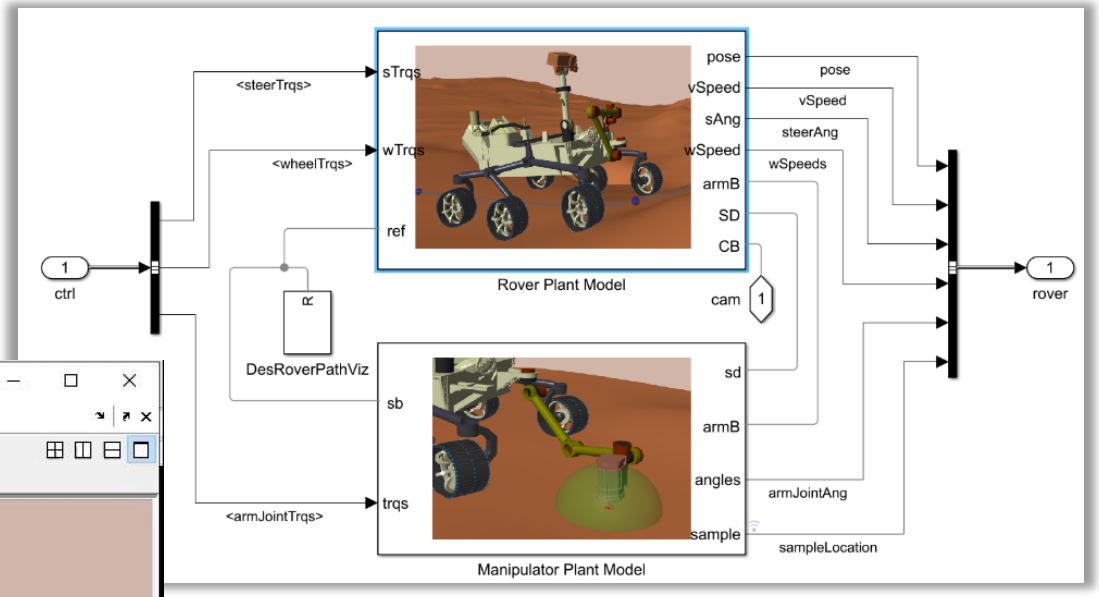
Chassis

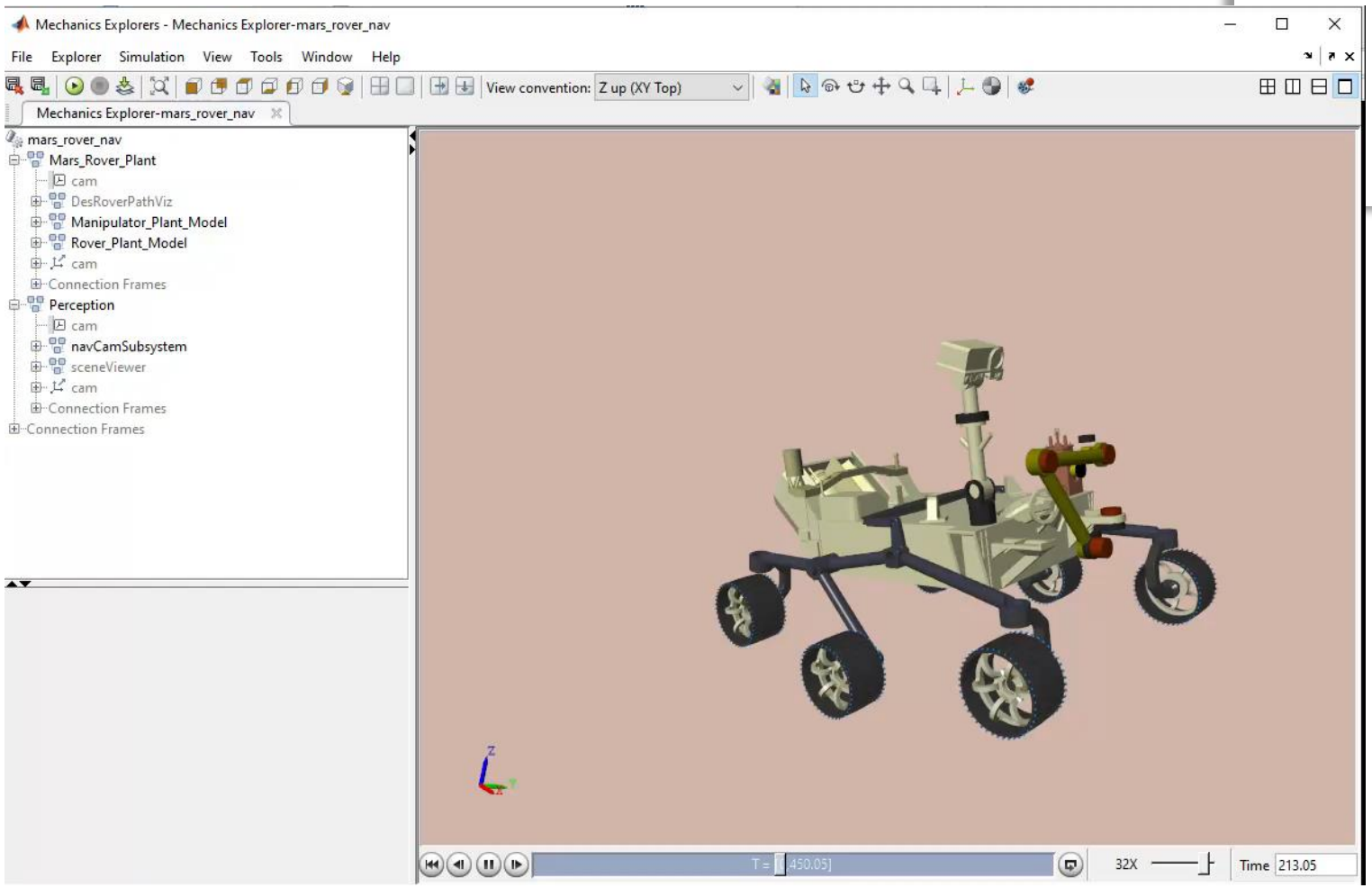
Rocker Bogie Suspension

Wheels

Arm

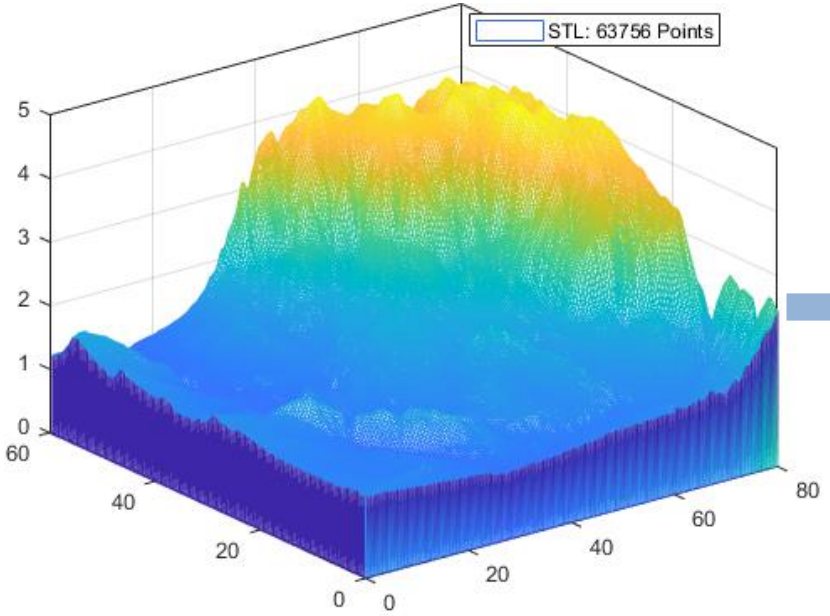
Modeling the Rover



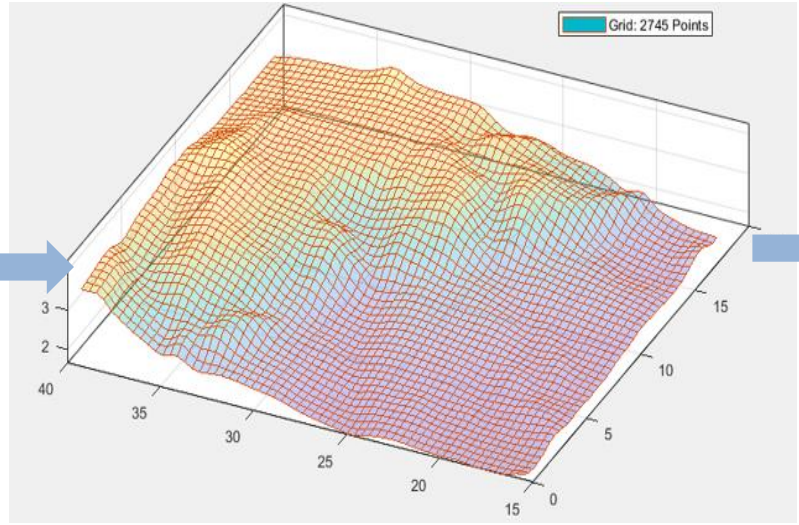


- Chassis
- Rocker Bogie Suspension
- Wheels
- Arm
- Motion Analysis

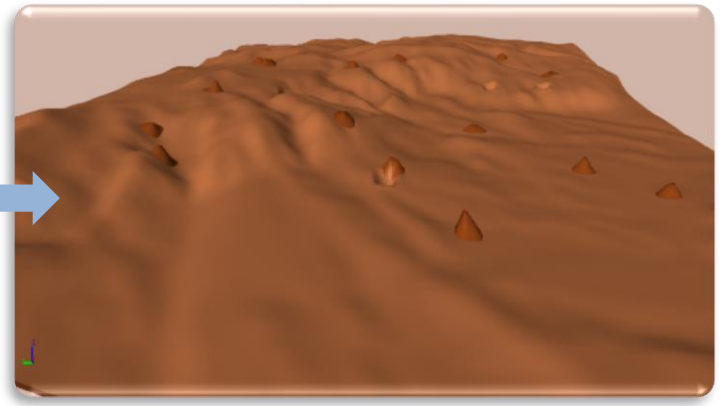
Virtual Mars Terrain



Terrain point cloud constructed using camera + satellite images

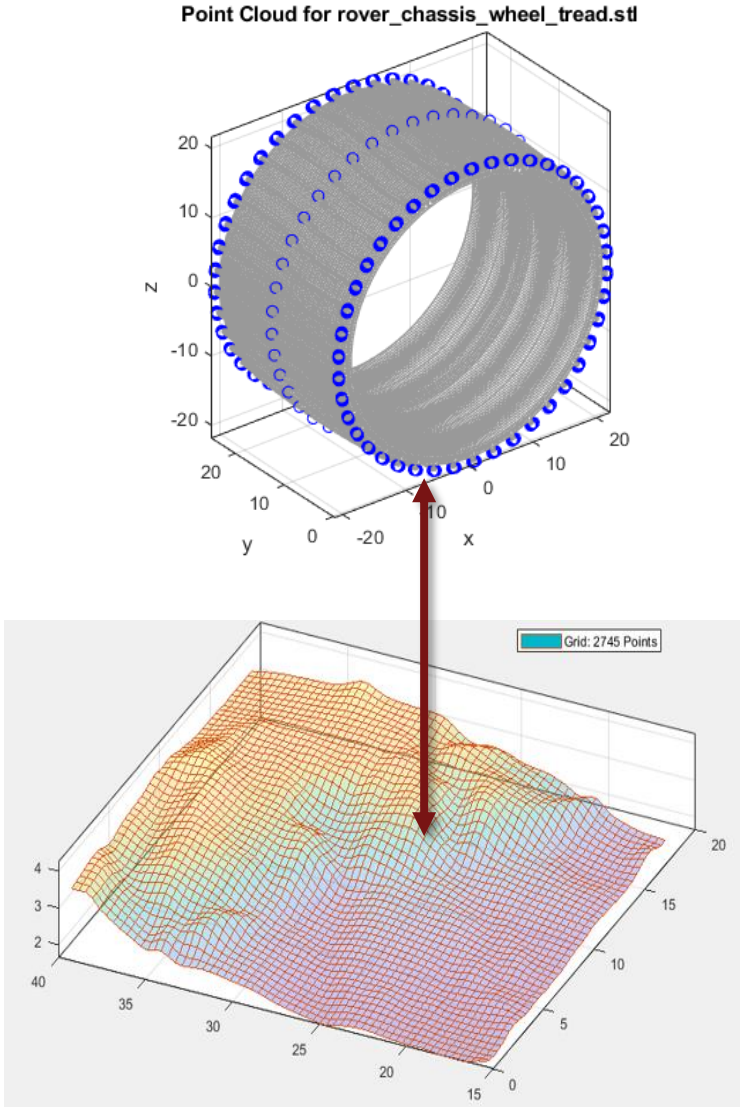


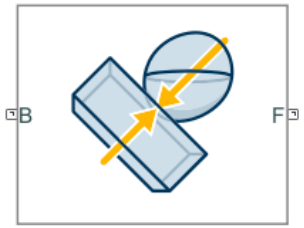
MATLAB to clean up data + create grid surface



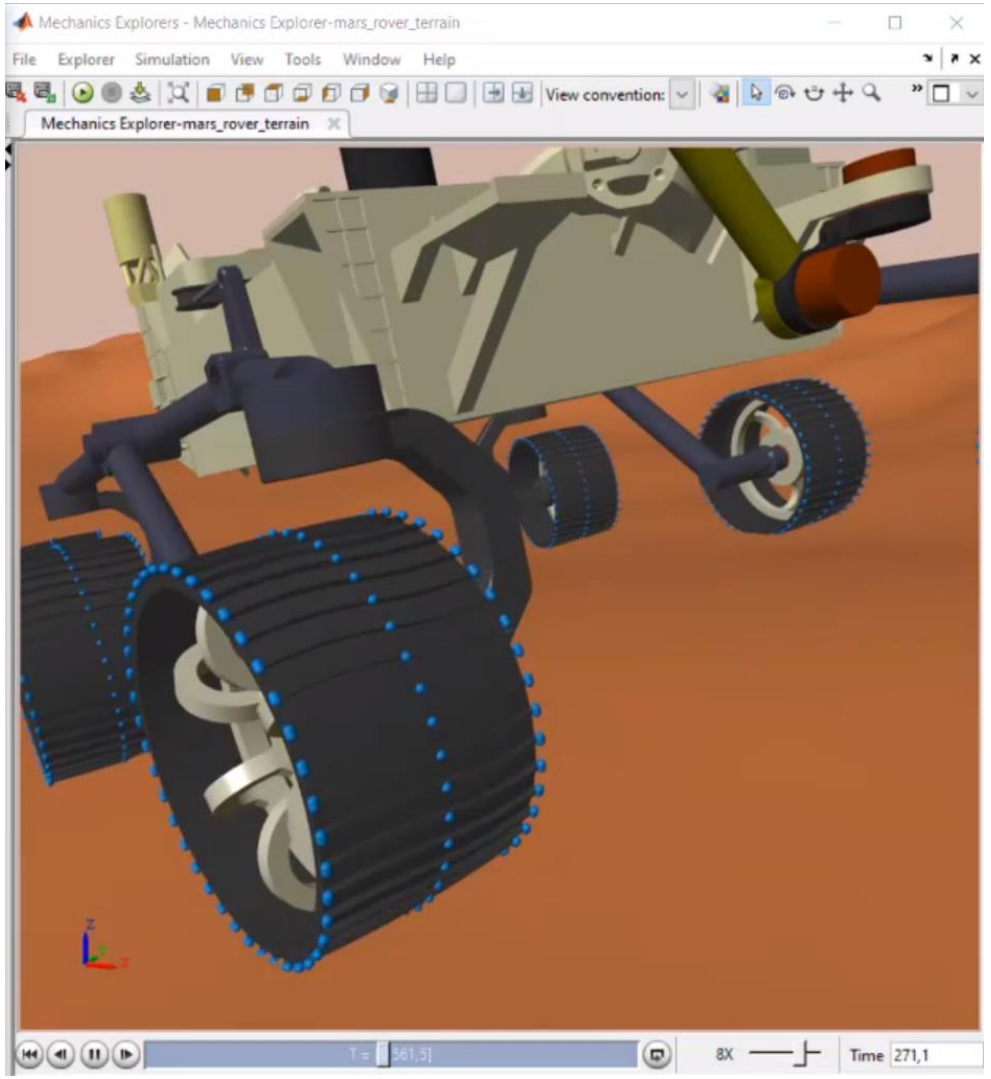
Simscape for terrain visualization + physics simulation

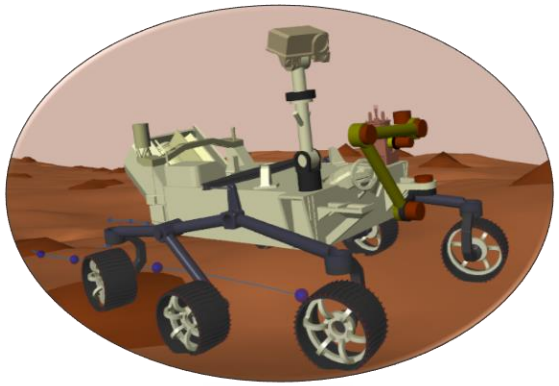
Modeling Ground forces





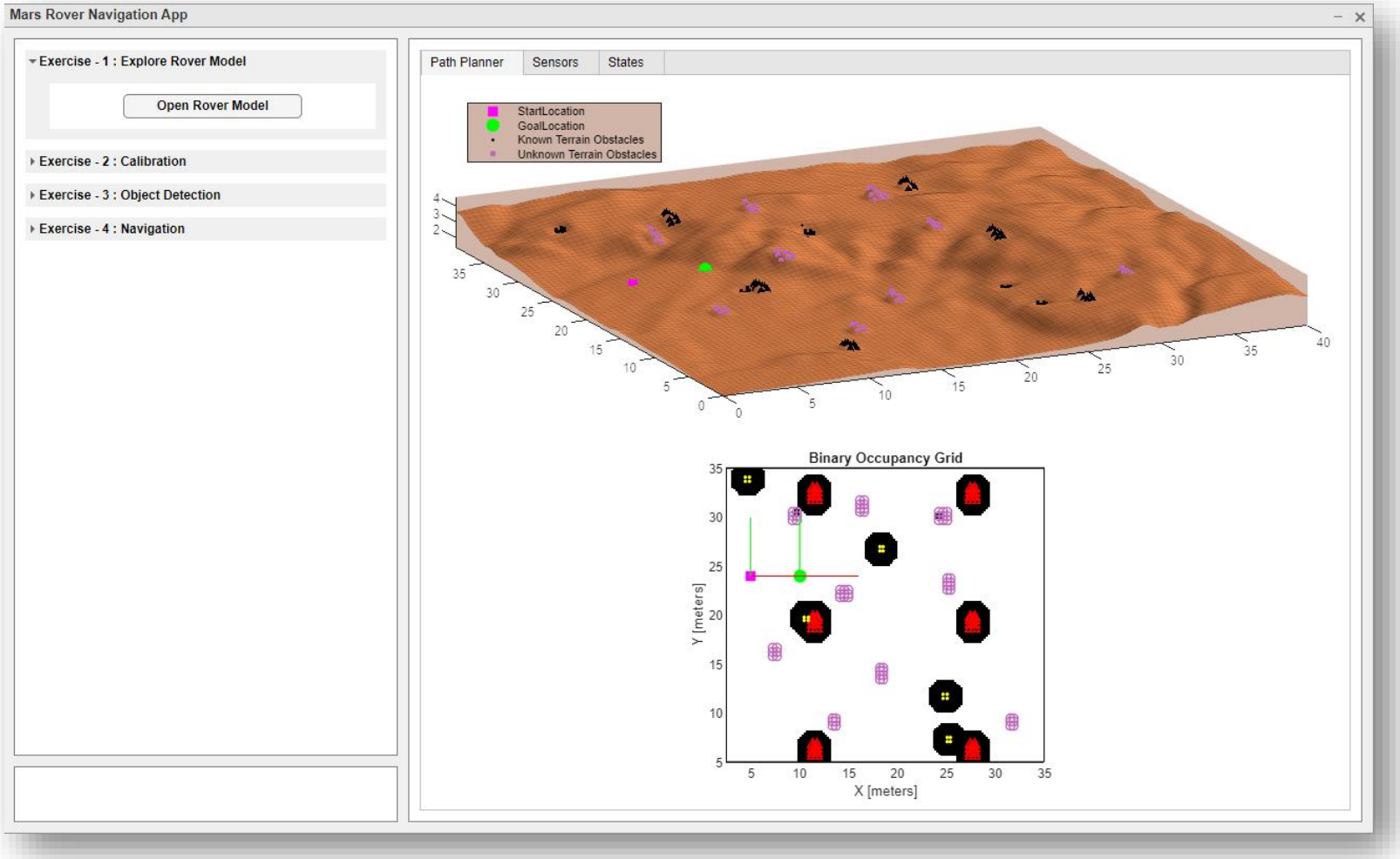
Spatial contact force block





Exercise 1: Explore Rover Model

Let’s dive into our hands-on exercise



Mars Rover Navigation App should now be open

```
>> mars_rover_startup
```

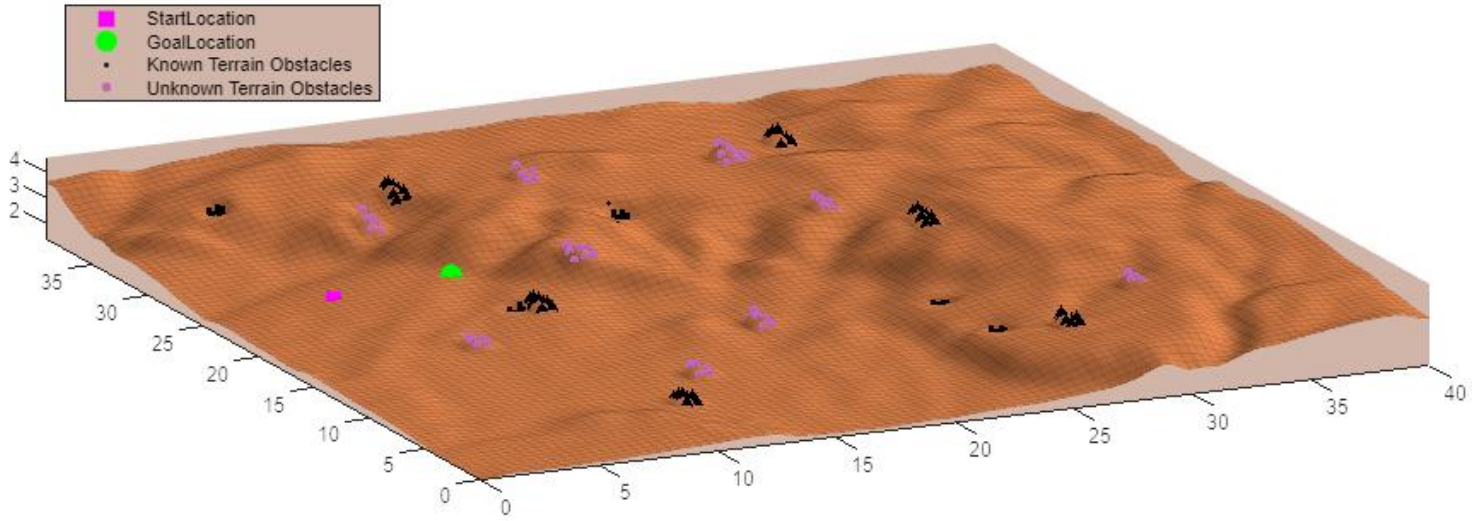
You will be given time at the end to work on the exercise

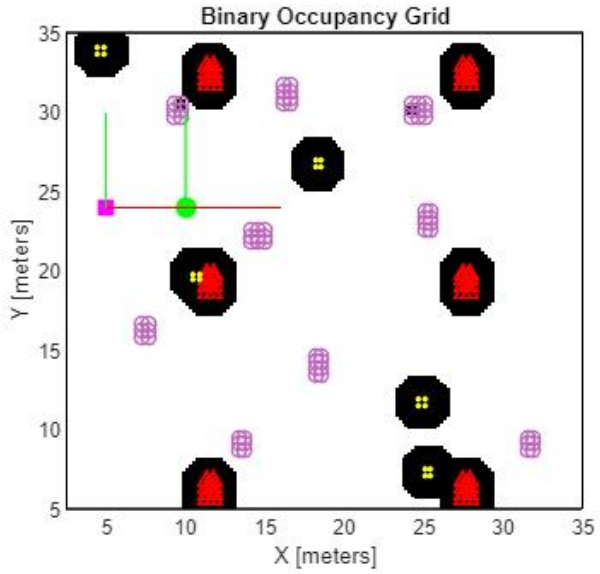
Mars Rover Navigation App

- Exercise - 1 : Explore Rover Model
  - Open Rover Model
- Exercise - 2 : Calibration
- Exercise - 3 : Object Detection
- Exercise - 4 : Navigation

Explore Rover Model

Path Planner | Sensors | States





Opens mars_rover_explore.slx

mars_rover_explore - Simulink

SIMULATION DEBUG MODELING FORMAT APPS

mars_rover_explore

mars_rover_explore

Control Dashboard

desCtrSig

rover

ctrl

Control

ctrl

wheels

rover

Mars Rover Plant

<--- CONNECT --->

terrain

Terrain

Outputs

Rover Drive Energy

Ready

138%

Notes - mars_rover_explore_notes.mldatx

Write Notes

Exercise 1 : Explore Rover Model

(Model window resizing help)

Modeling the physical components of the Mars rover is an important first step in developing the simulation of its autonomous navigation. Using Simscape, we can model, connect, and simulate various physical components of the rover.

