



Welcome to our workshop Model Autonomous Navigation of a Mars Rover



MathWorks 

@MathWorks

Share the EXPO experience
#MATLABEXPO

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





Model Autonomous Navigation of a Mars Rover

Harshal Upadhyay



Kritika Ramani



MATLAB EXPO

Model Autonomous Navigation of a Mars Rover

Harshal Upadhyay



Chinmayi Lanka



MATLAB EXPO

Model Autonomous Navigation of a Mars Rover

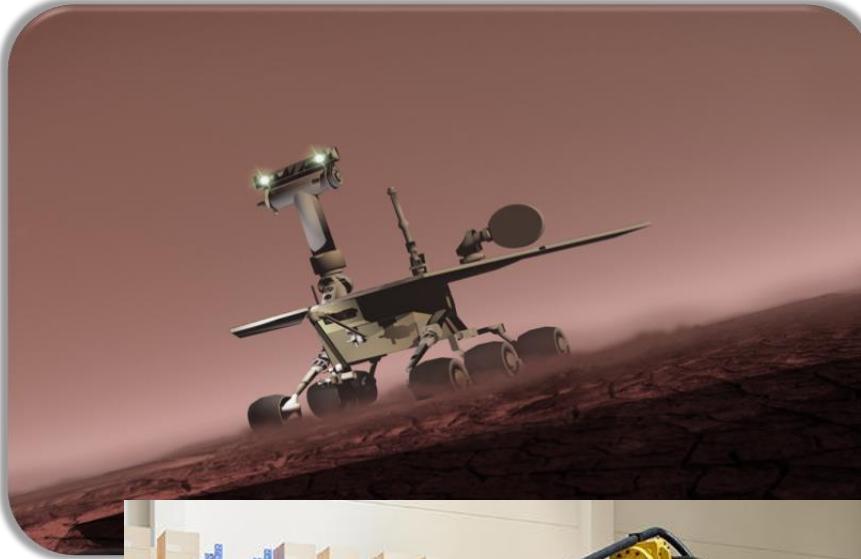
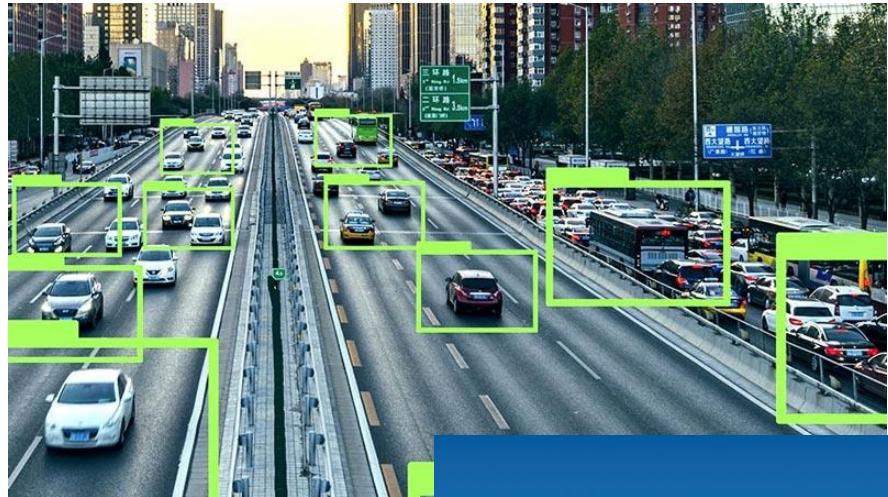
Chinmayi Lanka