Try the following steps to setup the model of our Mars rover and test it on different terrains via simulation. (*Click through the hyperlinks to execute the necessary commands*)

Model the ground contact forces between rover & terrain:

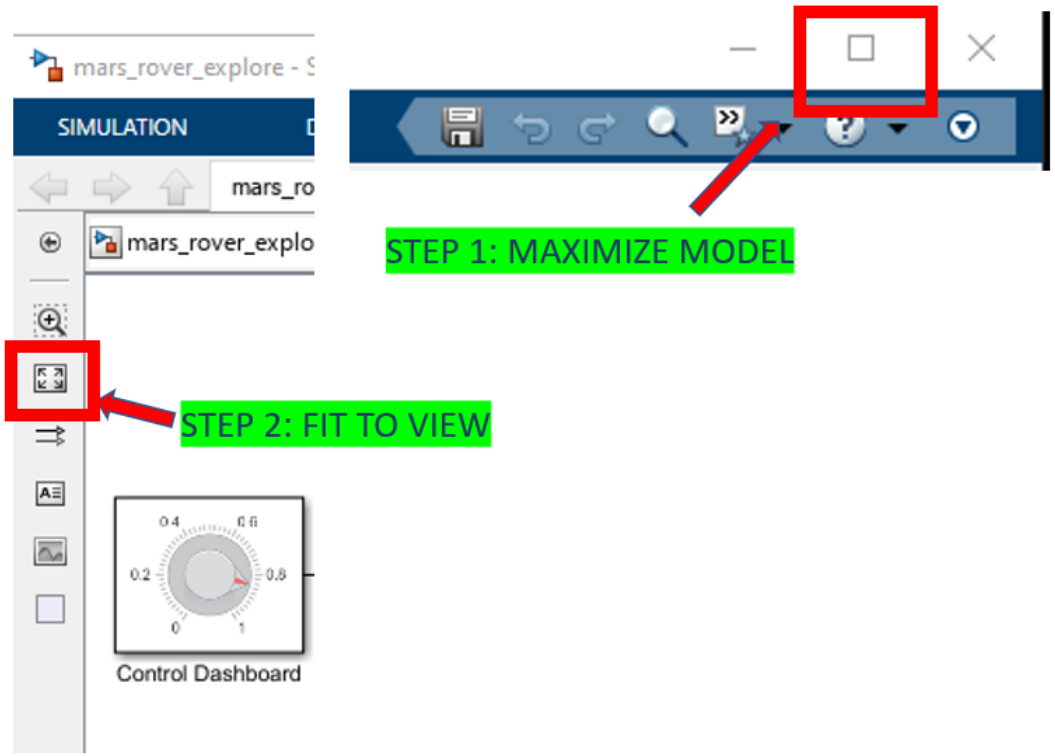
1. Uncomment the ***Terrain*** block & attach it to the ***Mars Rover Plant*** block by connecting ***wheels--->terrain*** ports.
2. Simulate Rover on a Flat surface. (Stop)
   - *Did the rover run ?*
   - *Observe the energy used by the rover to drive.*

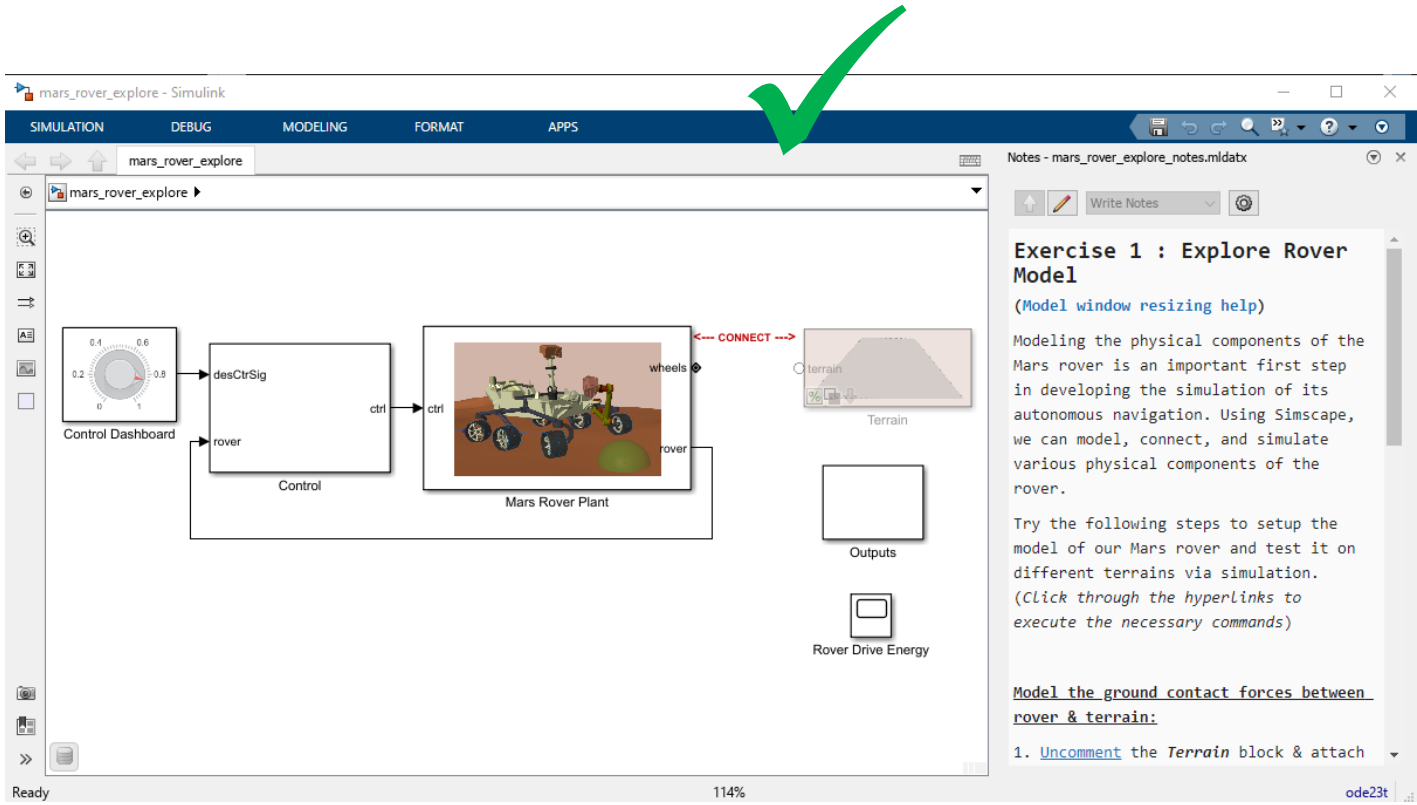
Model the virtual Mars terrain:

ode23t

If your model is too small...

- Ensure your browser window is maximized
- Ensure that your model is maximized or resized.
- Press the space bar or fit to view button.





Exercise 1 : Model ground contact

mars_rover_explore * - Simulink

SIMULATION DEBUG MODELING FORMAT APPS

mars_rover_explore

mars_rover_explore

Control Dashboard

desCtrSig

rover

ctrl

Control

ctrl

wheels

rover

<--- CONNECT --->

terrain

Terrain

Notes - mars_rover_explore_notes.mldatx

Write Notes

Model the ground contact forces between rover & terrain:

1. Uncomment the ***Terrain*** block & attach it to the ***Mars Rover Plant*** block by connecting ***wheels--->terrain*** ports.

2. Simulate Rover on a Flat surface. (Stop)

   - *Did the rover run ?*
   - *Observe the energy used by the rover to drive.*

Model the virtual Mars terrain:

3. Now using MATLAB, let us generate the necessary surface data required by Simscape to model the uneven terrain.

   - *Was a surface plot generated ?*

4. See which Simscape block uses this generated surface data to model our virtual terrain for simulation.

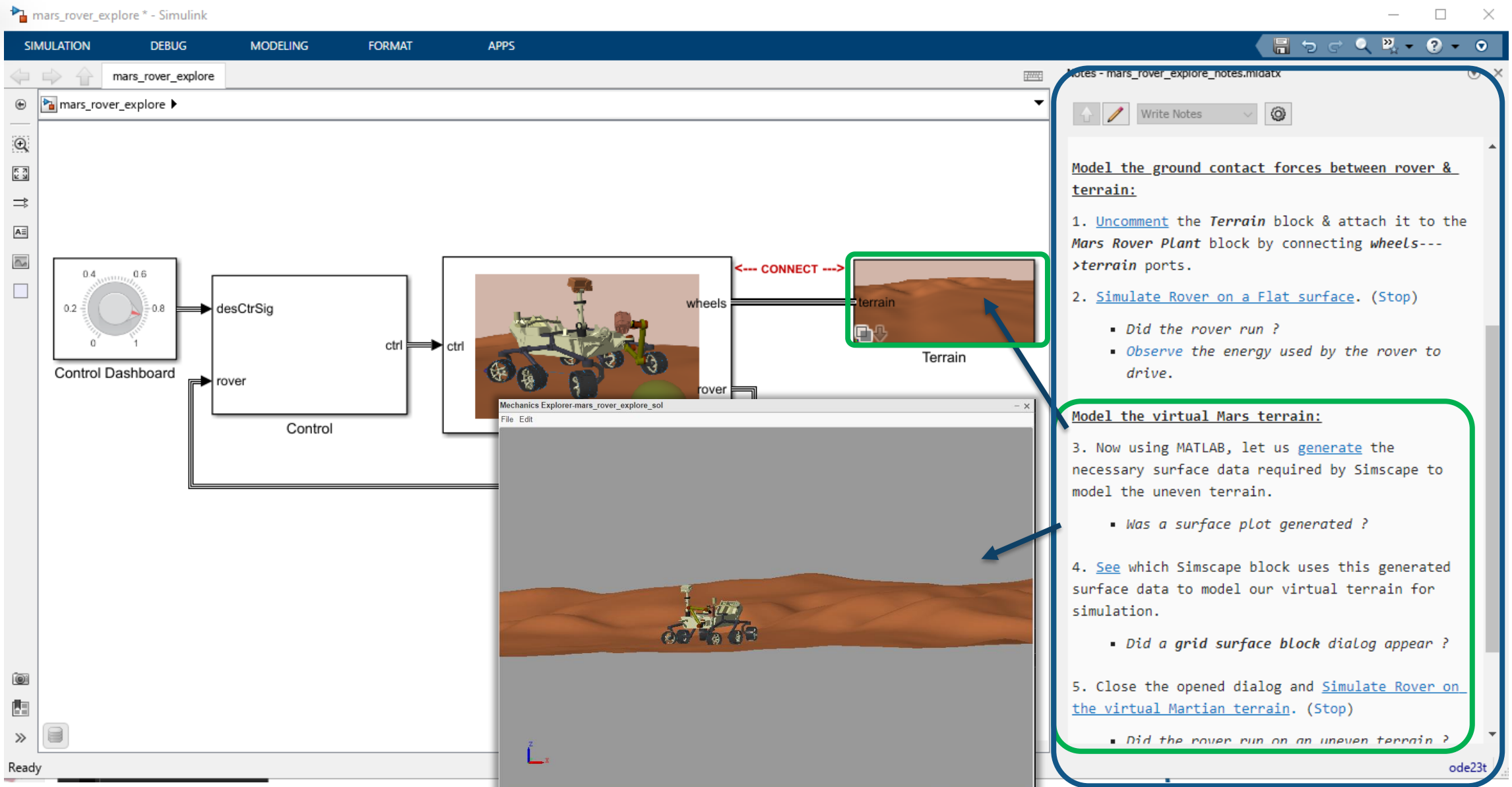
   - *Did a **grid surface block** dialog appear ?*

5. Close the opened dialog and Simulate Rover on the virtual Martian terrain. (Stop)

   - *Did the rover run on an uneven terrain ?*

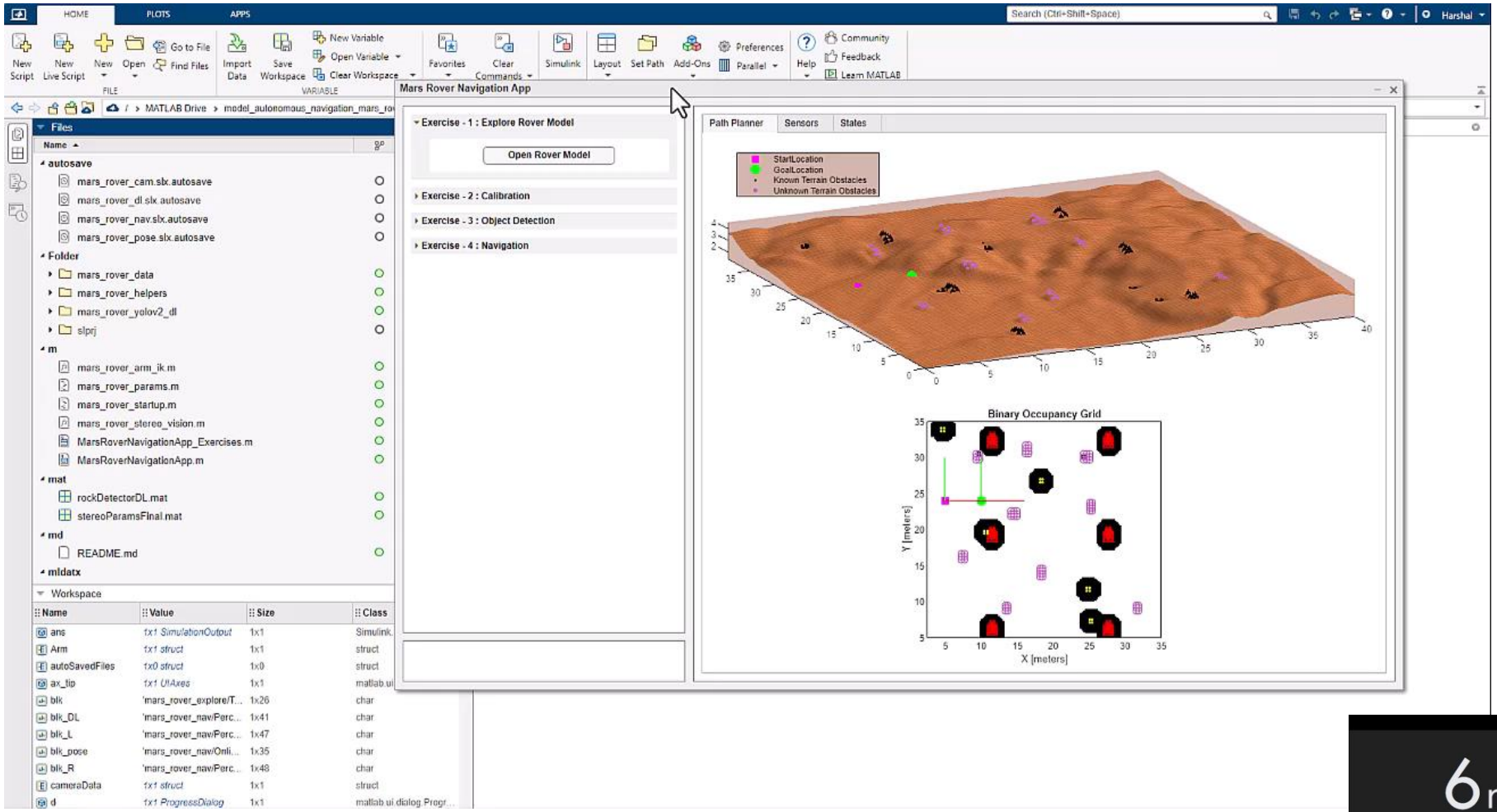
ode23t

Exercise 1 : Model Virtual Mars Terrain



Please take 6 minutes to complete exercise 1

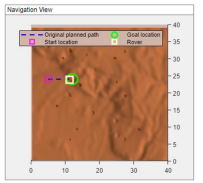
Need a hint? A video showing the solution will play here.

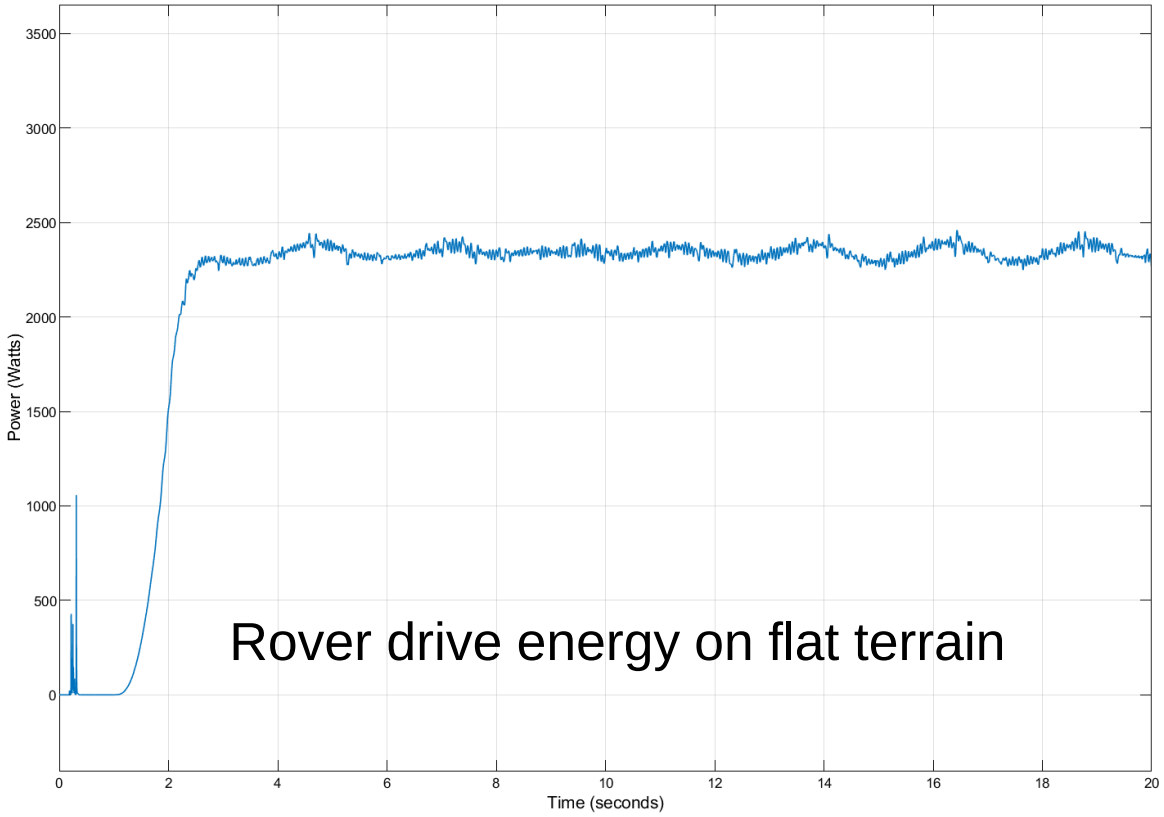


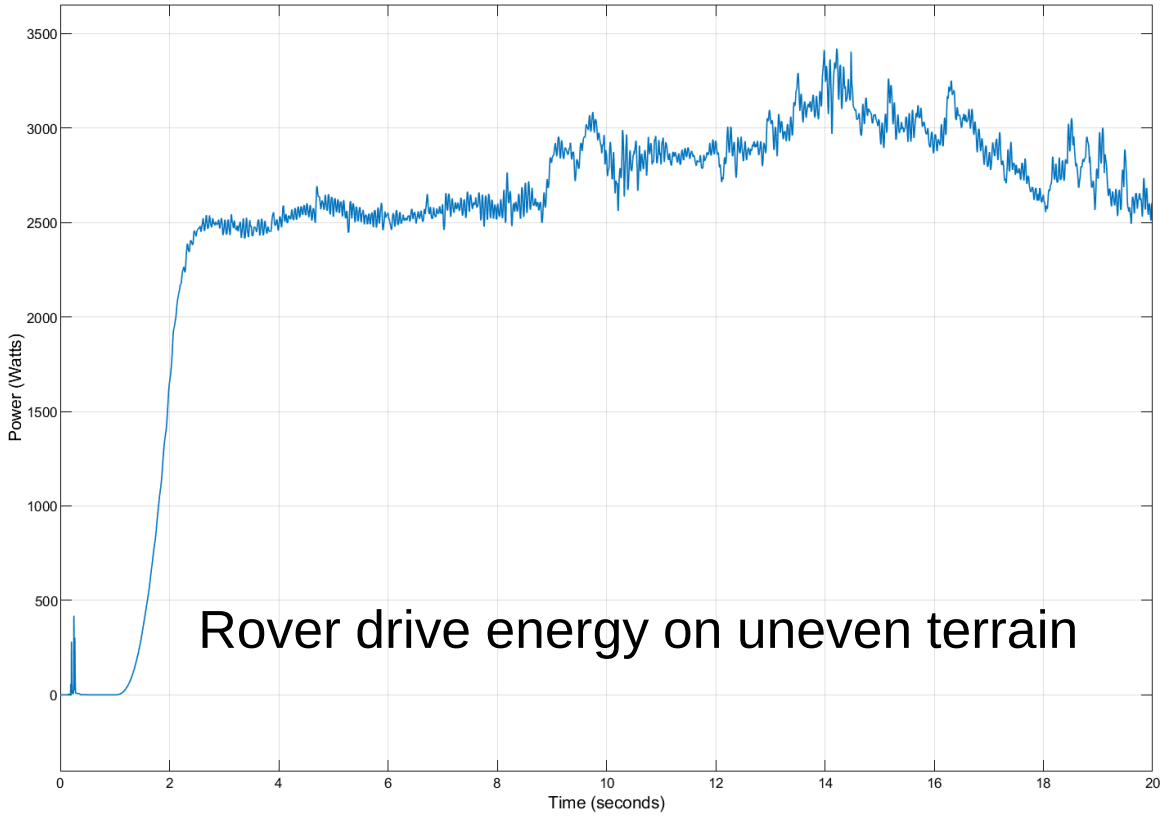
6m 00s

What did you observe?

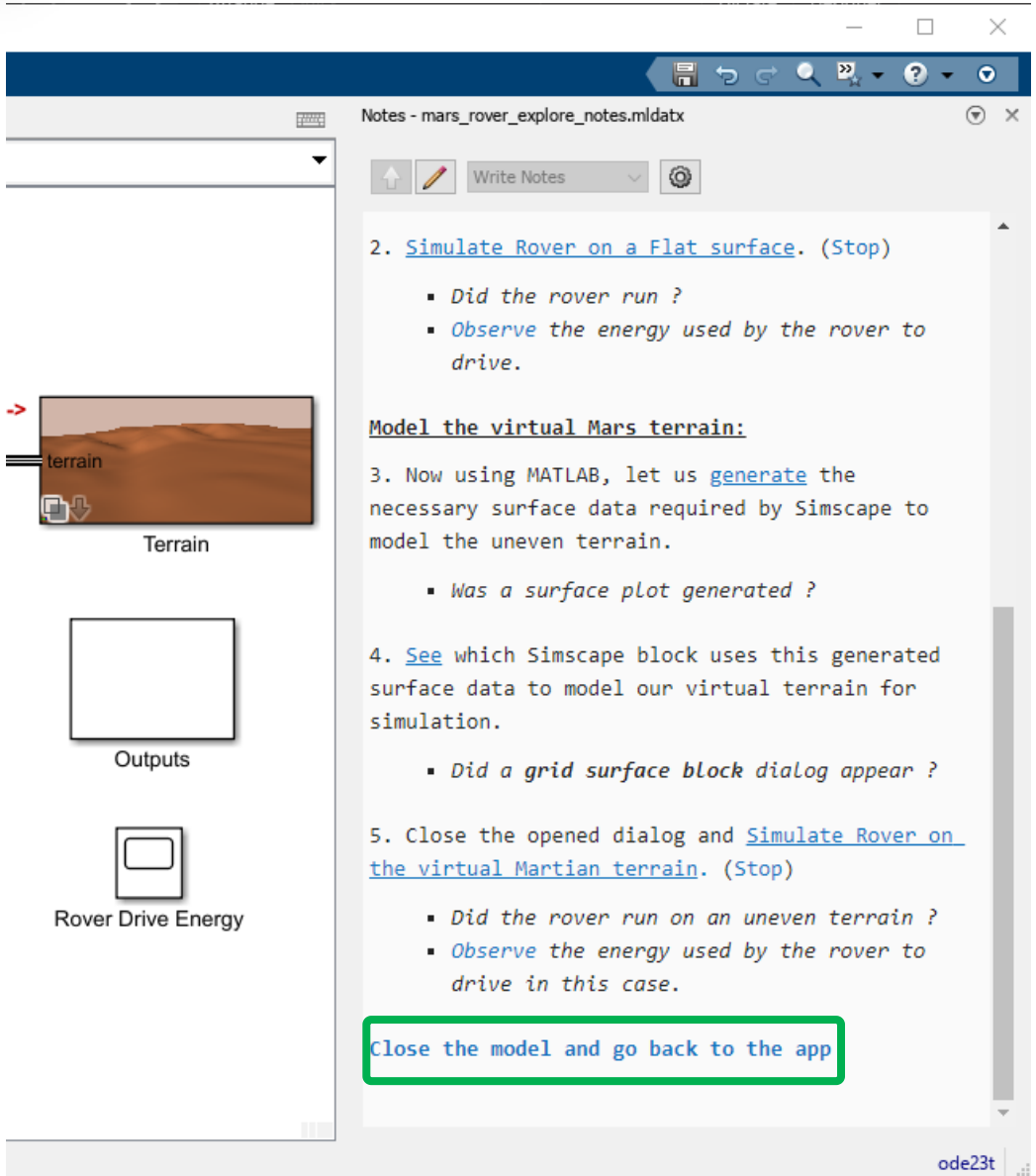
- Did the rover simulate on a flat surface ?
- Was the rover able to traverse the uneven terrain in the simulation ?
- Energy needed to drive the rover increases on slopy uneven terrain.



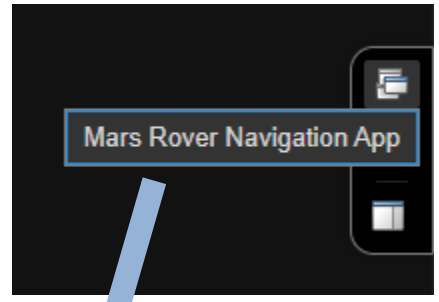


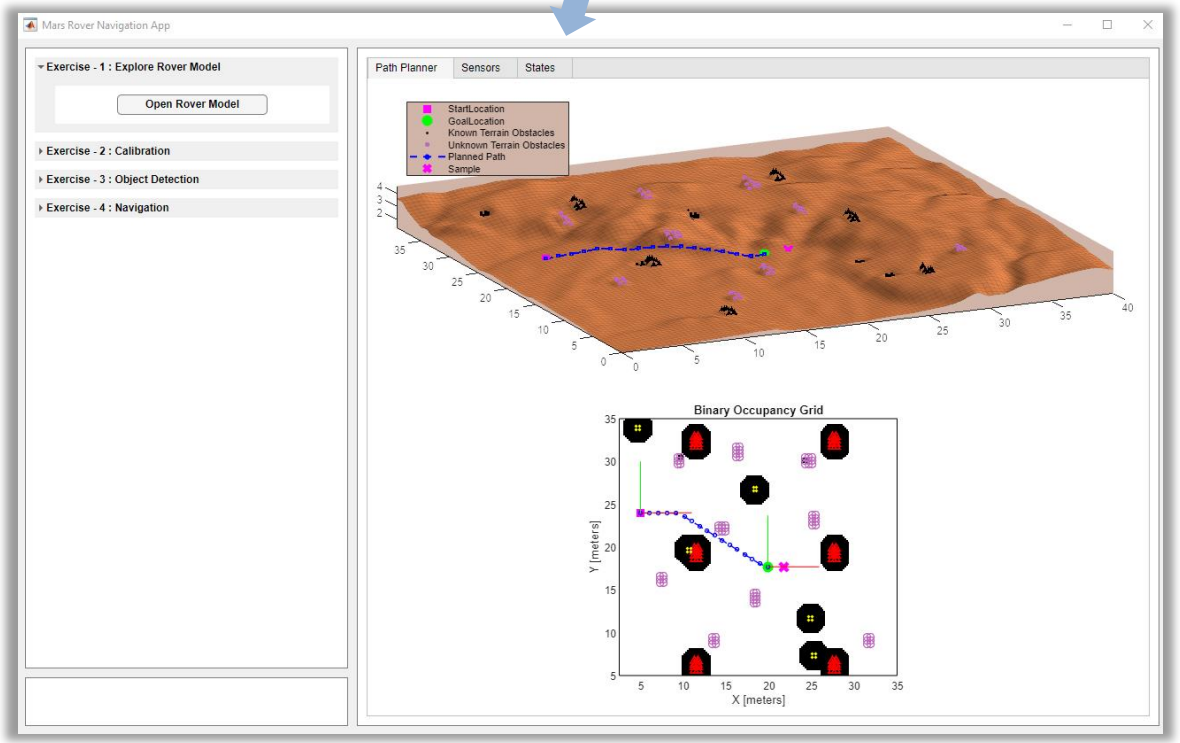


Exercise-1 clean up

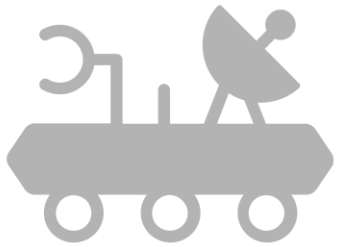


If your app has been minimized, then you can find it here at the lower right





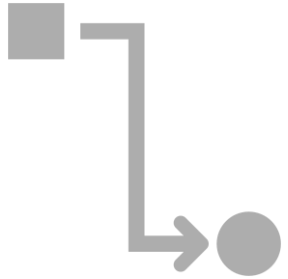
Our workshop today is split into multiple tasks



Who am I?
Rover Model

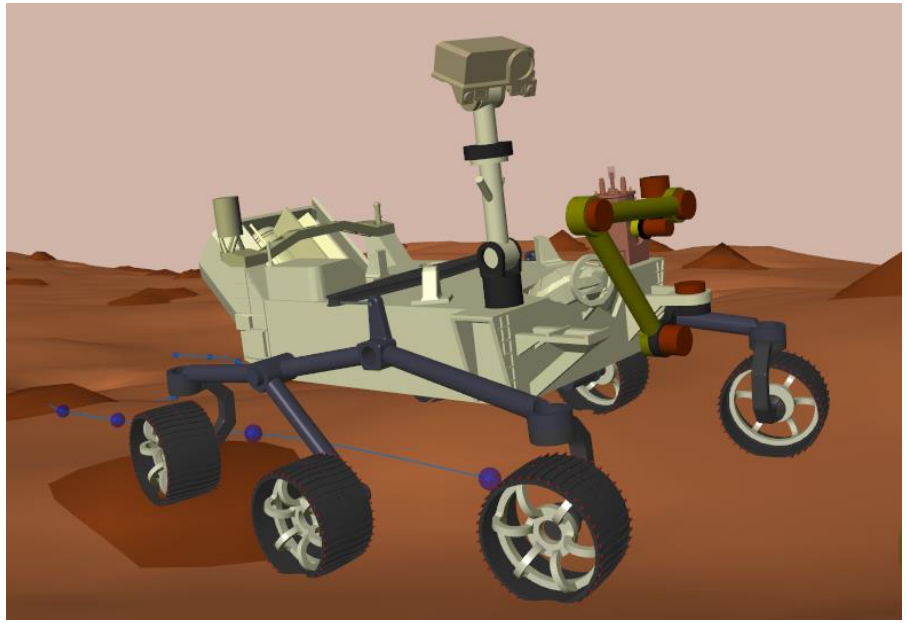
Where am I?
Calibration

What do I see?
Object Detection



Where can I go?
Navigation

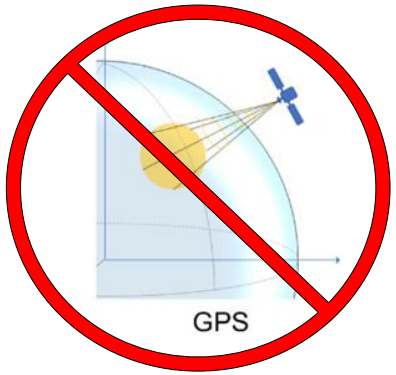
Calibrate and orient the rover using cached maps and on-board instruments



Rover Egress Phase

- Check instruments
- Check surroundings
- Rover boot up
- Calibration Path

Estimating Rover Position



No GPS on Mars → No geolocation information

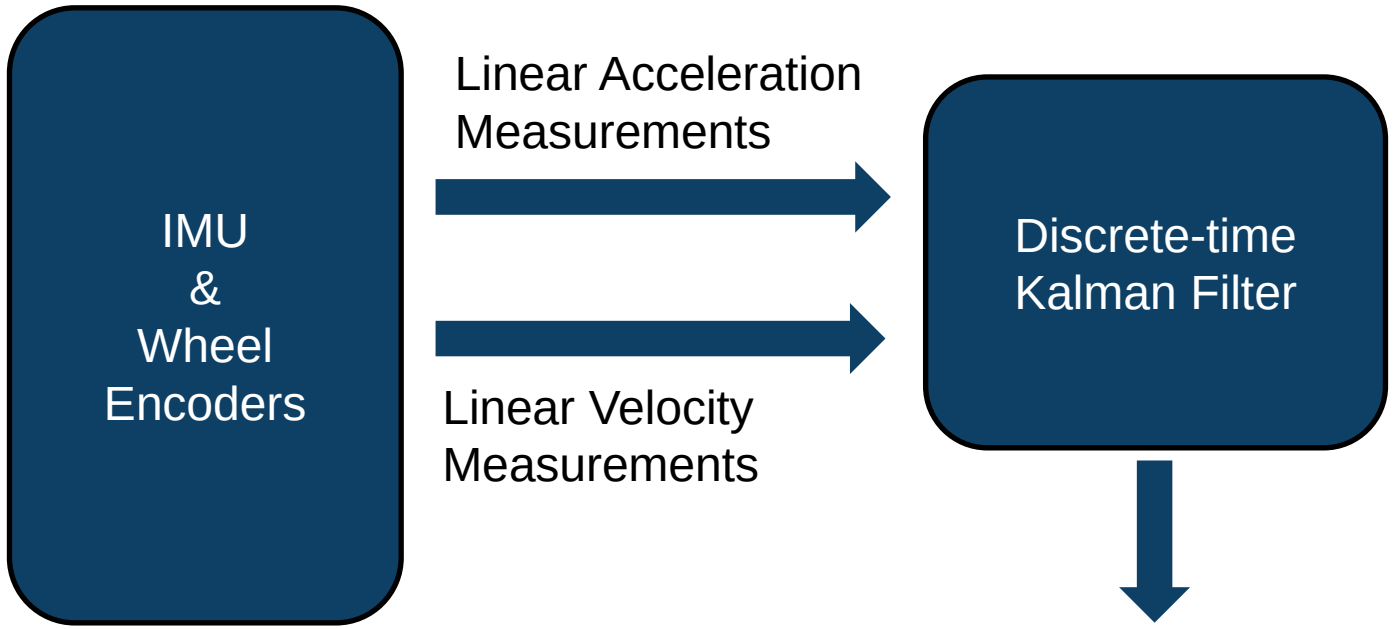
On-board Sensors

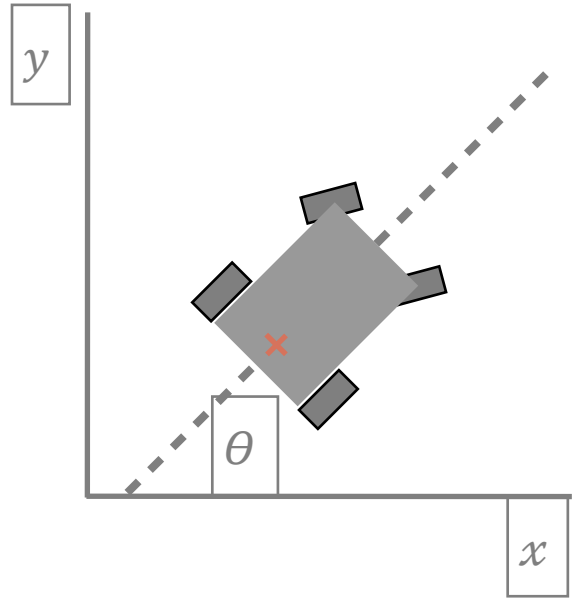
Wheel encoders can be used to estimate the linear velocity of the rover

IMU (Inertial Measurement Unit) returns acceleration measurements but results in large drift due to error accumulation

Estimating Rover Position

A Kalman filter is an **optimal estimation algorithm**.
It helps to estimate a system state when it cannot be measured directly

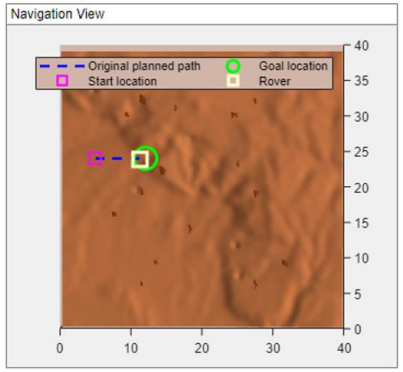




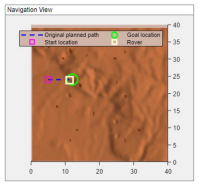
Simplified sensor models

Rover Position & Linear Velocity Estimates

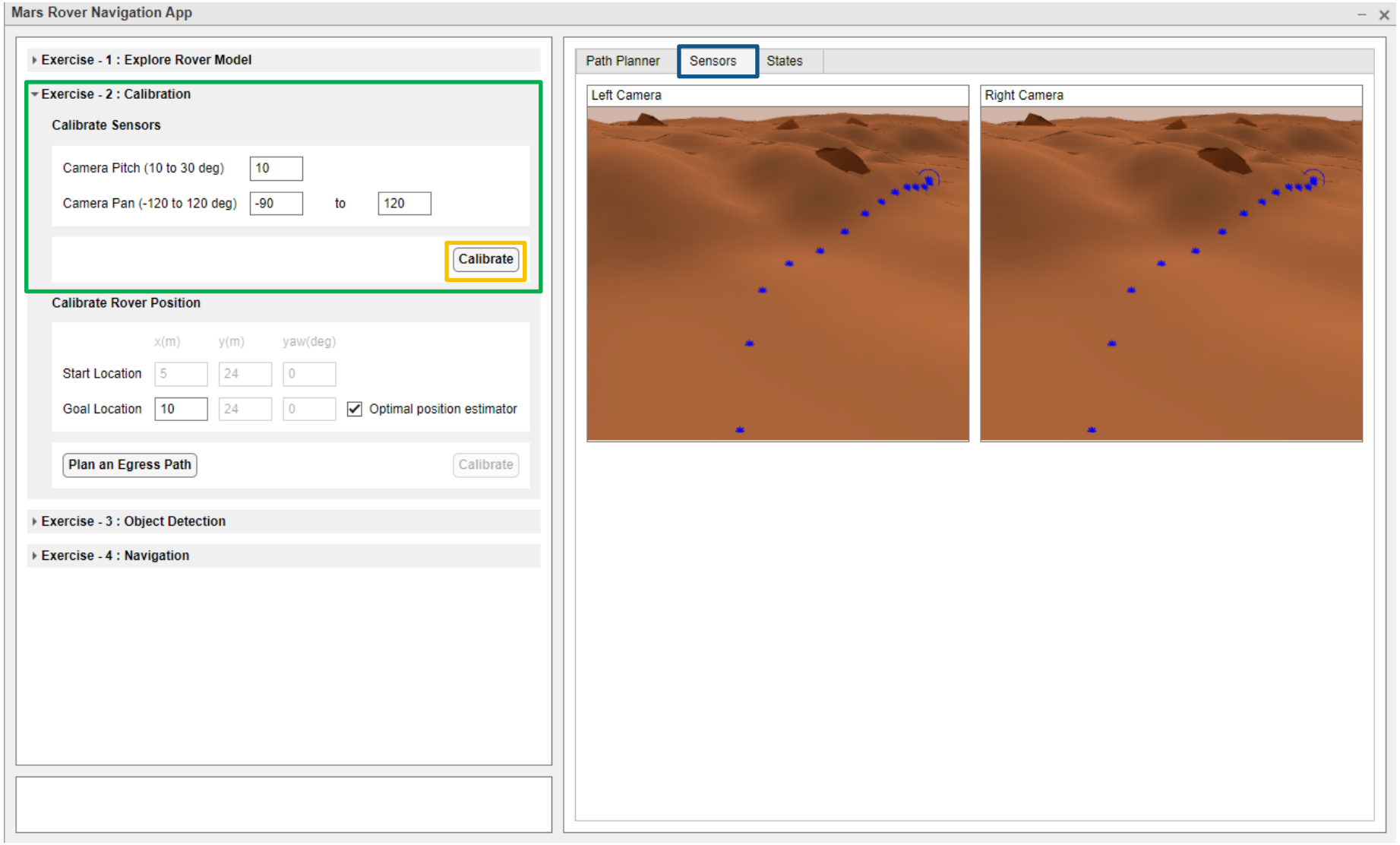
Wheel Encoders + IMU = Better Pose Estimation



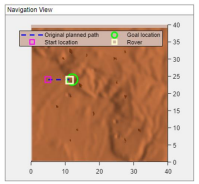
Exercise 2: Calibration



Calibrate camera



- Simulate the cameras to pan 360 degrees while the rover being stationary
- Observe the camera views and processed frames in the app to ensure they are calibrated



Calibrate linear position

Mars Rover Navigation App

Exercise - 1 : Explore Rover Model

Exercise - 2 : Calibration

Calibrate Sensors

Camera Pitch (10 to 30 deg) 10

Camera Pan (-120 to 120 deg) -120 to 120

Calibrate

Calibrate Rover Position

	x(m)	y(m)	yaw(deg)
Start Location	5	24	0
Goal Location	10	24	0

Optimal position estimator

Plan an Egress Path

Calibrate

Exercise - 3 : Object Detection

Exercise - 4 : Navigation

Path Planner | Sensors | States

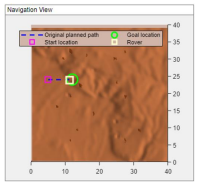
StartLocation
GoalLocation
Known Terrain Obstacles
Unknown Terrain Obstacles

Binary Occupancy Grid

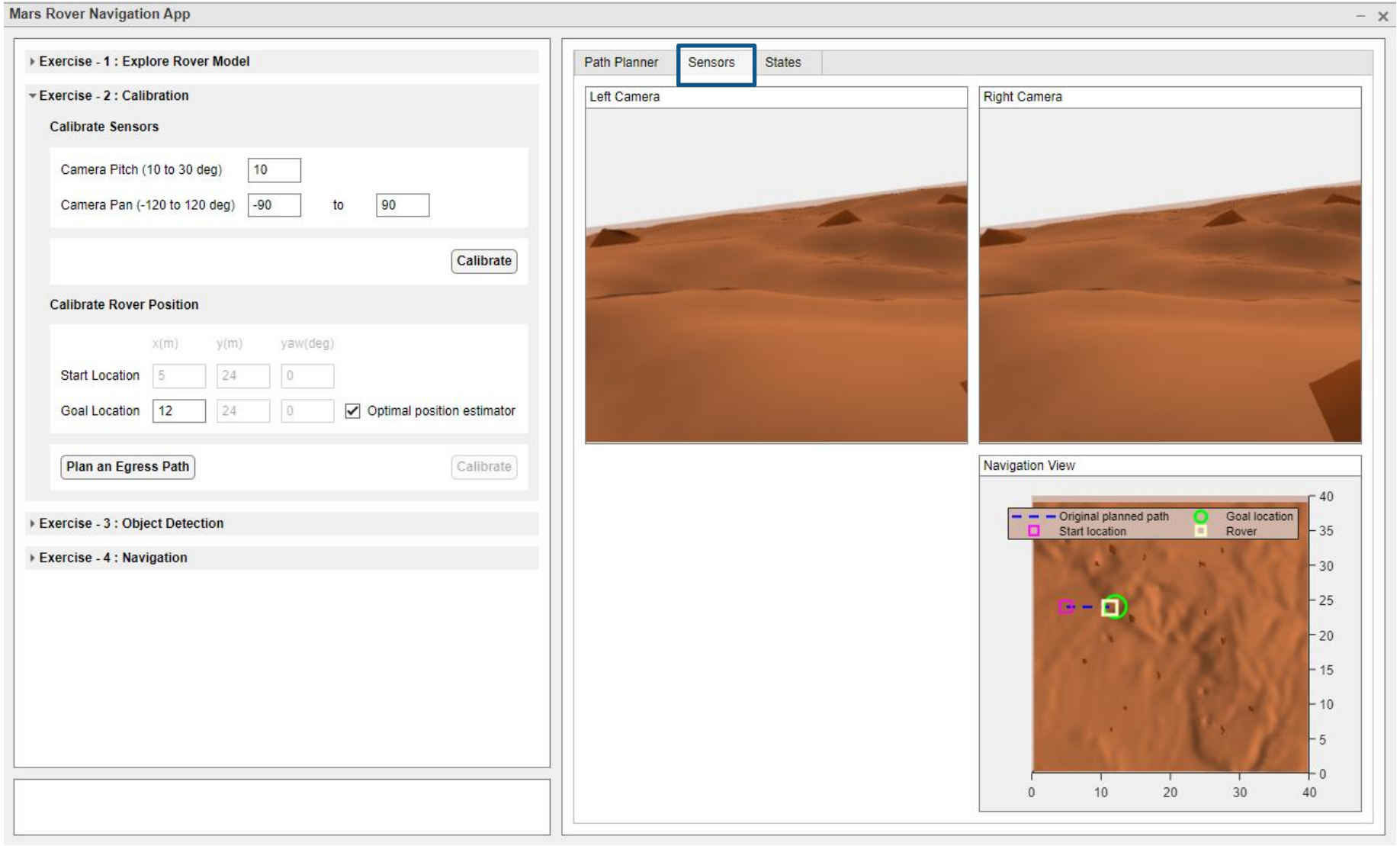
Y [meters]

X [meters]

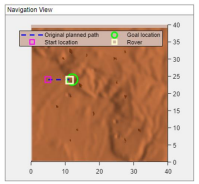
- Plan an Egress Path
- Simulate a straight path for the rover to move by clicking the calibrate button



Calibrate linear position



- Observe the Camera feed and the Rover position in the Sensors tab
- Observe the planned path and Rover position under Navigation View