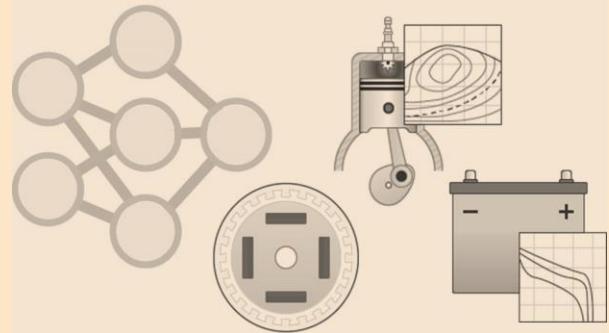
Ameya Rajendra Godbole



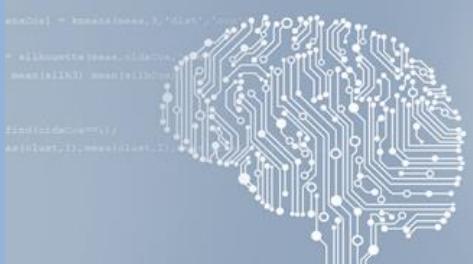
The rise of autonomous systems



Challenges



**System
Complexity**



**Decision
Making**



**Safety and
Reliability**

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

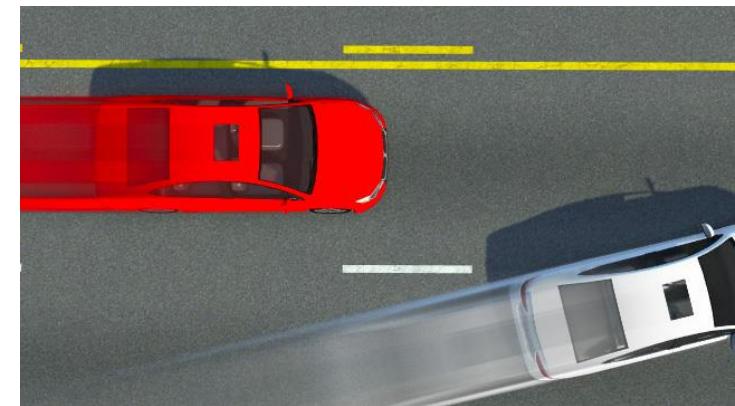
Expensive Hardware



Uncharted territories



Complex Scenarios

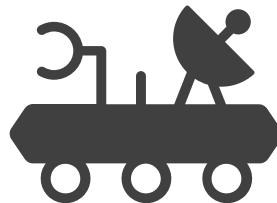


The problem we'll be tackling in this workshop



Simulate navigation of an autonomous rover on a virtual Mars environment

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



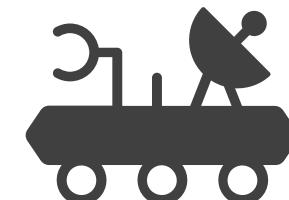
Use your laptop and a web browser to run exercises



Download instructions using the “Handouts” Tab

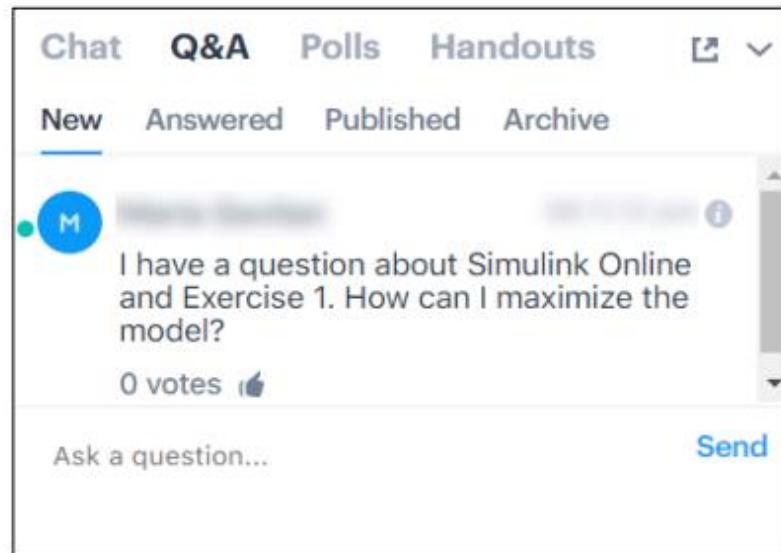


A group of TAs will assist with questions



How to get assistance during the workshop

BigMarker (use Q&A)



Today we will be using MATLAB Online



Products Solutions Academia Support Community Events

MATLAB & Simulink

Access MATLAB for your Hands on Workshop

MathWorks is pleased to provide a special license to you as a course participant to use for your Hands on Workshop. This is a limited license for the duration of your course and is intended to be used only for course work and not for government, research, commercial, or other organization use.

Course Name:	Model Autonomous Navigation of a Mars Rover
Organization:	MathWorks
Starting:	10 May 2023
Ending:	11 May 2023