Calibrate linear position

Mars Rover Navigation App

Exercise - 1 : Explore Rover Model

Exercise - 2 : Calibration

Calibrate Sensors

Camera Pitch (10 to 30 deg) 10

Camera Pan (-120 to 120 deg) -90 to 90

Calibrate

Calibrate Rover Position

	x(m)	y(m)	yaw(deg)
Start Location	5	24	0
Goal Location	12	24	0

Optimal position estimator

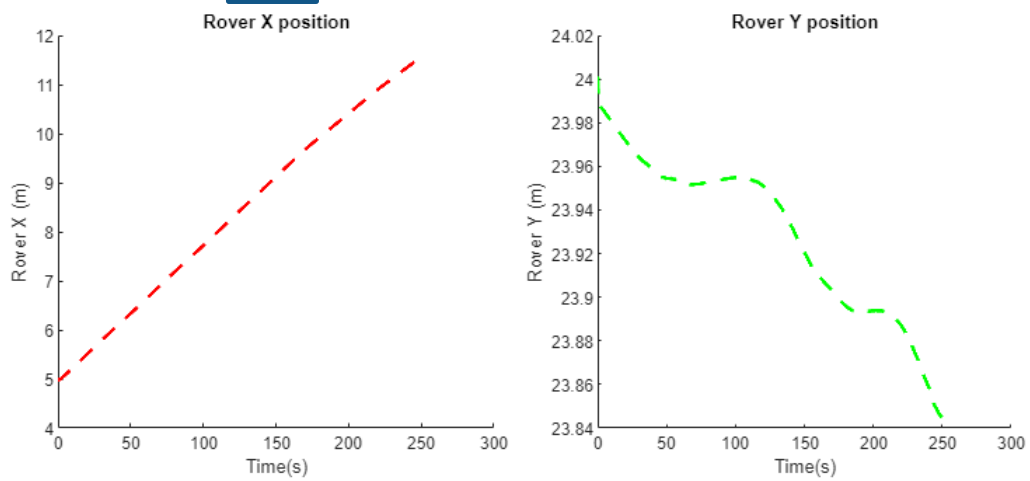
Plan an Egress Path

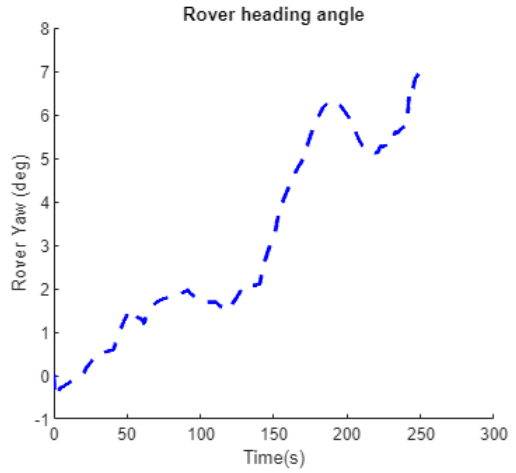
Calibrate

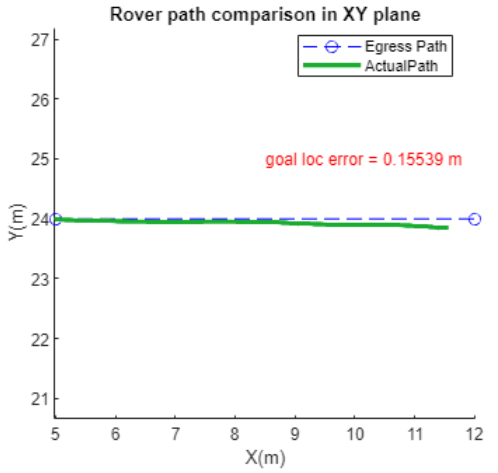
Exercise - 3 : Object Detection

Exercise - 4 : Navigation

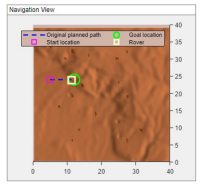
Path Planner | Sensors | States





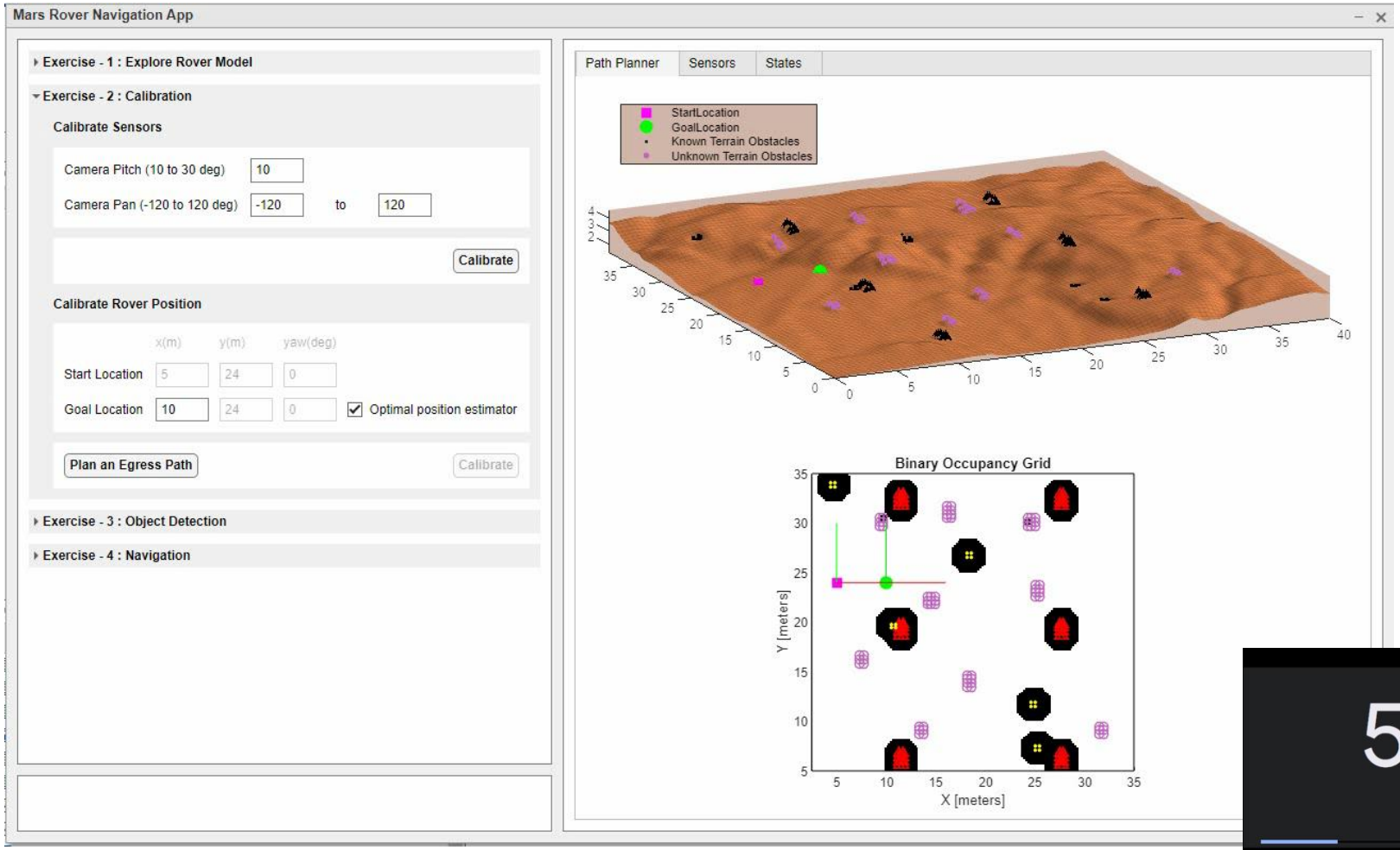


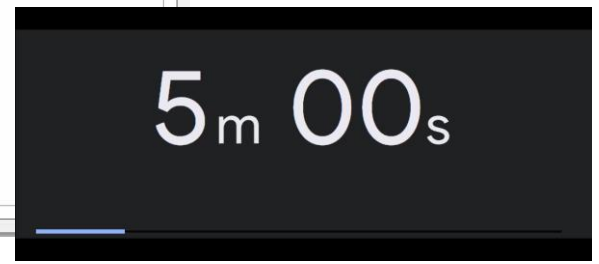
- Observe the difference in the state estimates with and without Optimal position estimator

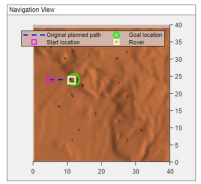


Please take 5 minutes to complete exercise 2

Need a hint? A video showing the solution will play here.



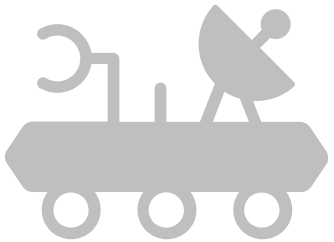




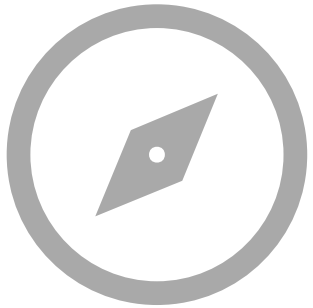
What did you observe?

- Did you get the visual feedback from the cameras?
- Did you observe any difference in the rover position values with and without using the optimal position estimator?

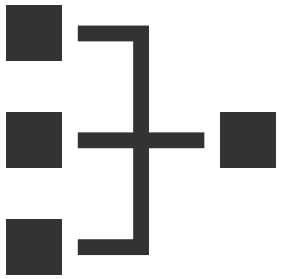
After calibration, the rover is ready to complete it's assigned tasks using AI



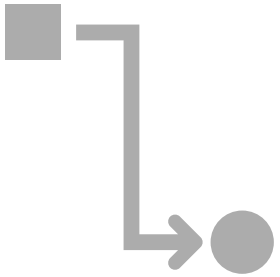
Who am I?
Rover Model



Where am I?
Calibration



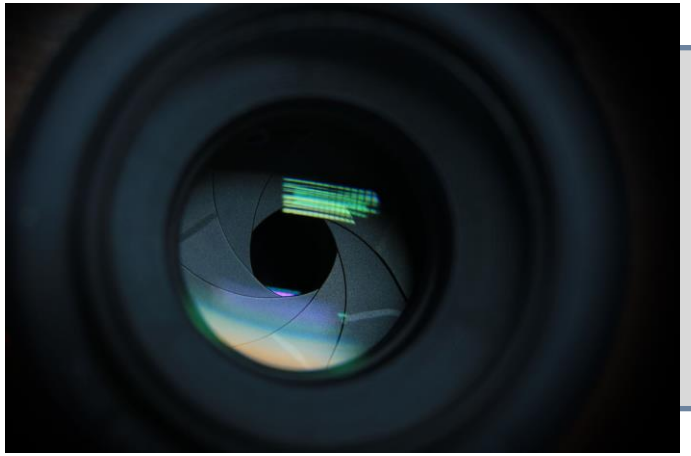
What do I see?
Object Detection



Where can I go?
Navigation

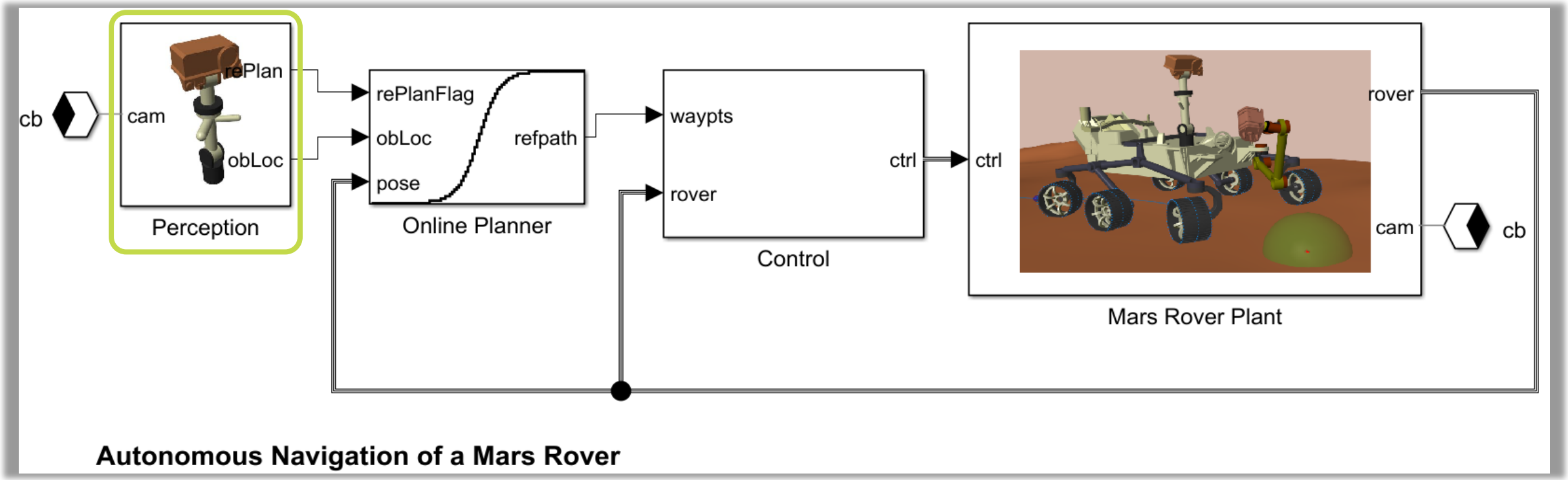
After calibration, the rover needs to traverse the terrain

Scan area, eyes closed and walk

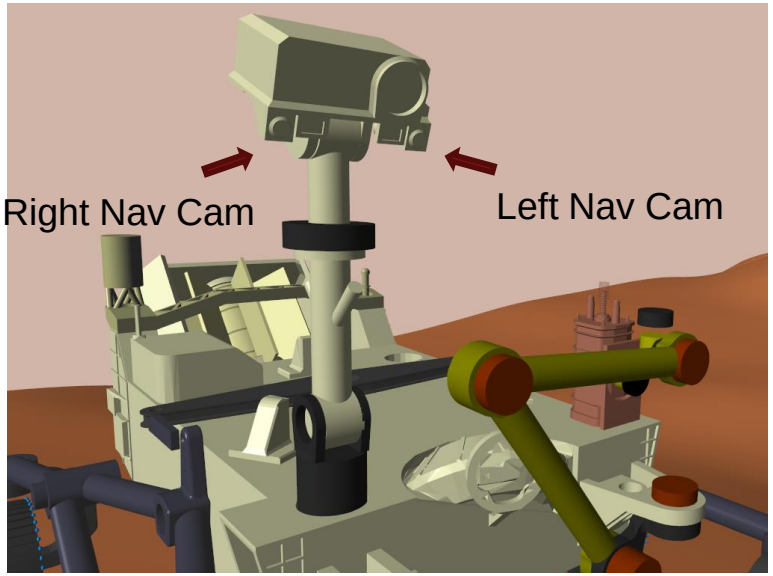


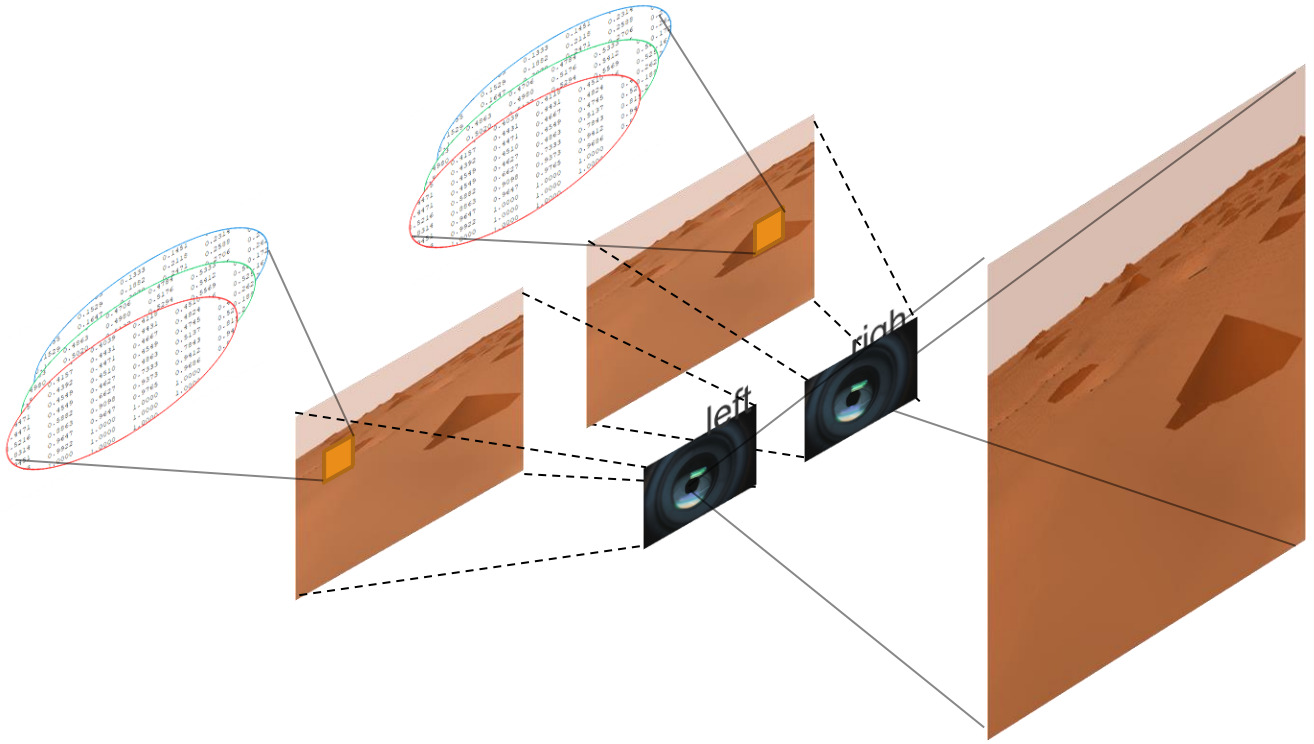
Camera sensors on the calibrated rover serve as the eyes

Perception Module



We can identify objects with vision sensors using AI techniques



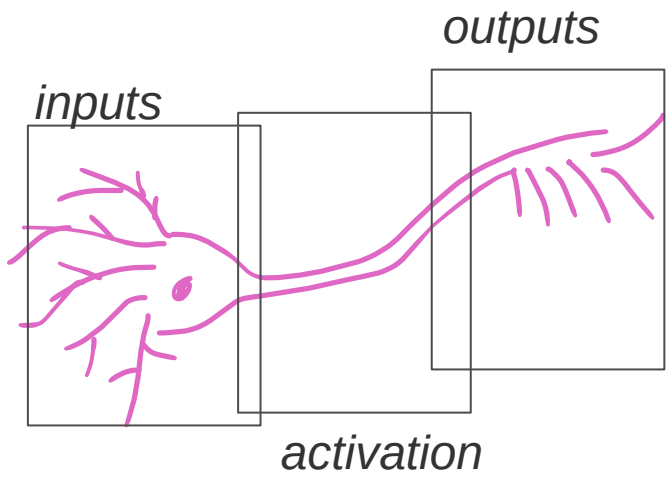


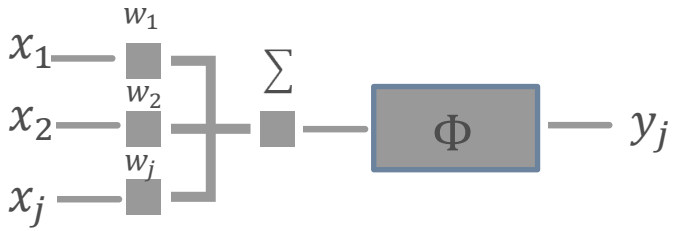
The rover collects image data using both left & right sensors

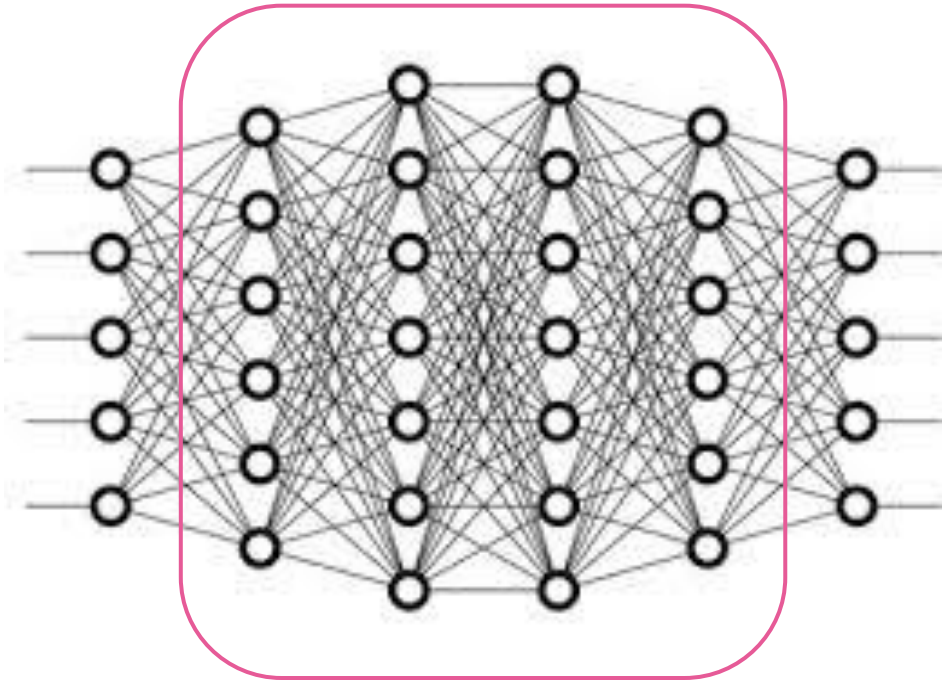
RGB matrices are fed into a DNN to identify obstacles

Artificial Neural Network

Deep Neural Networks are stacked neural networks





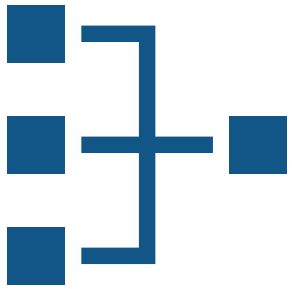


Deep Neural Network

Deep Neural Network plays a pivotal role in identifying objects

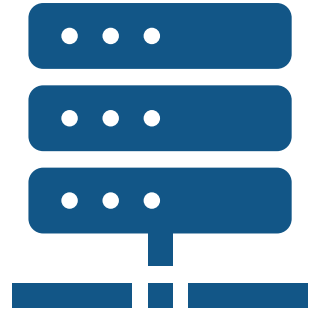
Preprocessing Data

Network is as good as your data



Object Detection Network

Resnet50 – feature extraction



Train Network

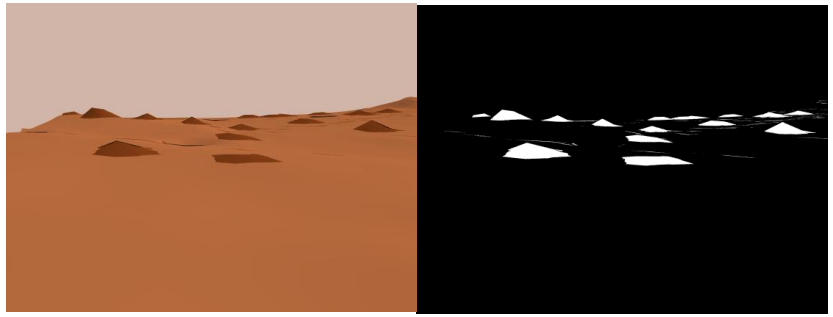
60:20:20 ratio

Validate

Use the last 20%

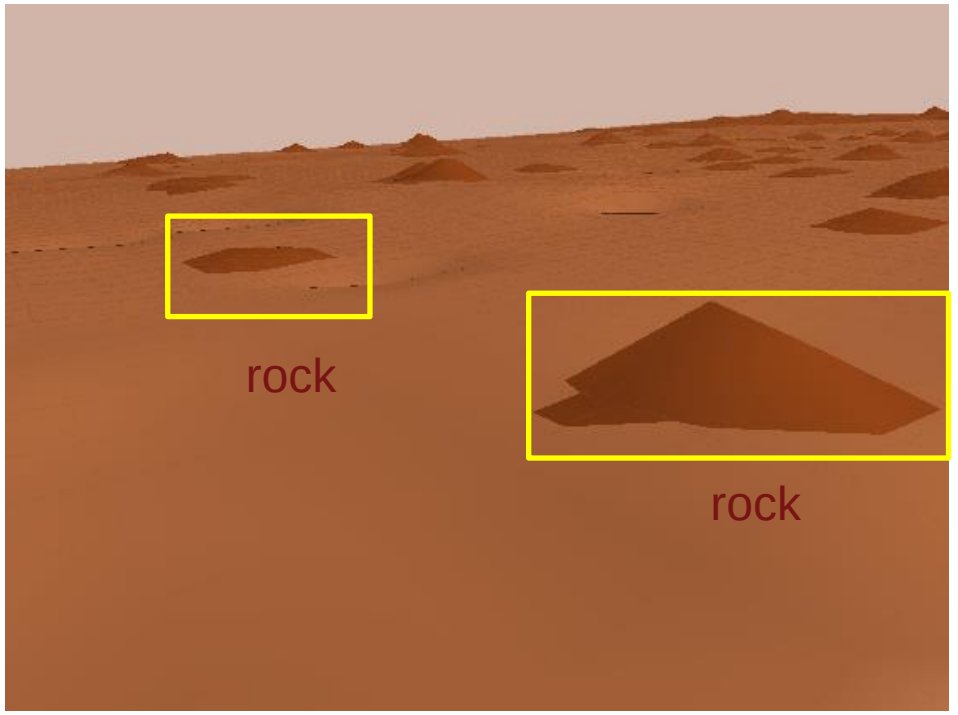
Preprocessing Data

Obstacle Detection using Deep Learning



10K training images

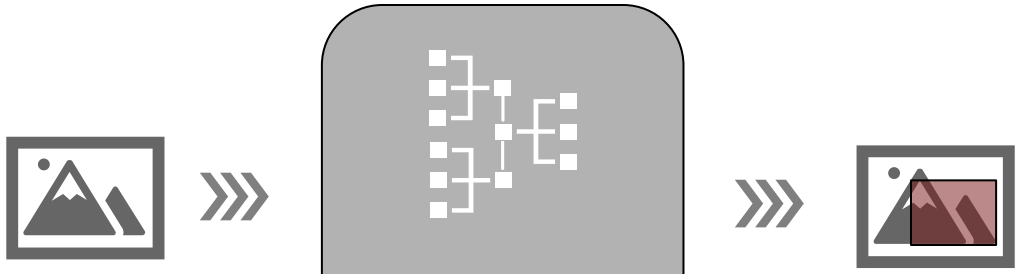
Labeled using binary masking



Yolov2 Object Detector

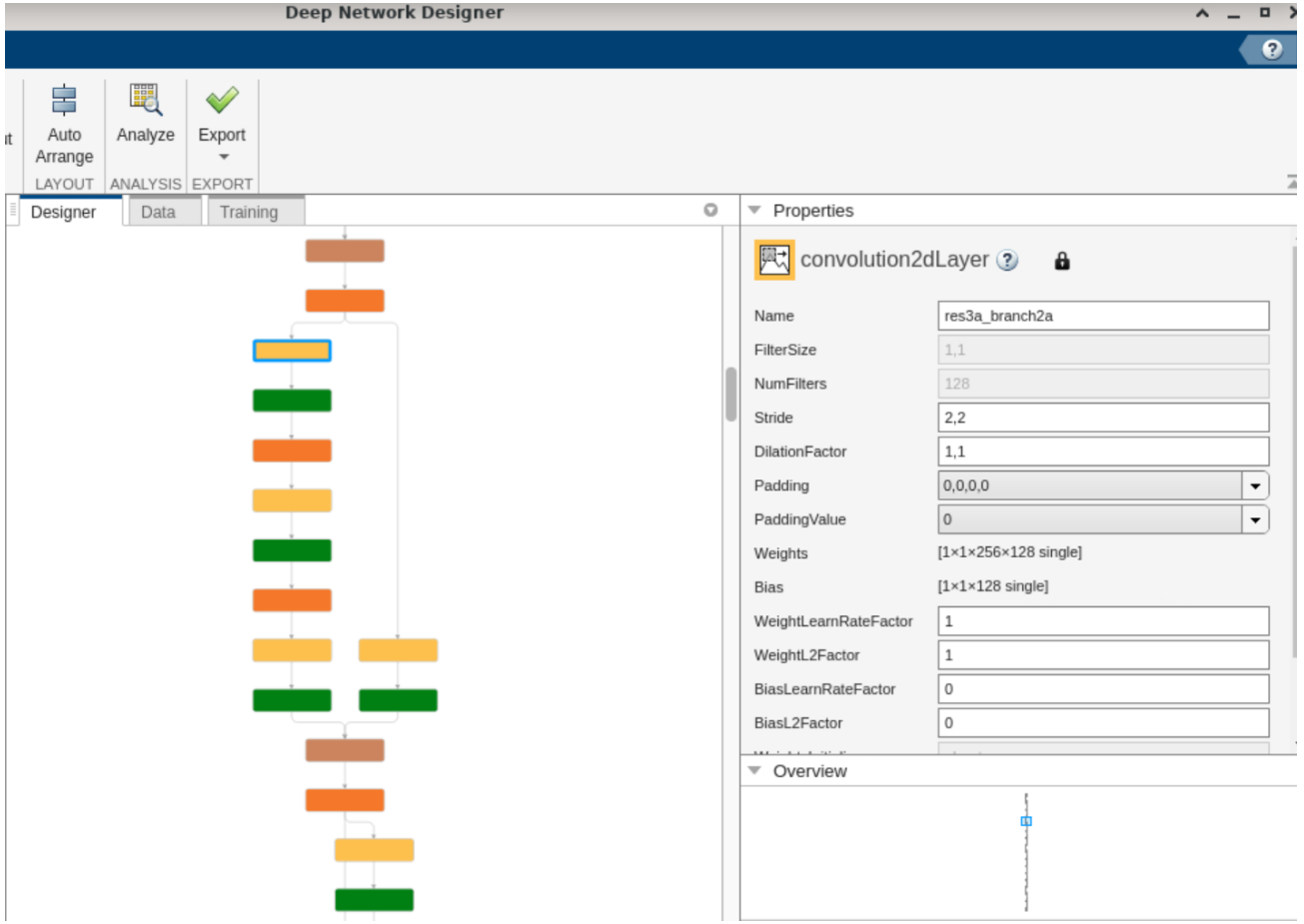
Object Detection Network

```
>> trainYOLOv2ObjectDetector()
```

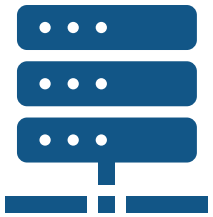


Classification + localization

Single - Stage Detection



Resnet-50 backbone network



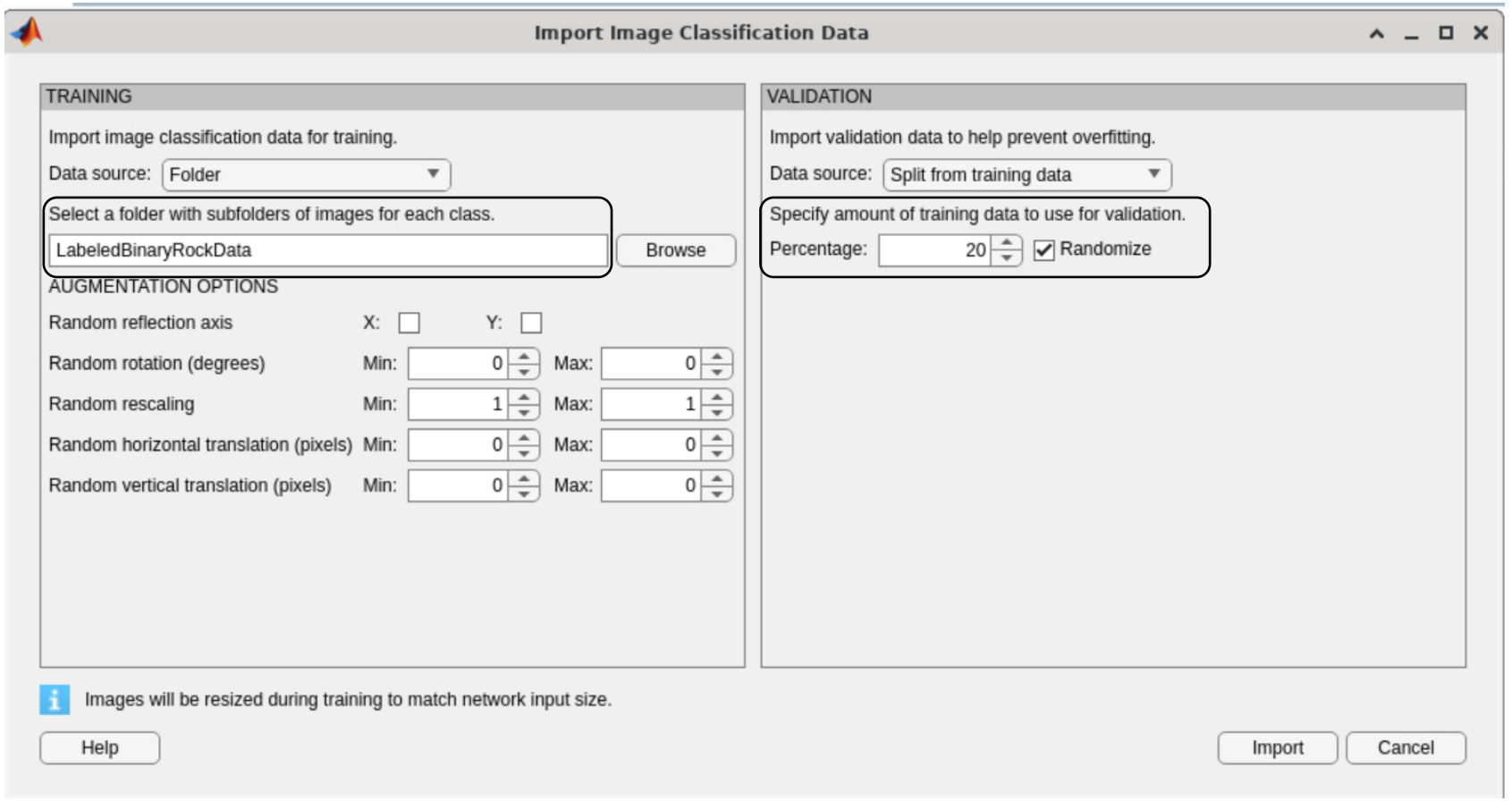
Train Network

Validate

Network training and Validation

MATLAB Deep Network Designer

Getting Started | Compare Pretrained Networks | Transfer Learning



Obstacle Depth Estimation

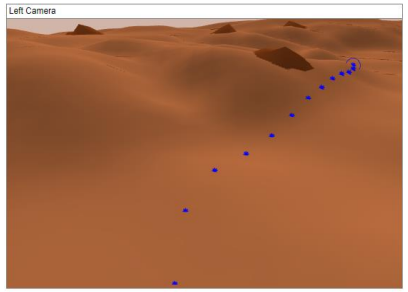
Image Processing Toolbox

Stereo Camera Calibration

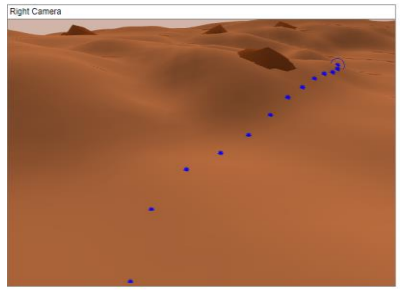
Reconstructed 3D scene

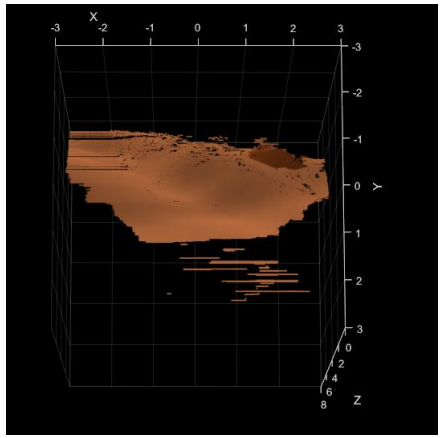
Disparity Map

Stereo vision is the process of extracting 3D information from multiple 2D views of a scene



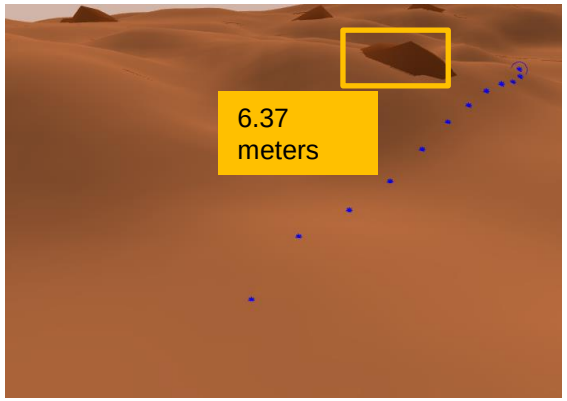
+





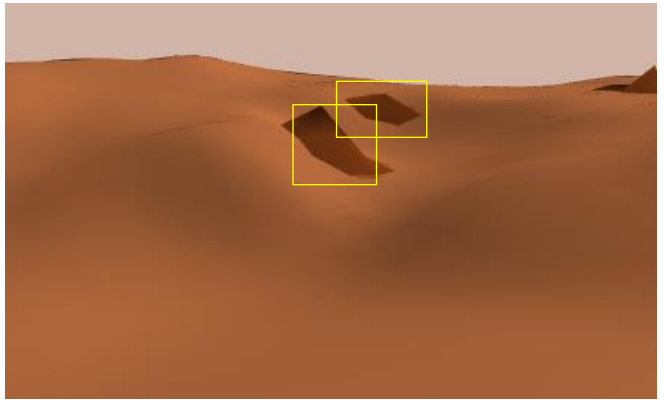
3D point cloud

Reconstructed scene

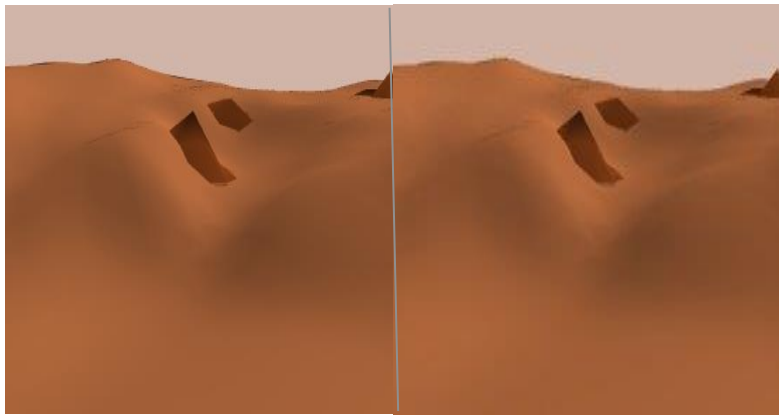


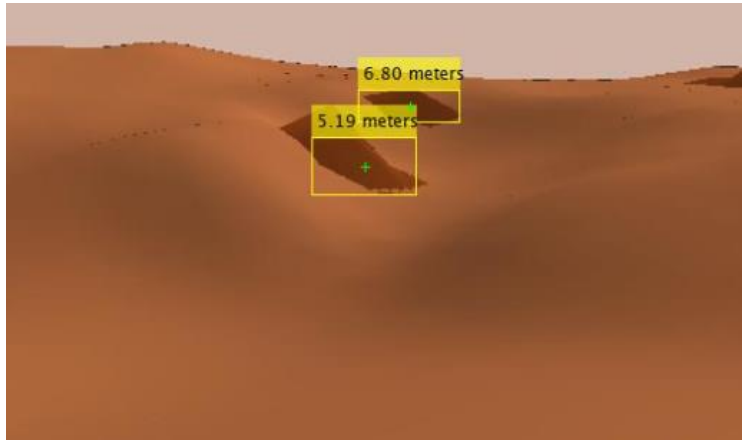
Obstacle detection

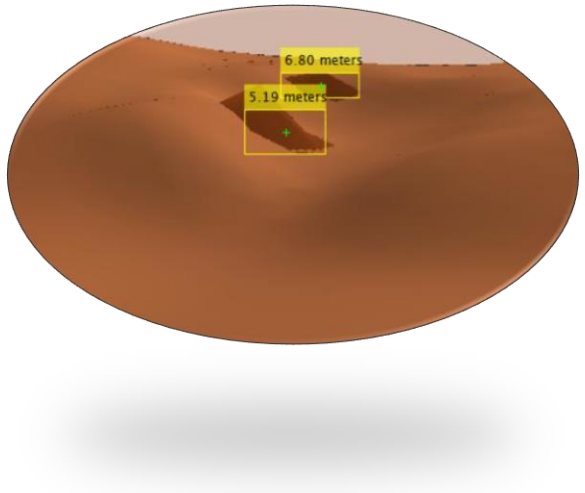
Object detection



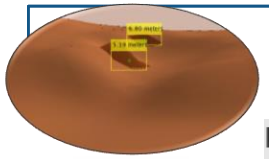
Depth Estimation







Exercise 3 : Obstacle Detection

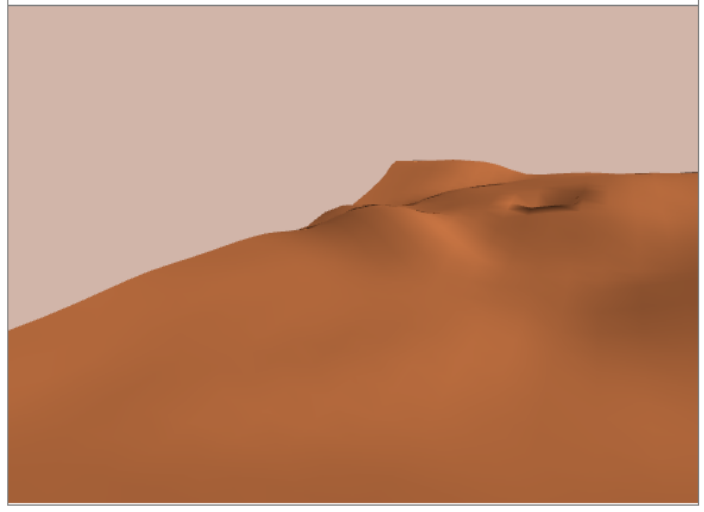


Mars Rover Navigation App

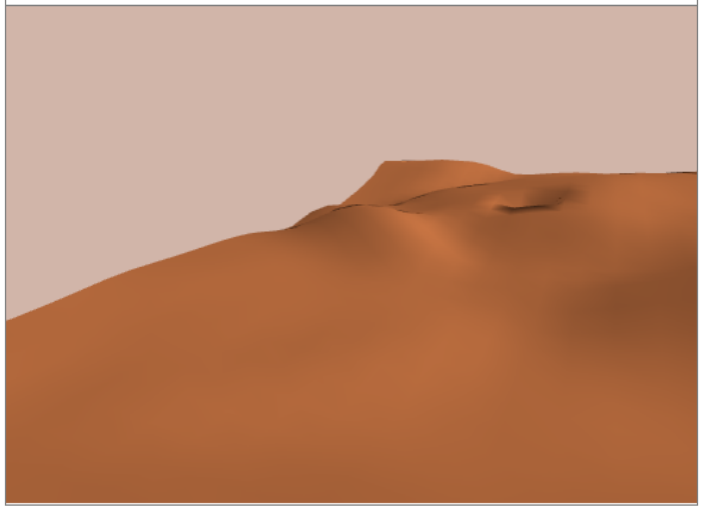
- Exercise - 1 : Explore Rover Model
- Exercise - 2 : Calibration
- Exercise - 3 : Object Detection
  - Camera Pan (-120 to 120 deg) -120 to 120
  - Detection Threshold (0 to 1) 0.55
  - Detect
- Exercise - 4 : Navigation

Path Planner | Sensors | States

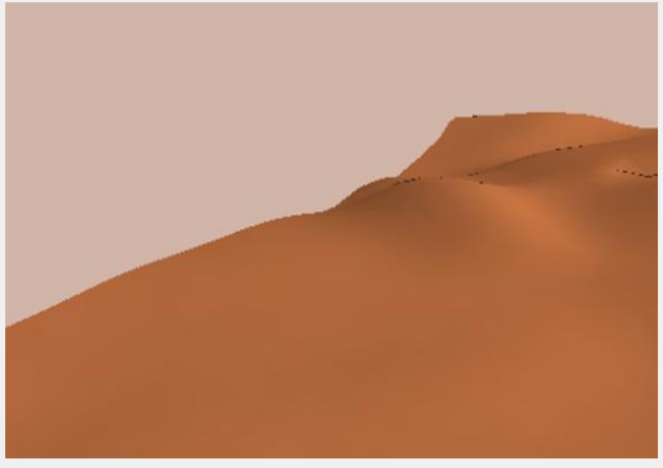
Left Camera



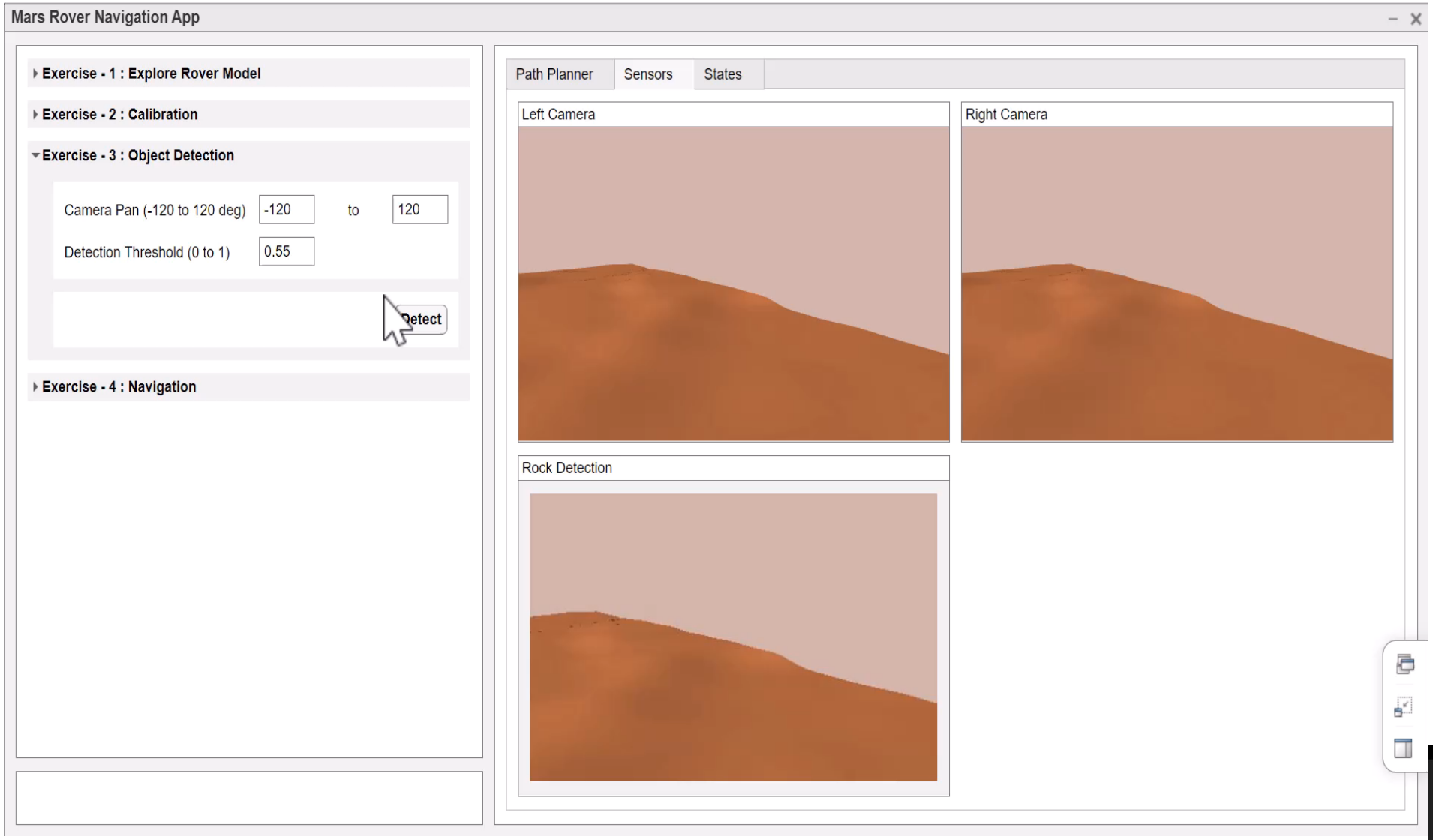
Right Camera

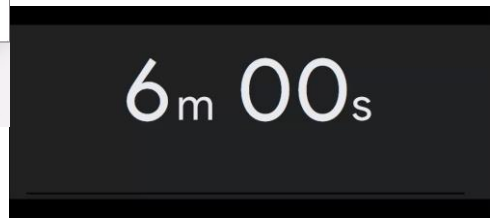


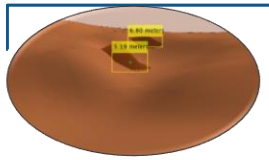
Rock Detection



Please take 6 minutes to complete exercise 3



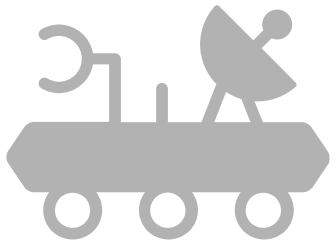




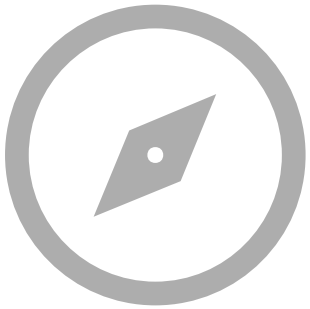
What did you observe?