Access MATLAB Online

<https://tinyurl.com/ExpoMarsRover>

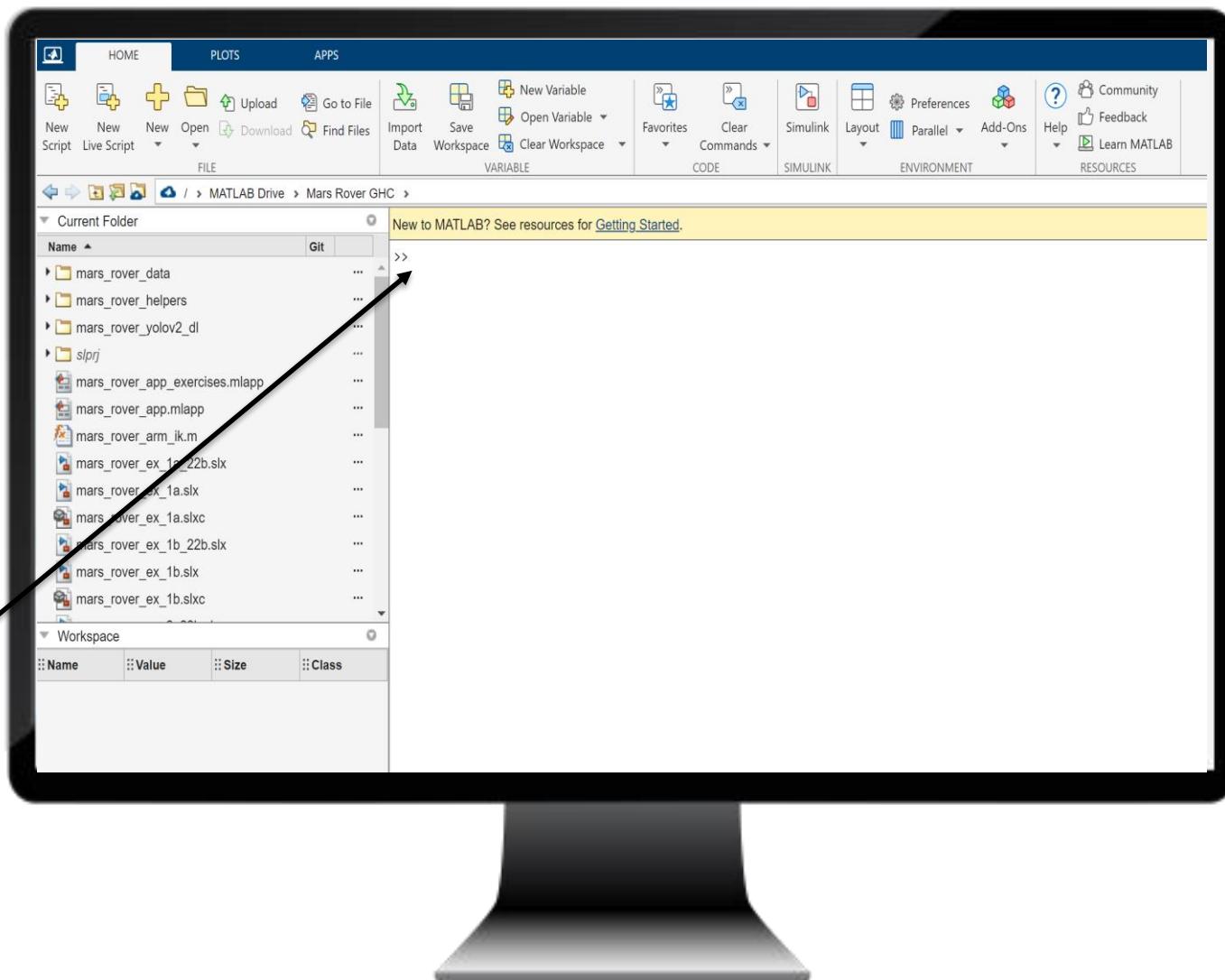
Open MATLAB Online

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

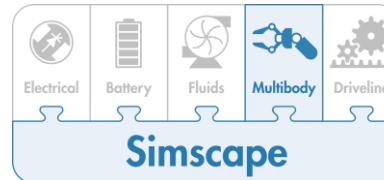
The screenshot shows a web browser displaying the MathWorks website. The URL in the address bar is <https://tinyurl.com/ExpoMarsRover>. The page title is "MATLAB & Simulink". A sub-header reads "Access MATLAB for your Hands on Workshop". Below this, a message from MathWorks states: "MathWorks is pleased to provide a special license to you as a course participant to use for your Hands on Workshop. This is a limited license for the duration of your course and is intended to be used only for course work and not for government, research, commercial, or other organization use." To the right, there is a login form with fields for "Email" (containing a placeholder "Email") and a "Next" button. At the bottom right of the page, there is a timer counting down from 5 minutes and 0 seconds.