- What objects did the detector find?
- Any false positives?
- Any false negatives?

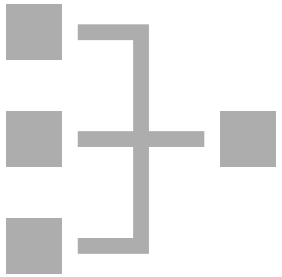
Onto the next task…



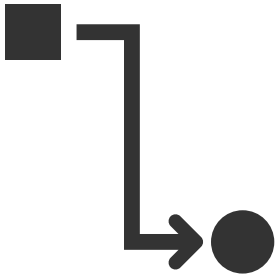
Who am I?
Rover Model



Where am I?
Calibration

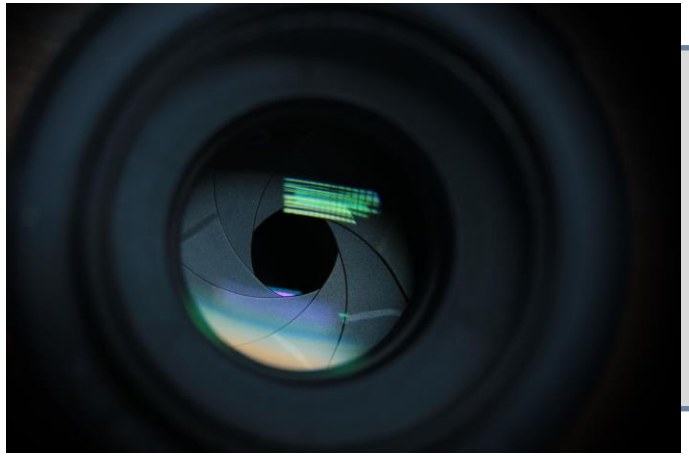


What do I see?
Object Detection

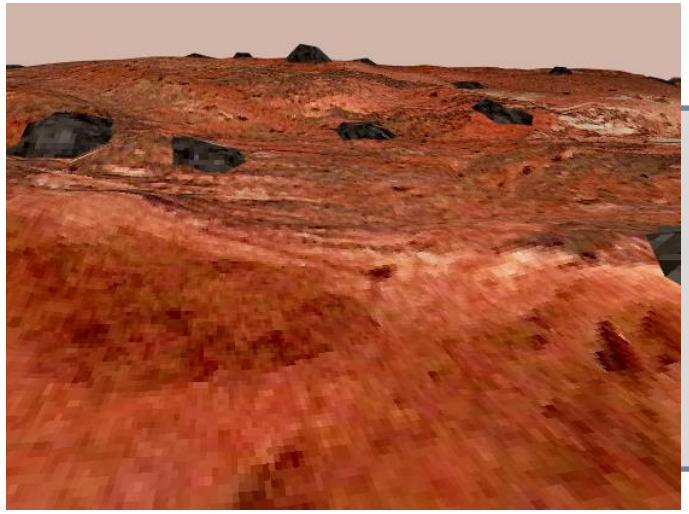


Where can I go?
Navigation

After correctly identifying objects, we need to traverse



Recall our previous exercise, we can now detect objects in our path



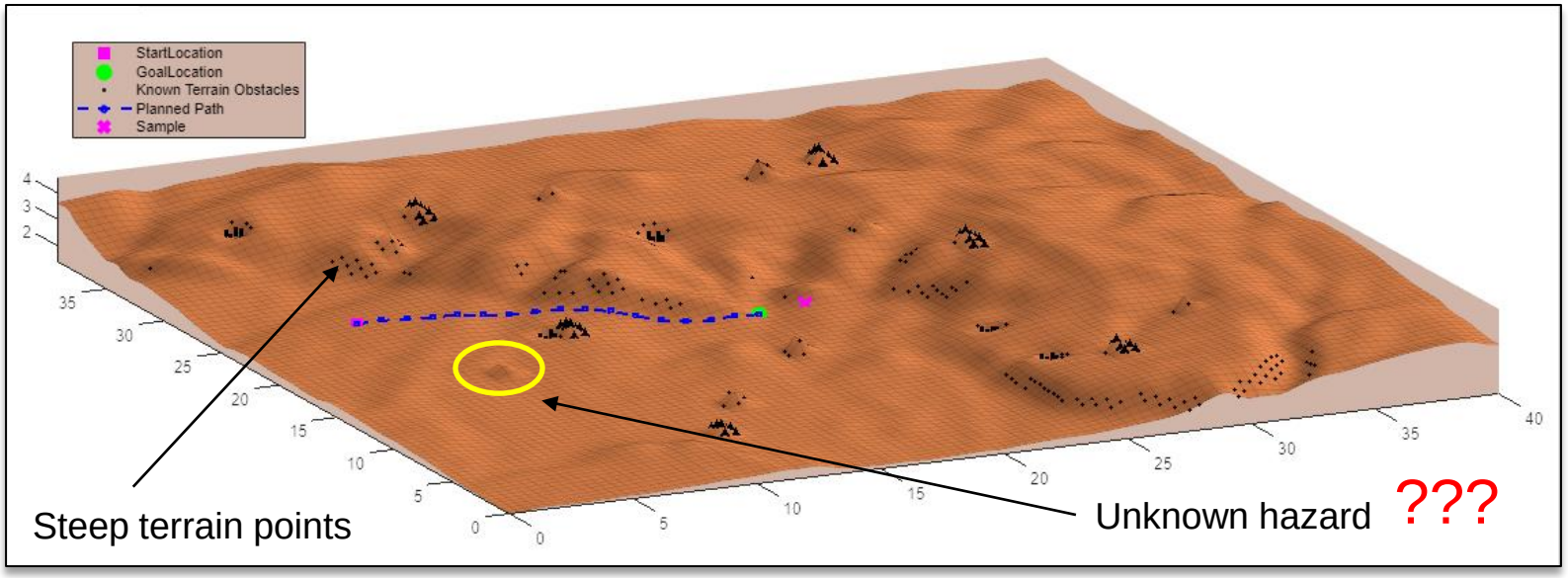
With predetermined maps and updates from DNN, the rover should be able to carry out its mission successfully

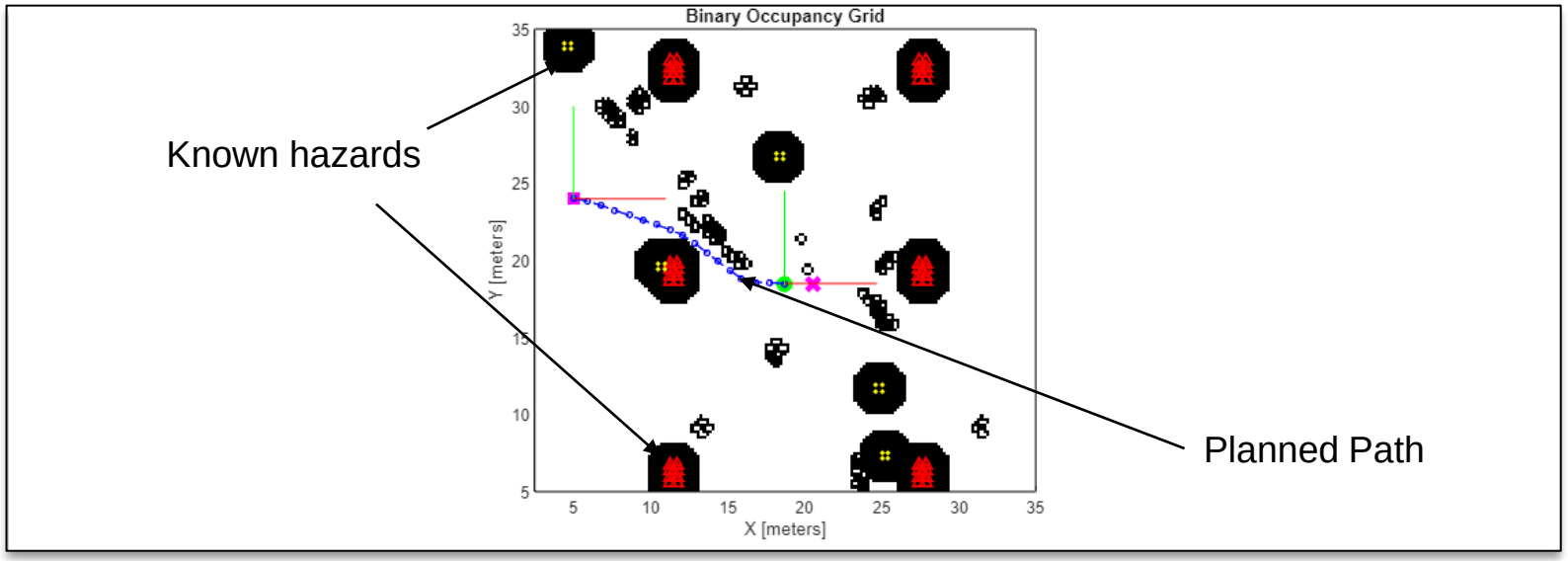
Offline Path Planning

3D terrain map known using point cloud reconstruction from multiple satellite / stereo vision images.

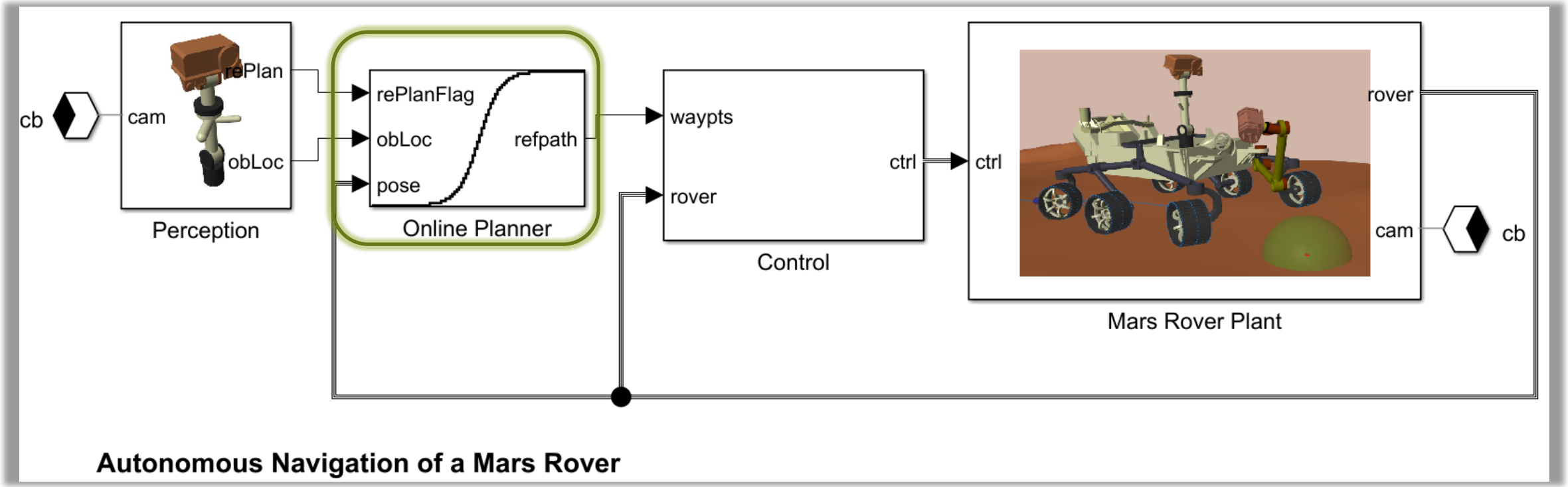
Some obstacles are assumed to be known using / terrain analysis. offline mapping.

Hybrid A star algorithm is used to plan paths avoiding the known obstacles.

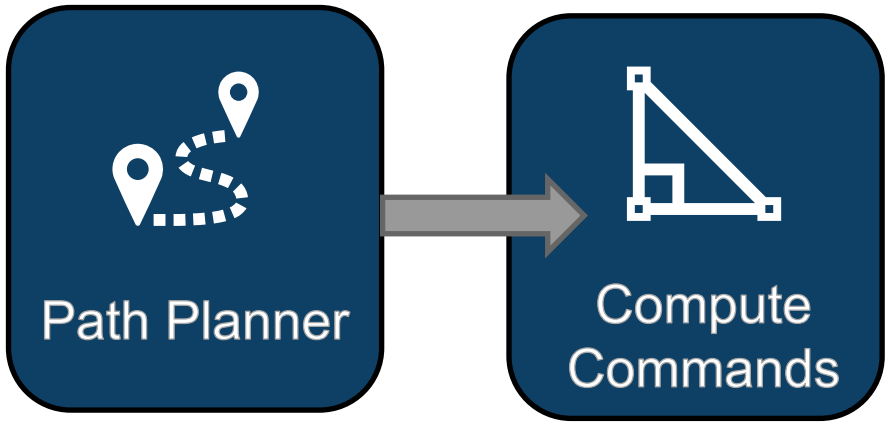




Online Planner / Obstacle Avoidance



Obstacle Avoidance helps navigate in unknown terrains



A to B?

Find an optimal path using A*

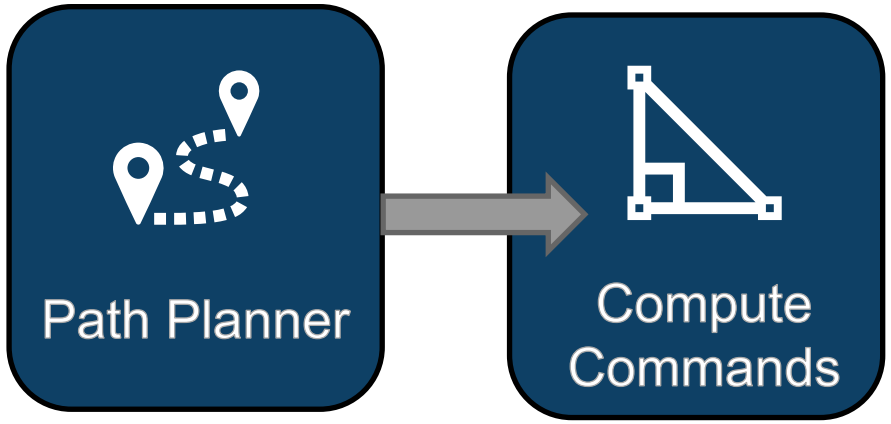
Result: Waypoints

Waypoints are a set of coordinates

Compute commands using Pure Pursuit

Result: Velocity and Theta

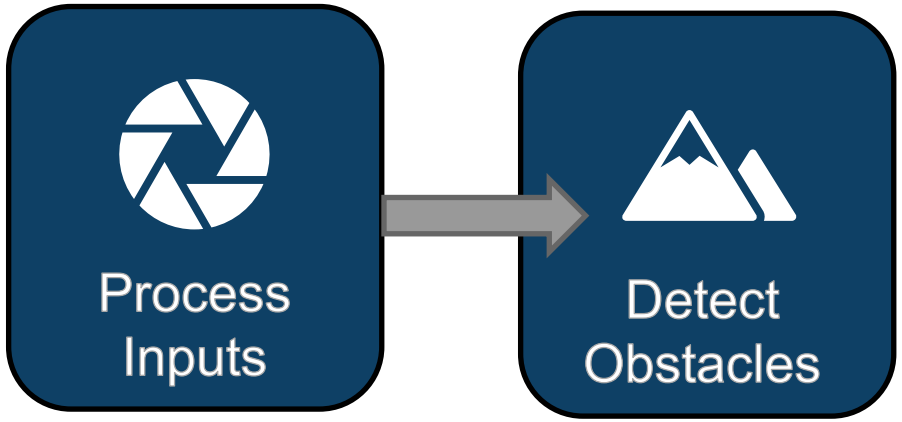
Obstacle Avoidance helps navigate in unknown terrains



Scan surroundings

Process frames from both cameras

Result: Frames

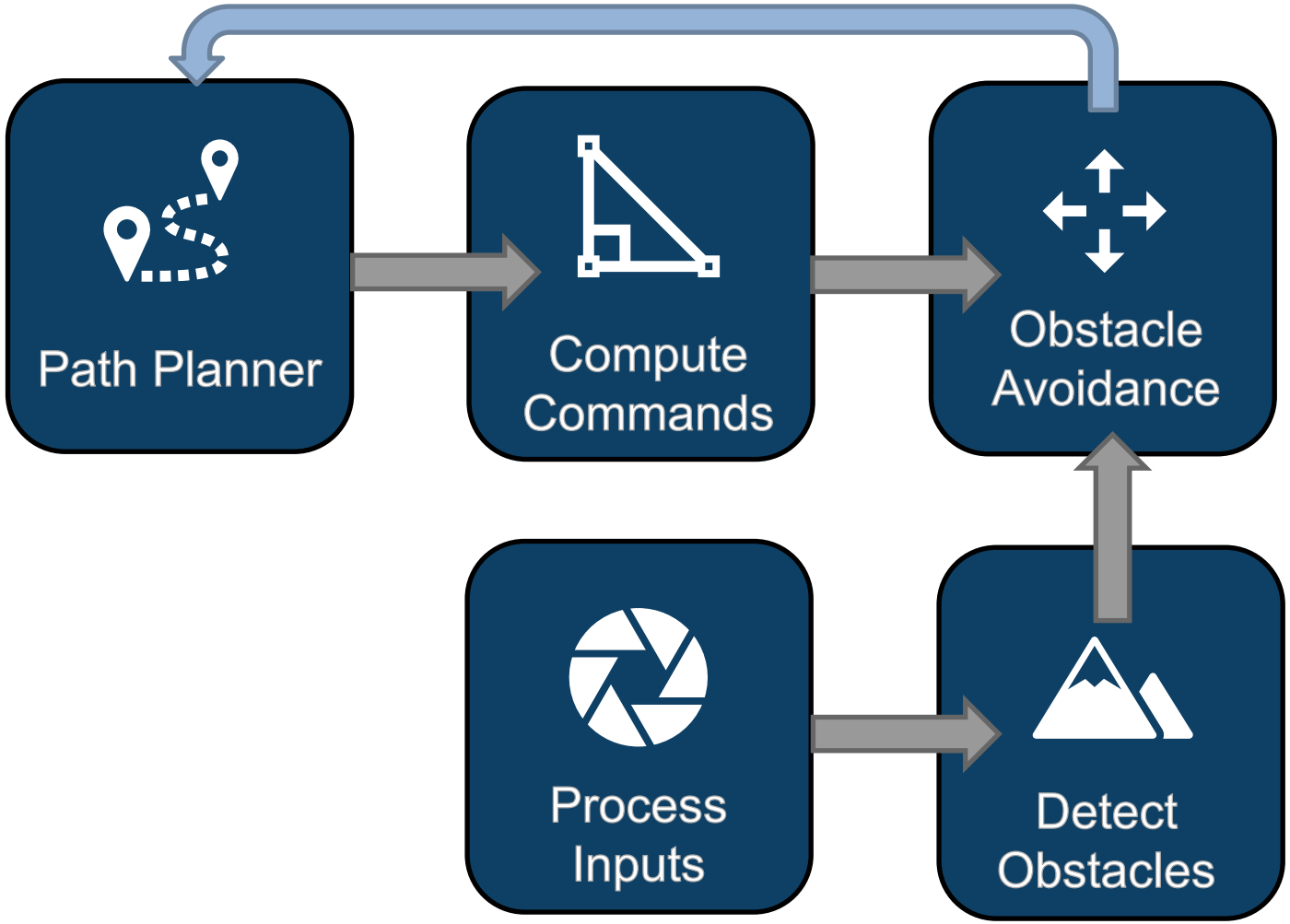


Pre-trained network

Process using stereo vision

Result: Bounding boxes & Distance

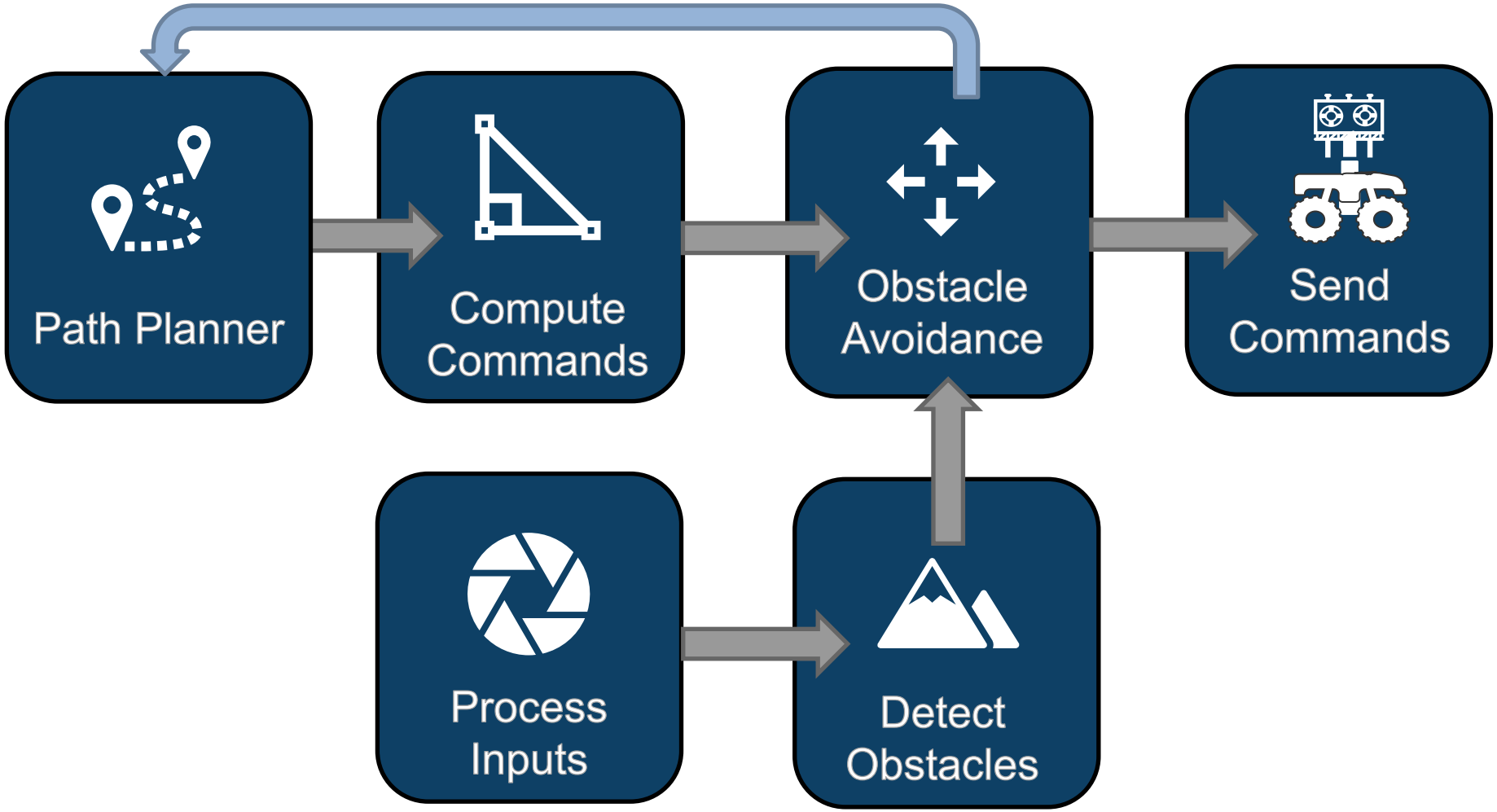
Obstacle Avoidance helps navigate in unknown terrains



Check if obstacle lies on planned path

Result: Poses & Flag

Obstacle Avoidance helps navigate in unknown terrains

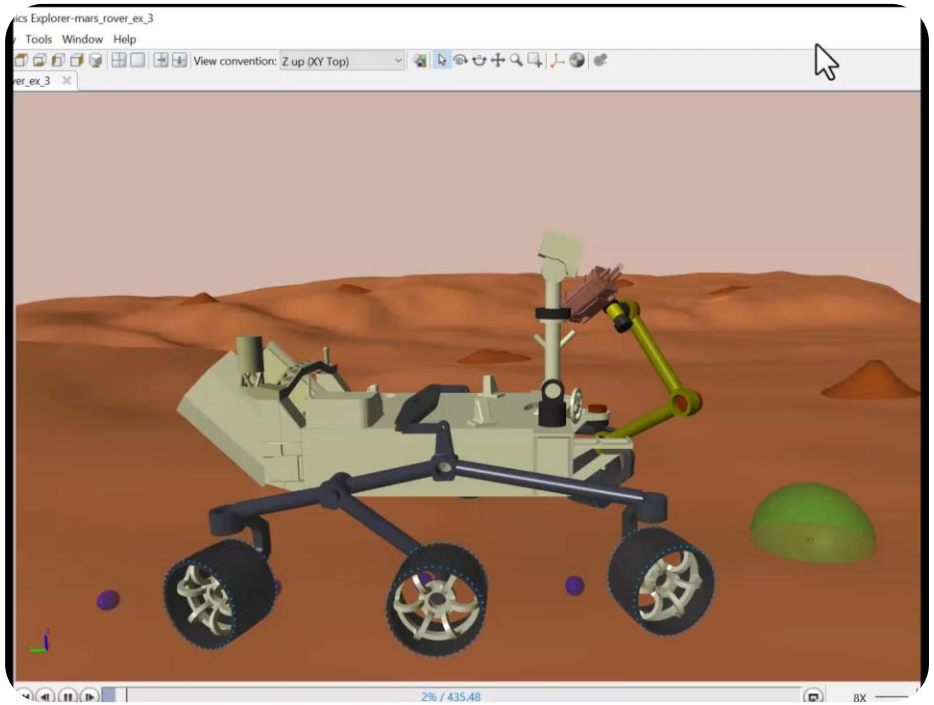


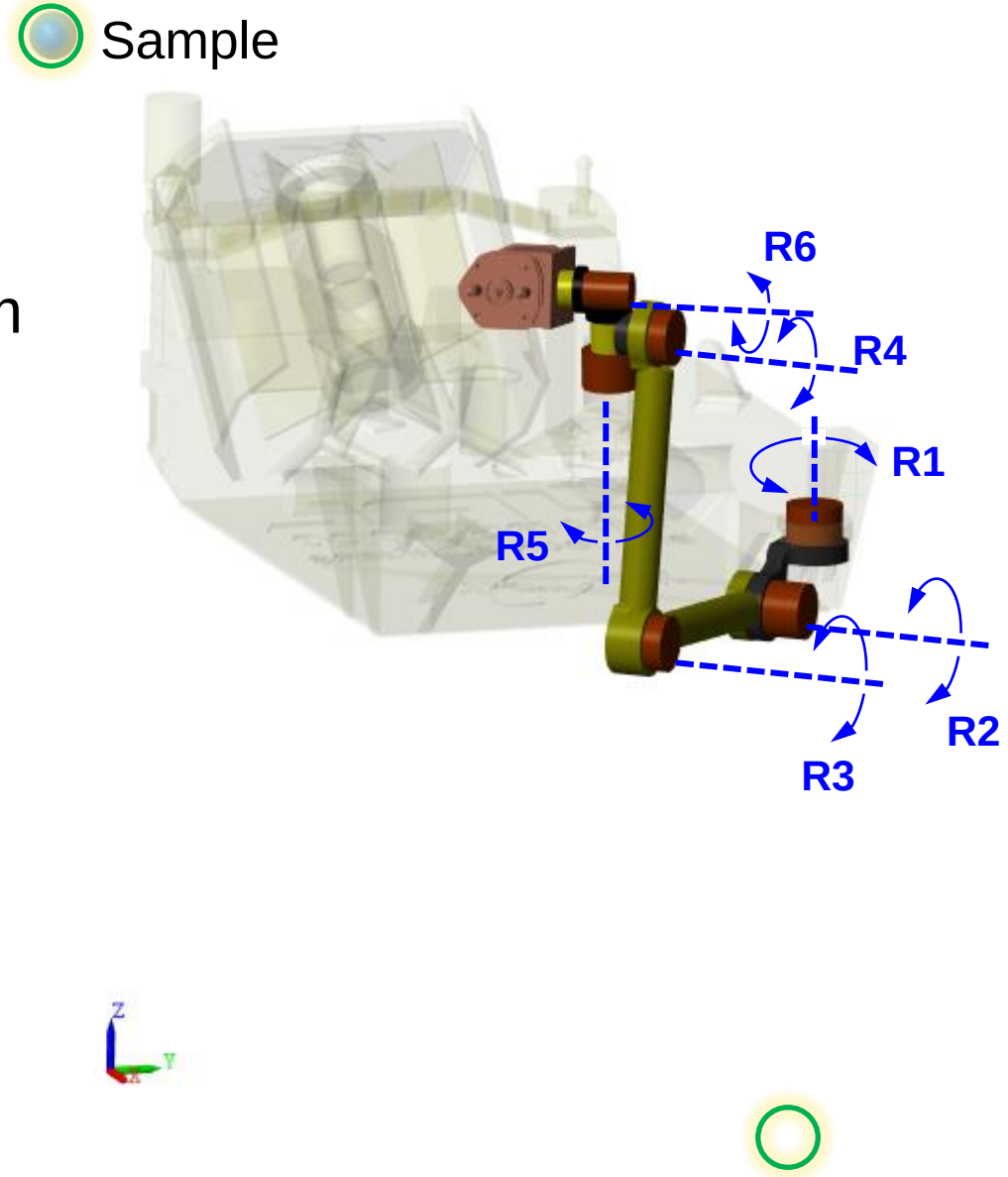
Send final velocity & heading commands to the rover

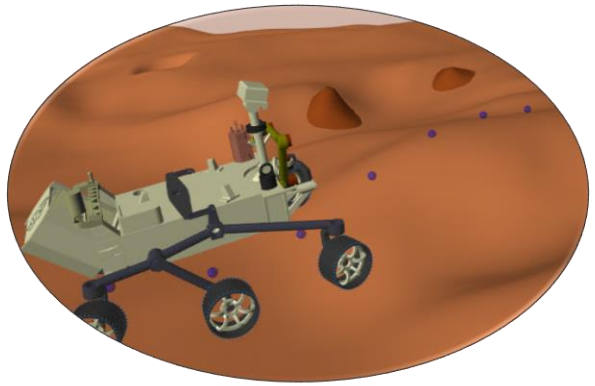
Result: Poses

Mars Rover Arm

- Six degrees of freedom
- Inverse kinematics algorithm to control the arm

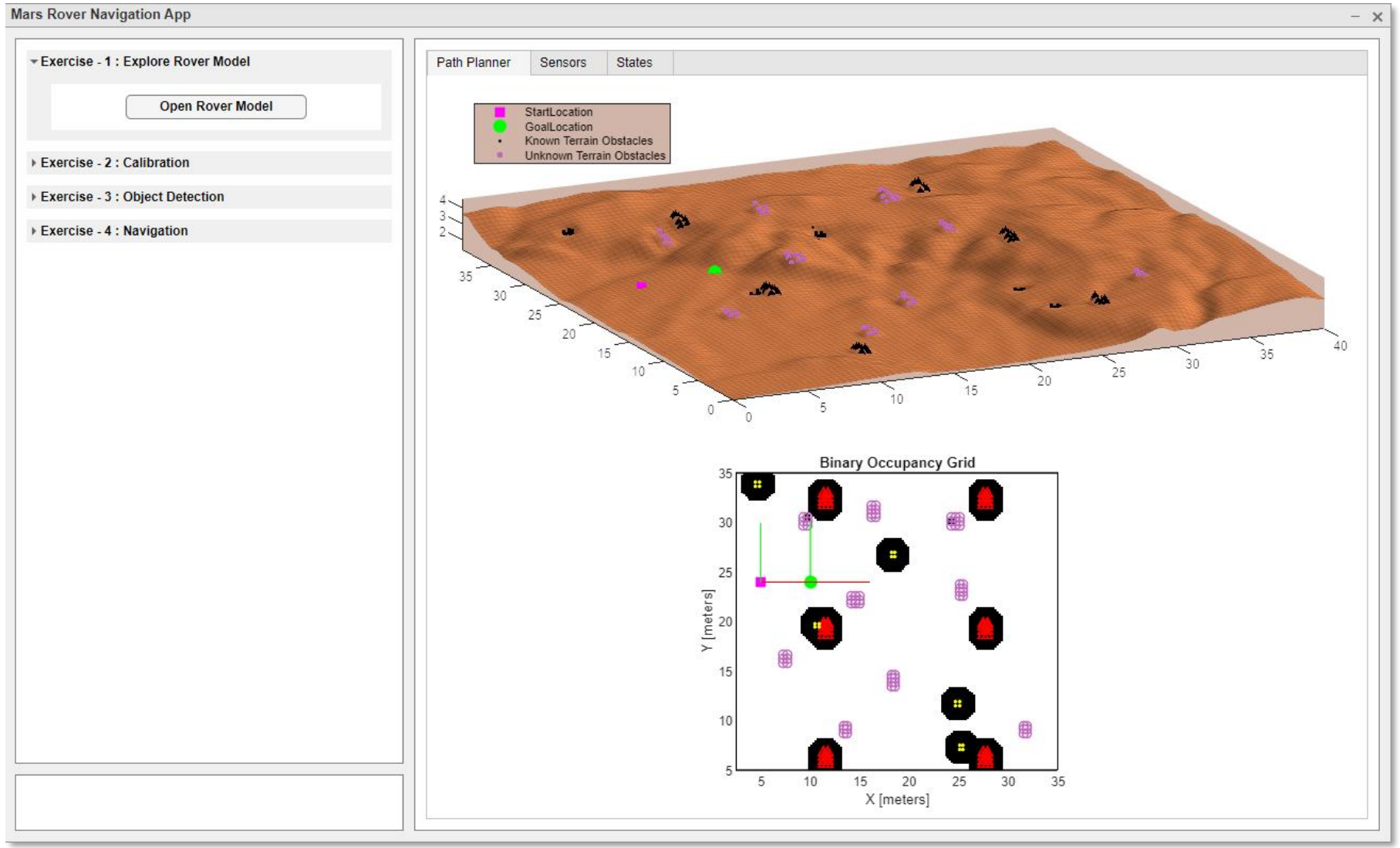






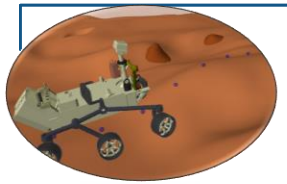
Exercise 4: Navigation

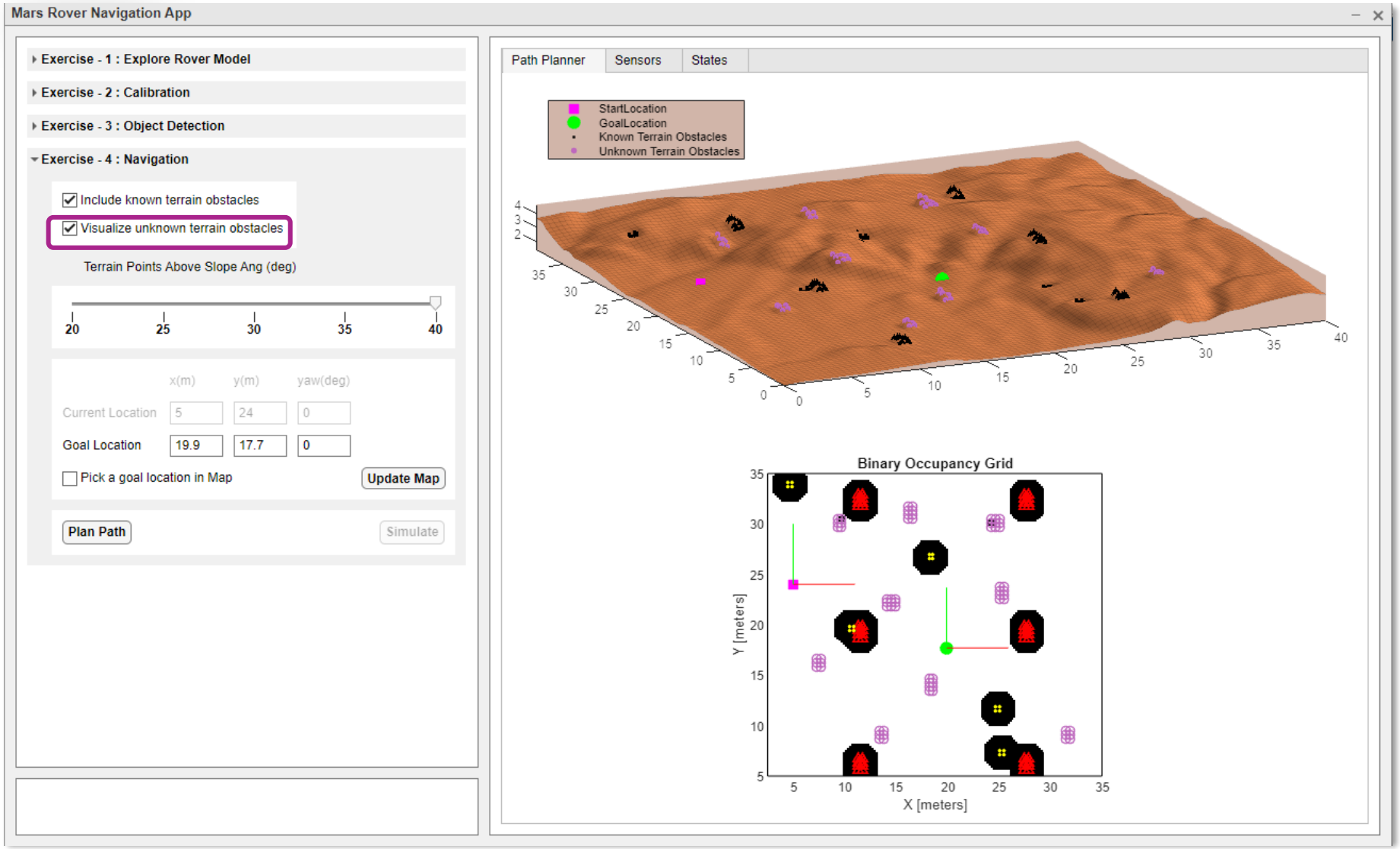
Let’s dive into our hands-on exercise



Navigate to a specific goal location

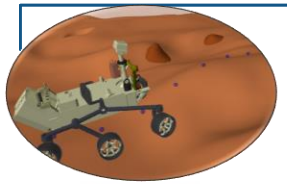
- Let’s explore the app and help navigate the rover

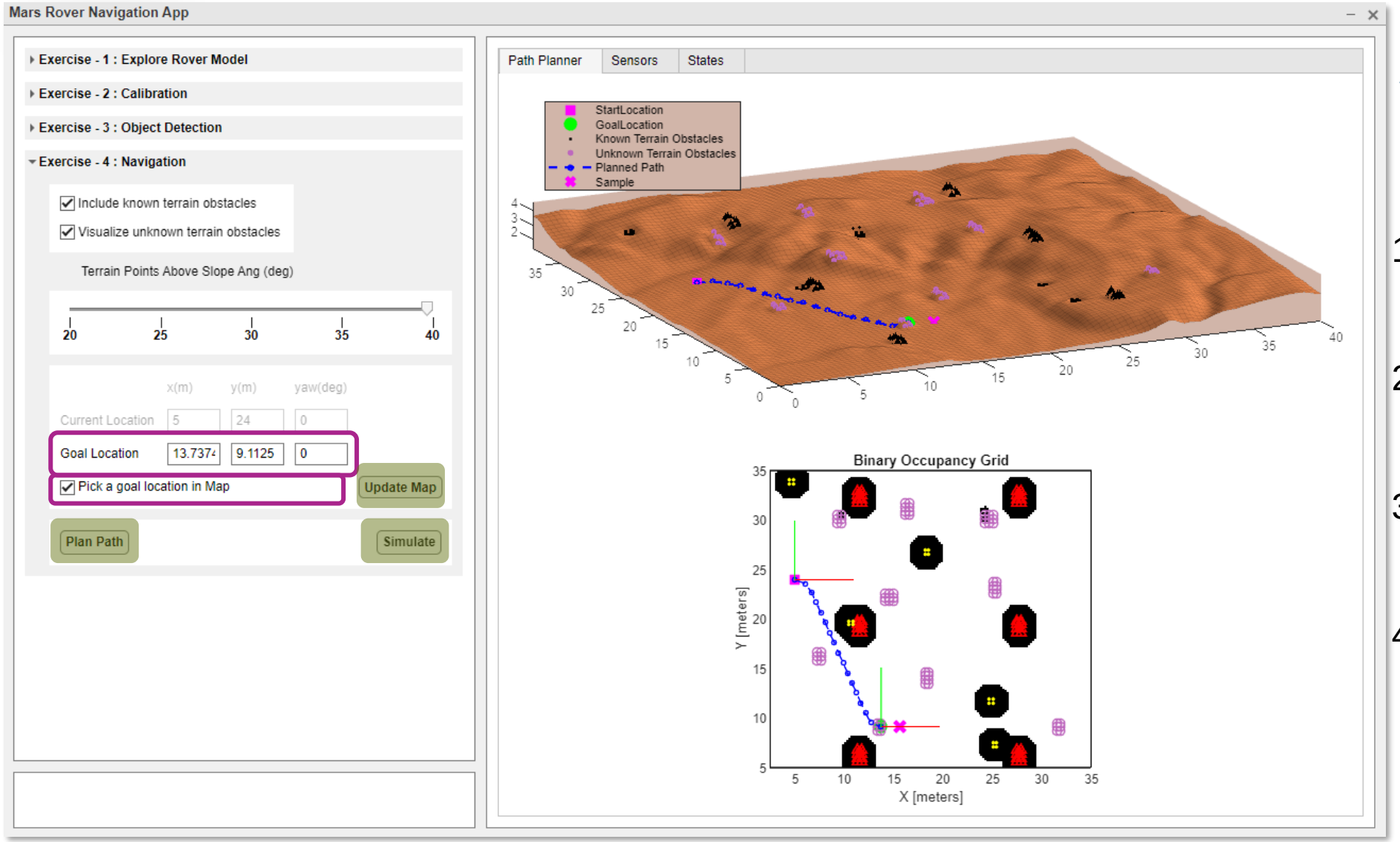




Choose to include/exclude unknown rocks

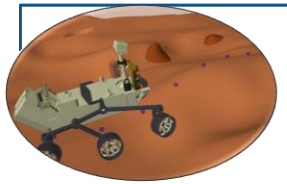
1. Select the checkbox
2. Observe the updates in the plot area





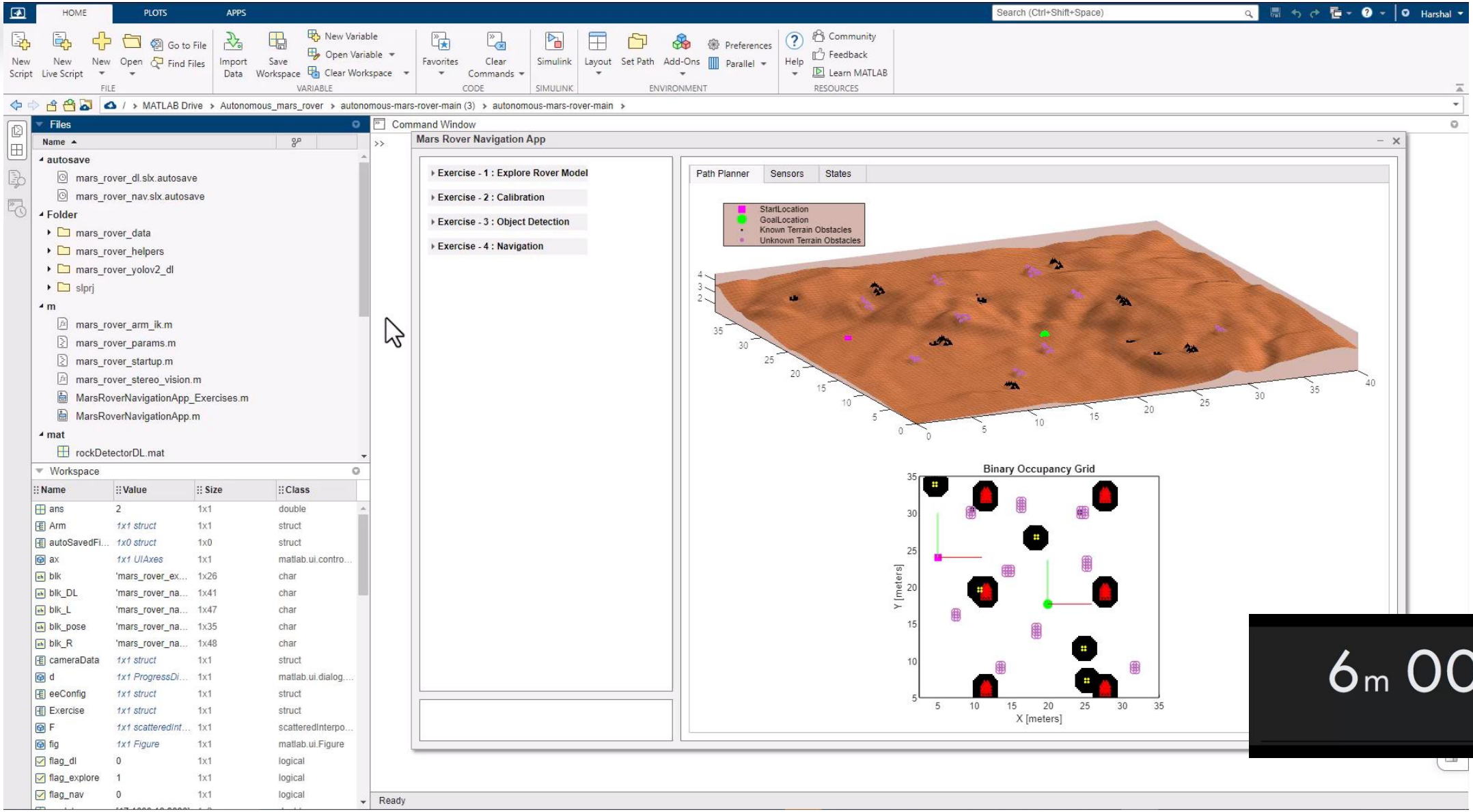
Specify Goal Position and plan a path

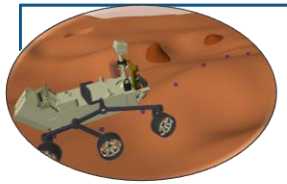
1. Either pick a point for goal location or modify them in the edit fields
2. Hit “Update Map” to observe the results in the map area
3. Hit “Plan Path” to visualize the planned path on the Occupancy map.
4. Then press “Simulate” to get your rover moving.



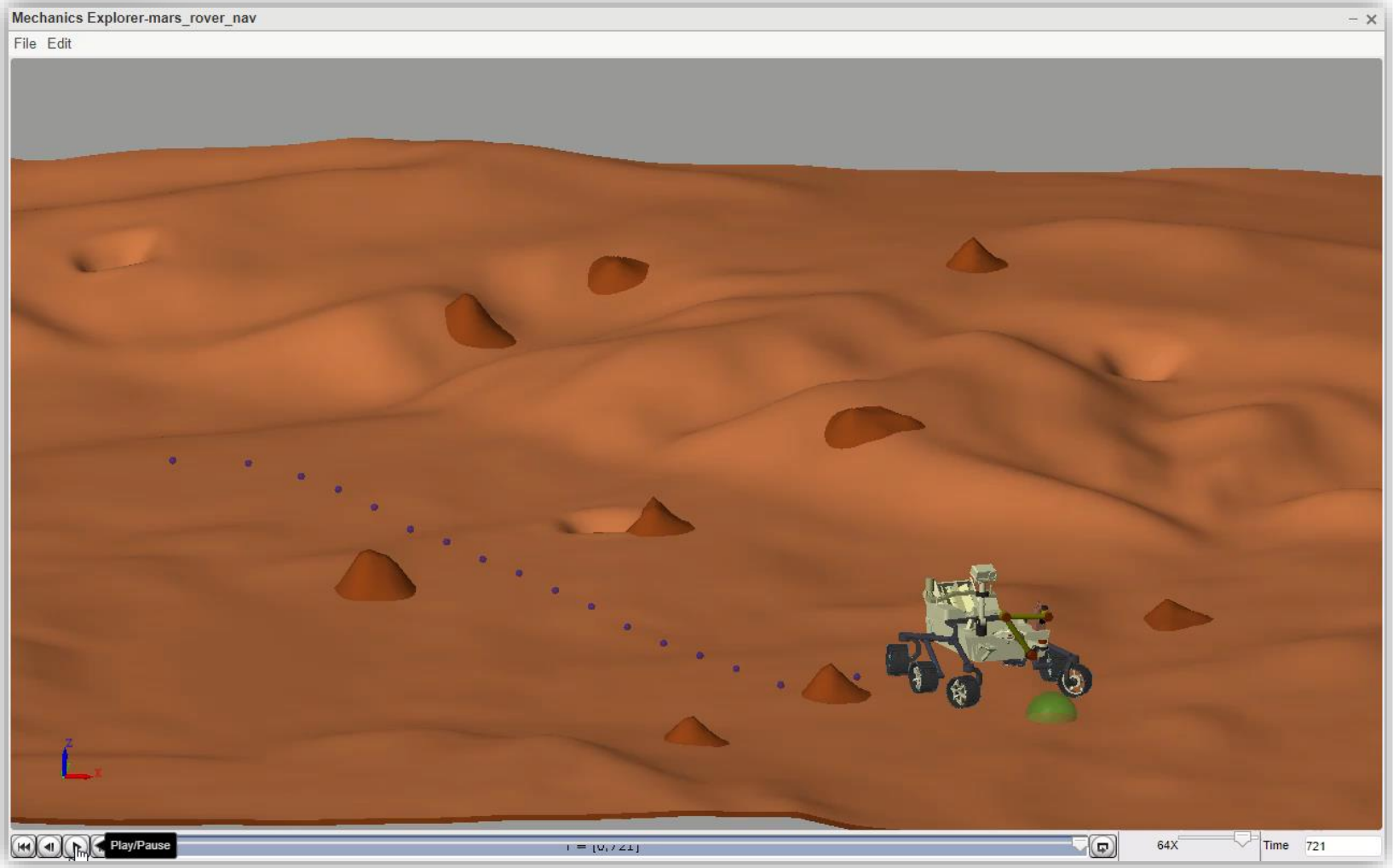
Please take 6 minutes to complete exercise 4

Need a hint? A video showing the solution will play here.



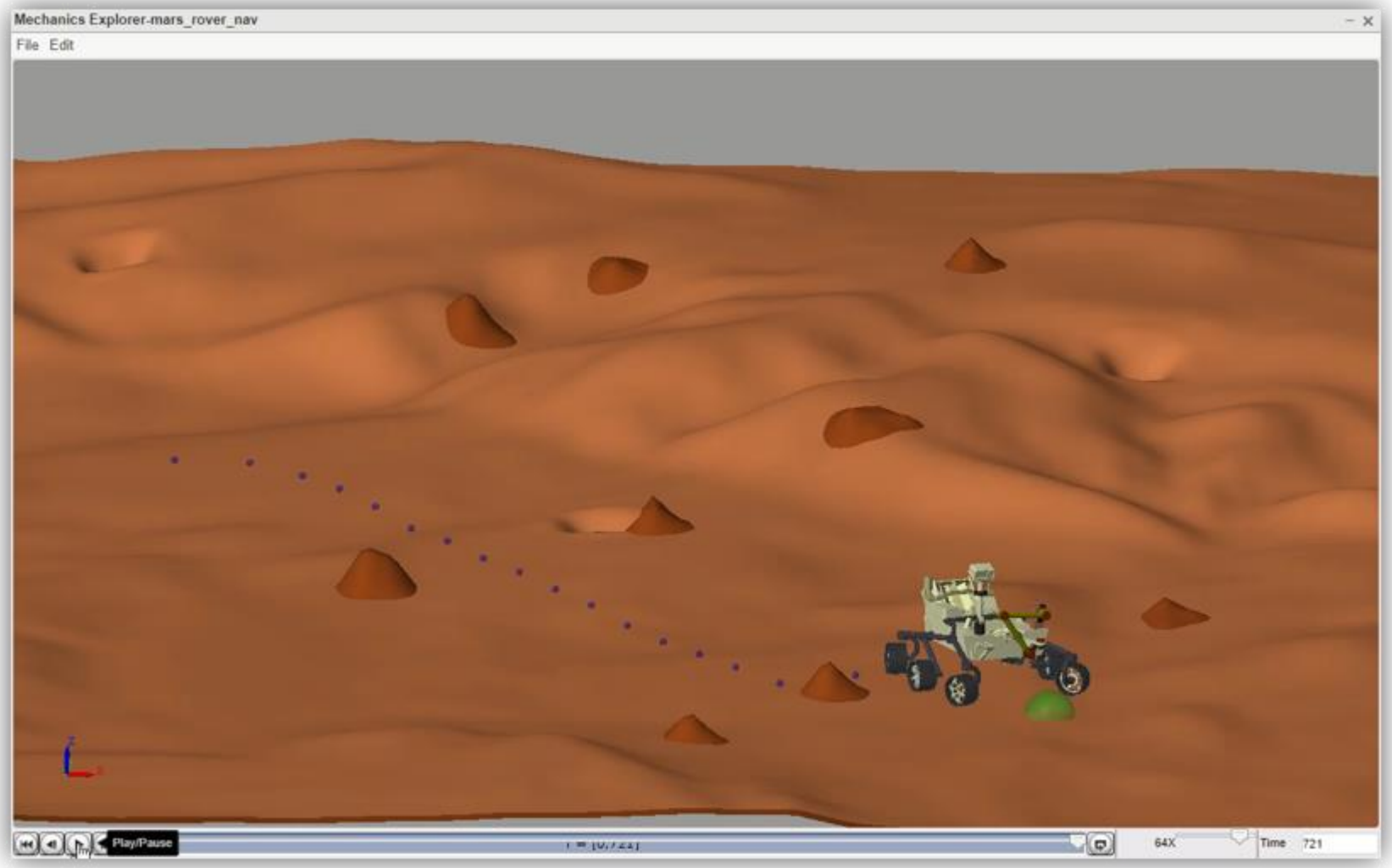


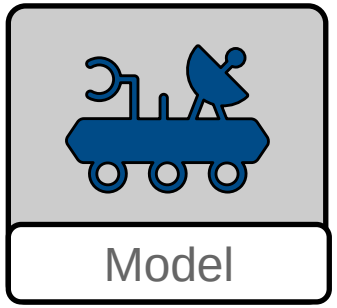
How did your rover perform?

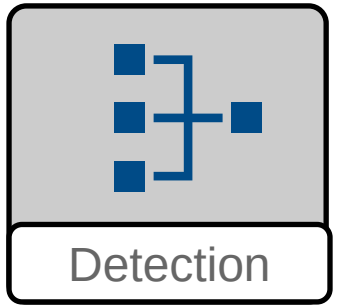


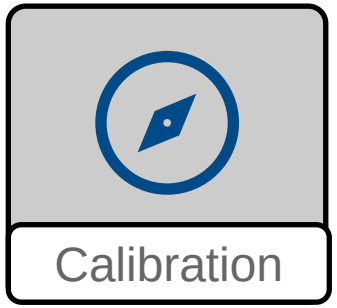
- What objects did you detect?
- Did you avoid obstacles?
- Did rover arm pick up a sample ?

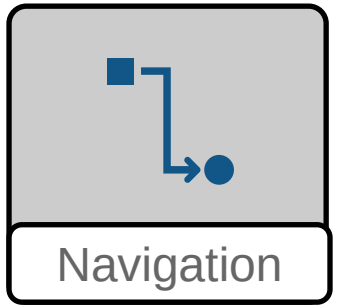
Congratulations ! You've just planned an effective mission using simulation.



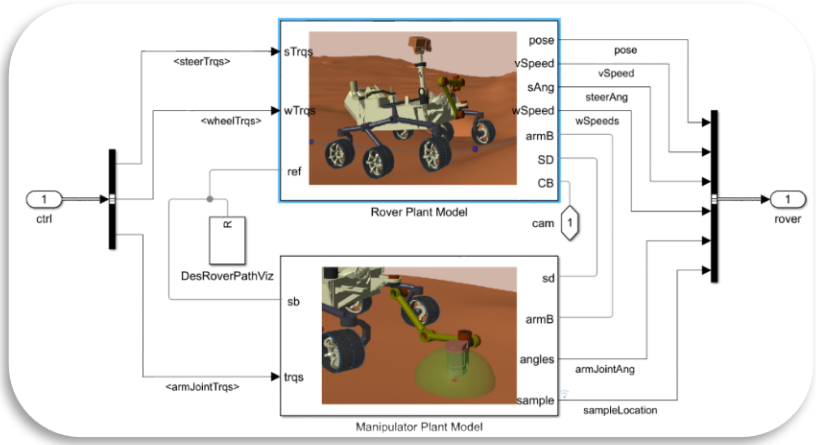




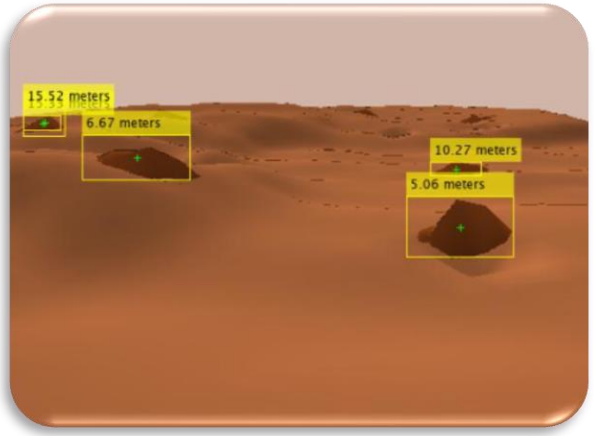




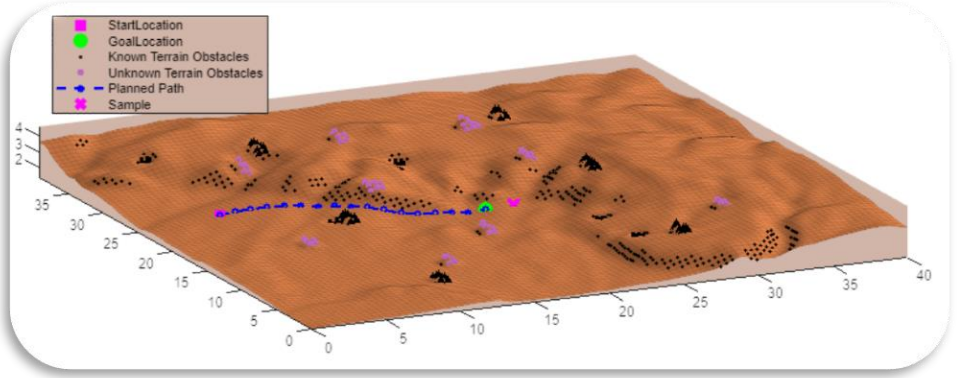
Key takeaways



Physical Modeling (Rover Model)



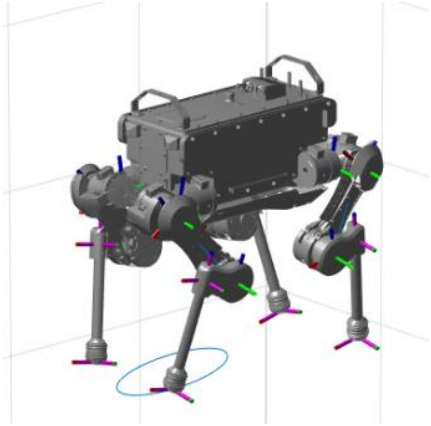
Deep learning for object detection



Path planning in autonomous systems

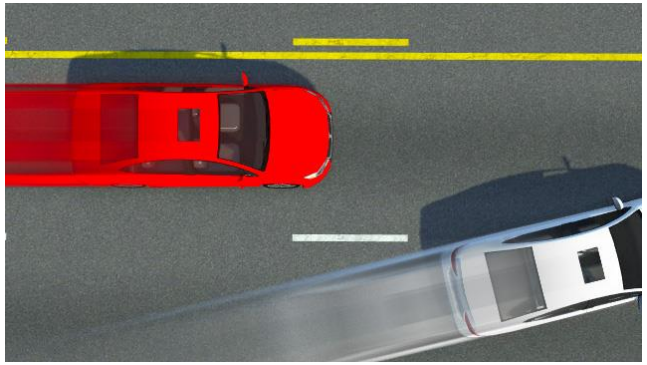
Simulation plays a key role in development and testing of autonomous systems

Design complex physical systems



Develop and test AI algorithms

Validate huge number of scenarios



Take the Next Step

Learn more:

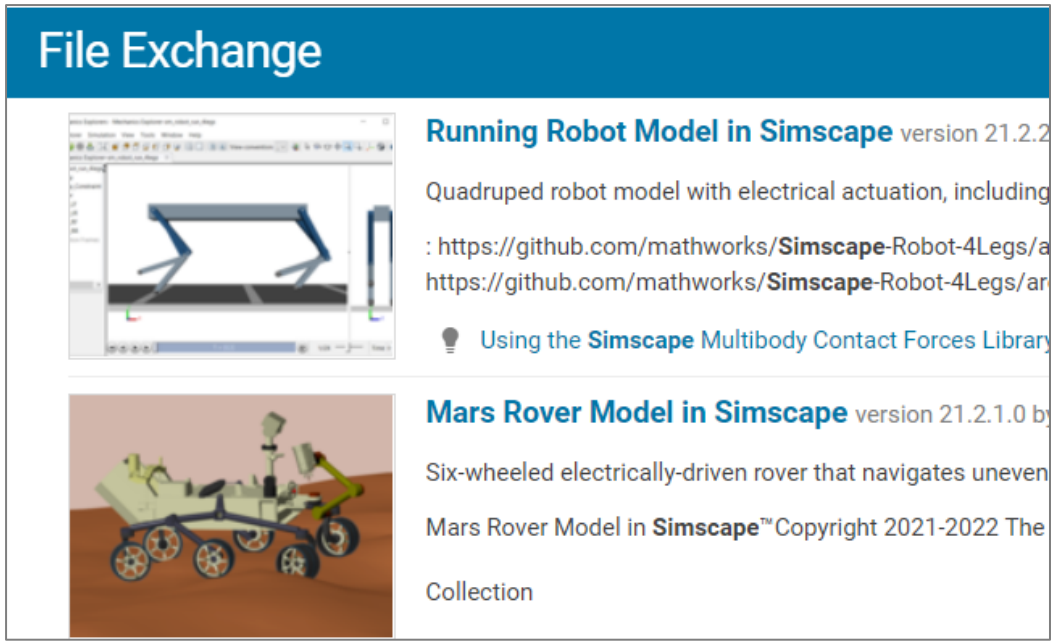
Robotics and Autonomous Systems
https://www.mathworks.com/solutions/robotics.html

Simscape Onramp
www.mathworks.com/learn/tutorials/simscape-onramp.html

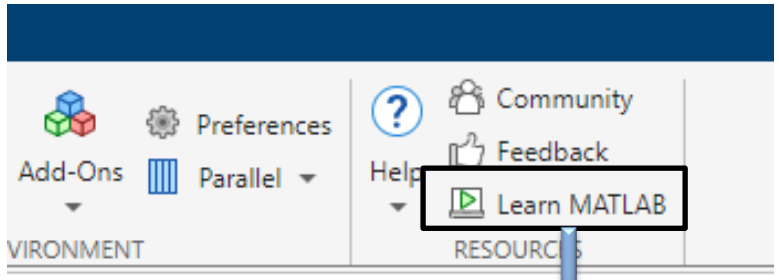
Deep Learning Onramp
https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning

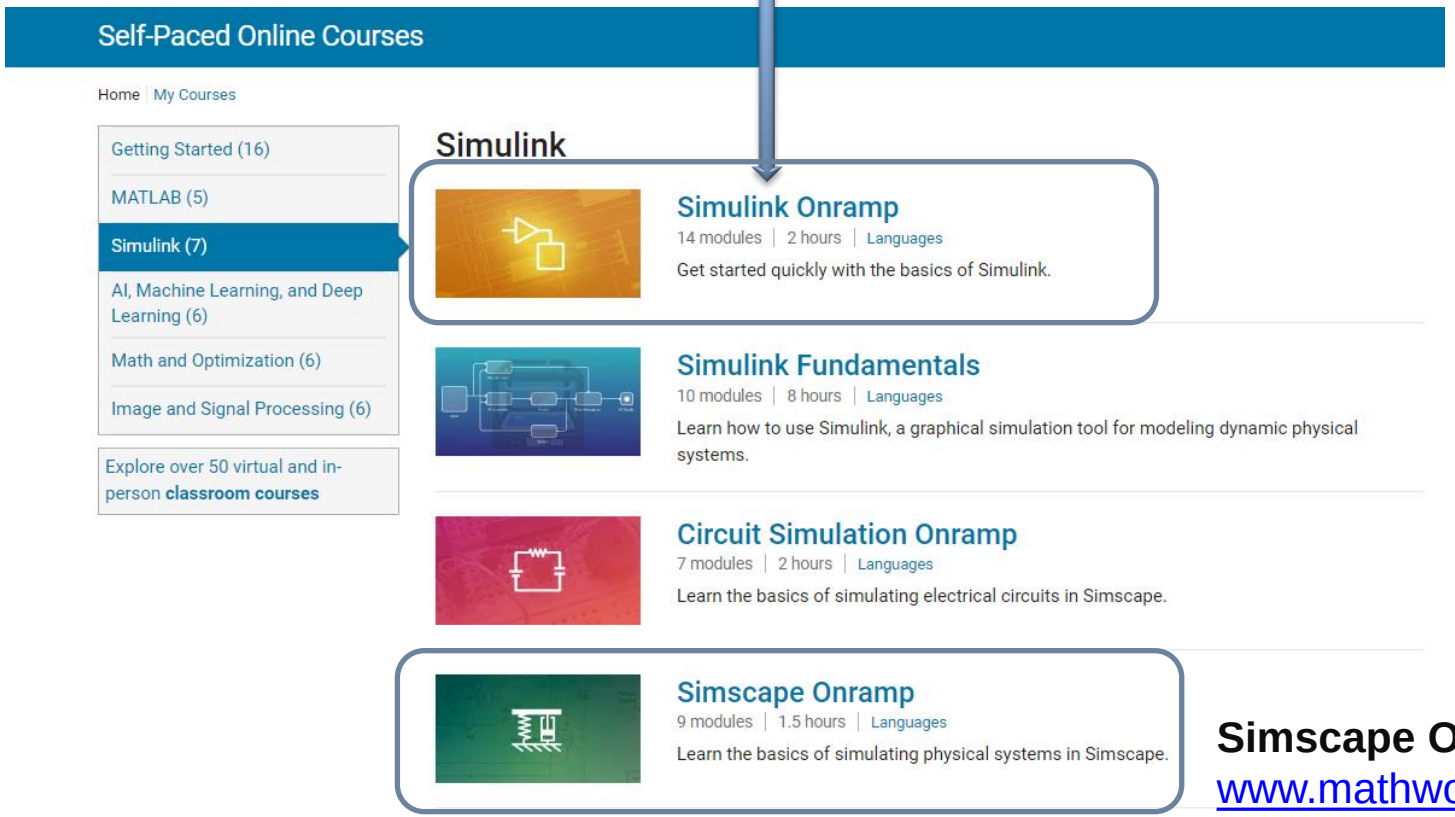
Deep Learning for Image Analysis
https://www.mathworks.com/solutions/deep-learning.html

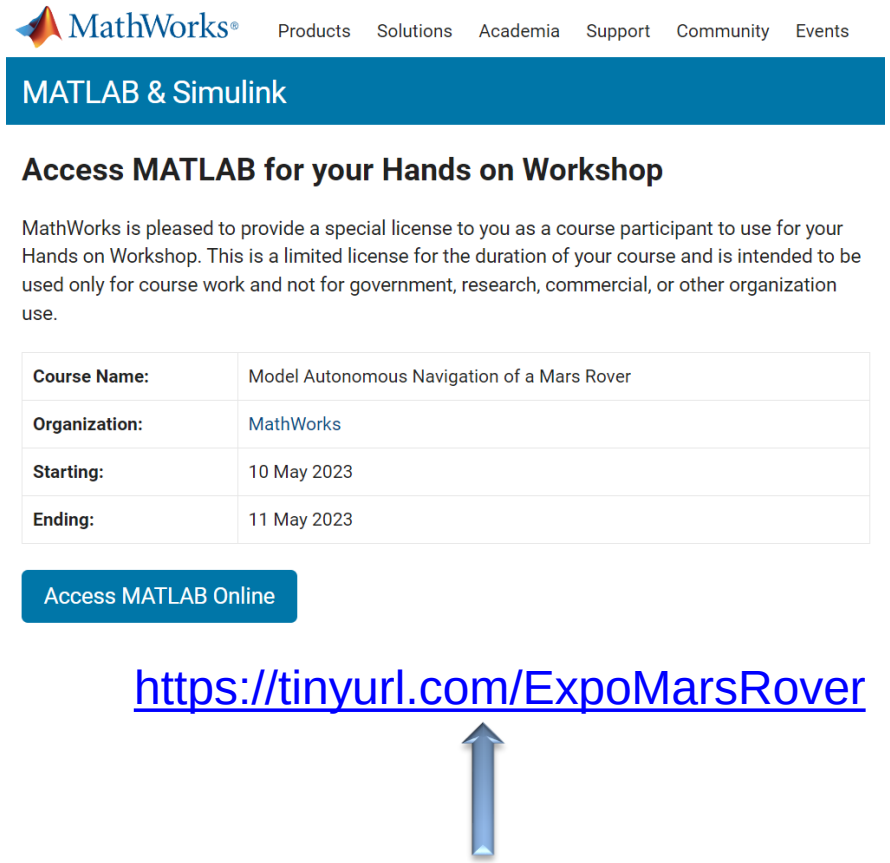
Find More Examples on File Exchange
https://www.mathworks.com/matlabcentral/fileexchange/?q=Simscape&sort=relevancy



Take the Next Step







https://tinyurl.com/ExpoMarsRover

Access till May 26th

Simscape Onramp
www.mathworks.com/learn/tutorials/simscape-onramp.html

Shout out to the TA team!


Bonita Vormawor


Tharikaa R Kumar


Timothy Kyung


Aditya Sakhare


Brendan Amorin


Corey Taylor


Divya Yerraguntla


Gabija Marsalkaite


Vikram Venkatesh

Connect with us on social media

Harshal Upadhyay
https://www.linkedin.com/in/harshal-upadhyay

Kritika Ramani
https://www.linkedin.com/in/kritika-ramani

#MATLABEXPO #ModelAutonomousMarsRover #Workshop

Connect with us on social media

Harshal Upadhyay
https://www.linkedin.com/in/harshal-upadhyay

Chinmayi Lanka
https://www.linkedin.com/in/chinmayilanka

#MATLABEXPO #ModelAutonomousMarsRover #Workshop

Connect with us on social media

Ameya Godbole
https://www.linkedin.com/in/ameya-rajendra-godbole

Chinmayi Lanka
https://www.linkedin.com/in/chinmayilanka

#MATLABEXPO #ModelAutonomousMarsRover #Workshop

MATLAB EXPO

Thank you