MATLAB Online link

<https://tinyurl.com/ExpoMarsRover>

>> mars_rover_startup

Products used in this workshop



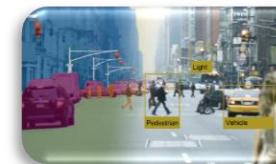
Simscape™

Today we will be using **Simulink Online**

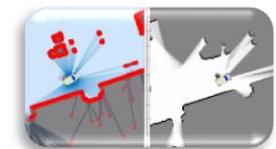
The screenshot shows the Simulink Online landing page. At the top, there's a navigation bar with links for "Overview", "Specifications and Limitations", "System Requirements", "Trial software", and "Contact sales". Below the navigation bar, the main heading is "Simulink Online" with the subtext "Use Simulink through your web browser". A prominent blue button says "Start using Simulink Online". Below this, there's a note: "Sign in to MATLAB Online to use Simulink or open an existing model. Simulink Online is available with select licenses. [Check your eligibility](#)". At the bottom of the page, there are three callout boxes: one with a calendar icon for "Use Simulink with no downloads or installations", one with two people icons for "Collaborate through online sharing", and one with a cloud icon for "Store, manage, and access your files anywhere".



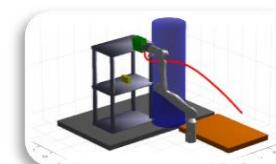
Deep Learning Toolbox™



Computer Vision Toolbox™

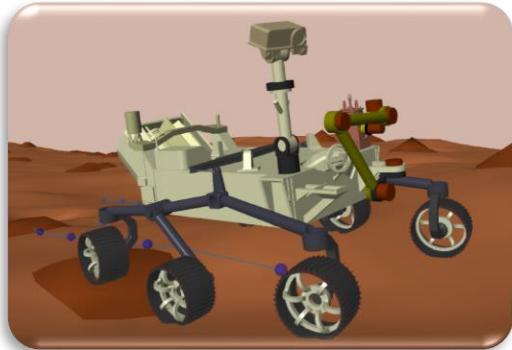


Navigation Toolbox™

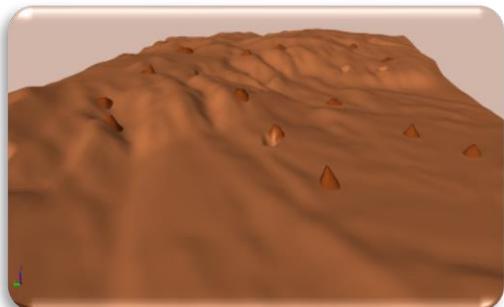


Robotics System Toolbox™

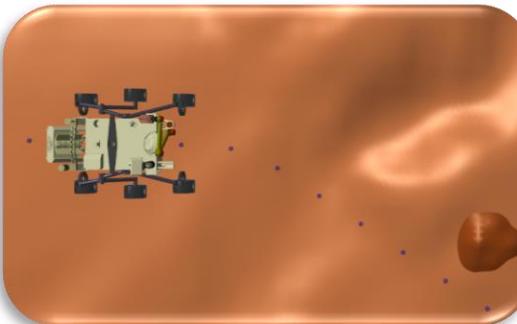
Key capabilities



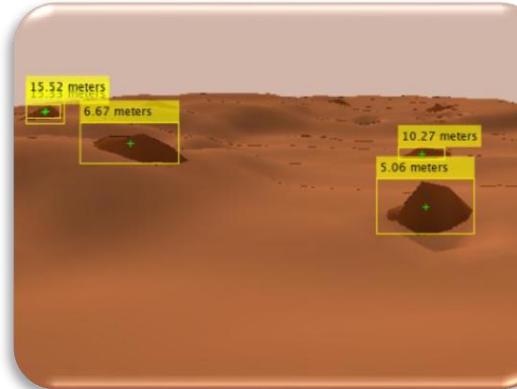
Model a rover



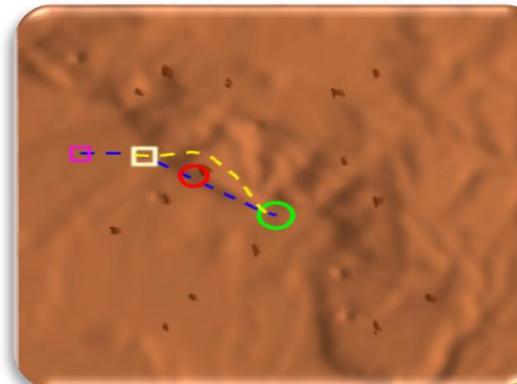
Virtual terrain



Position estimation

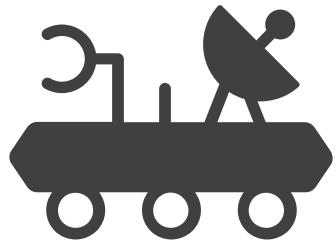


Hazard detection

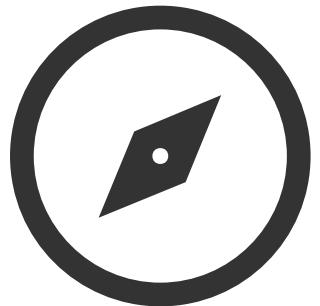


Path Planning

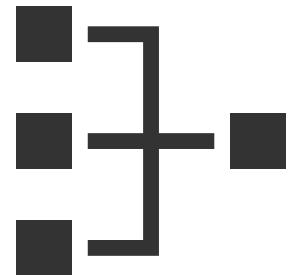
Our workshop today is split into multiple tasks



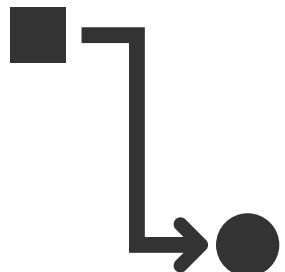
Who am I?
Rover Model



Where am I?
Calibration

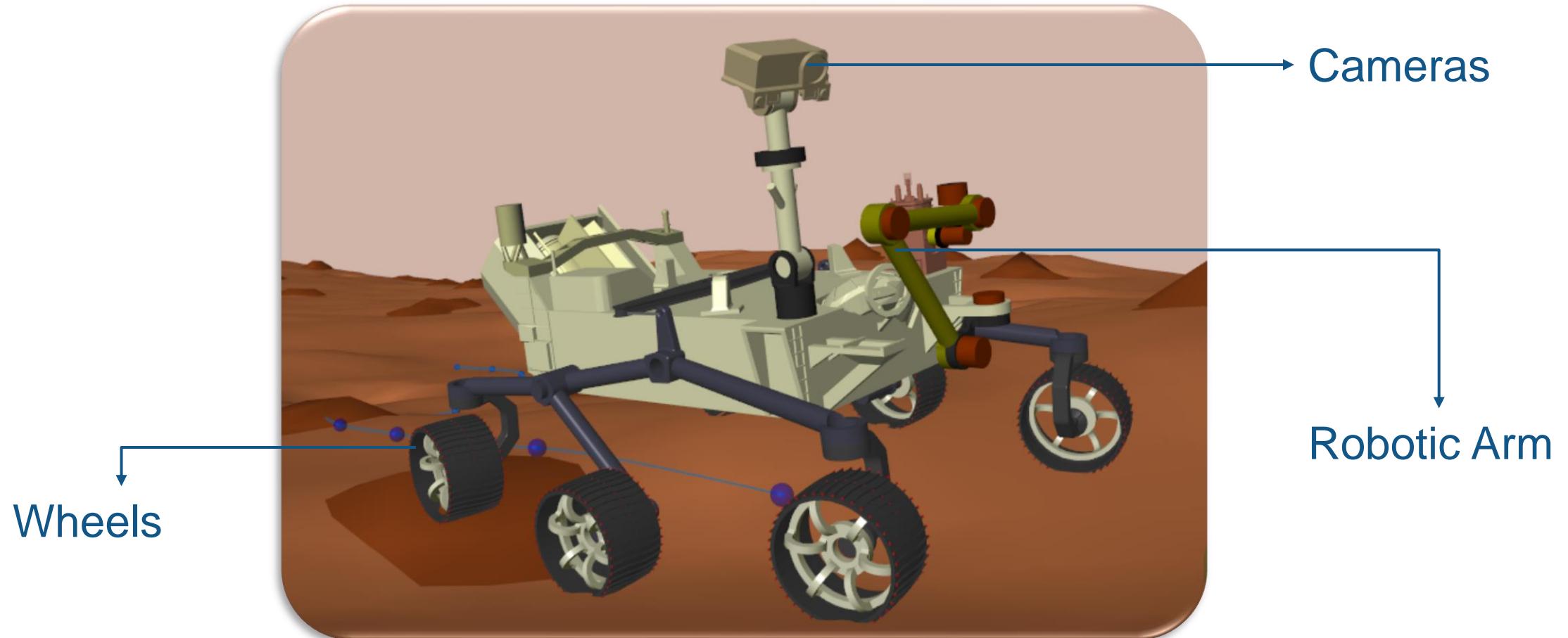


What do I see?
Object Detection

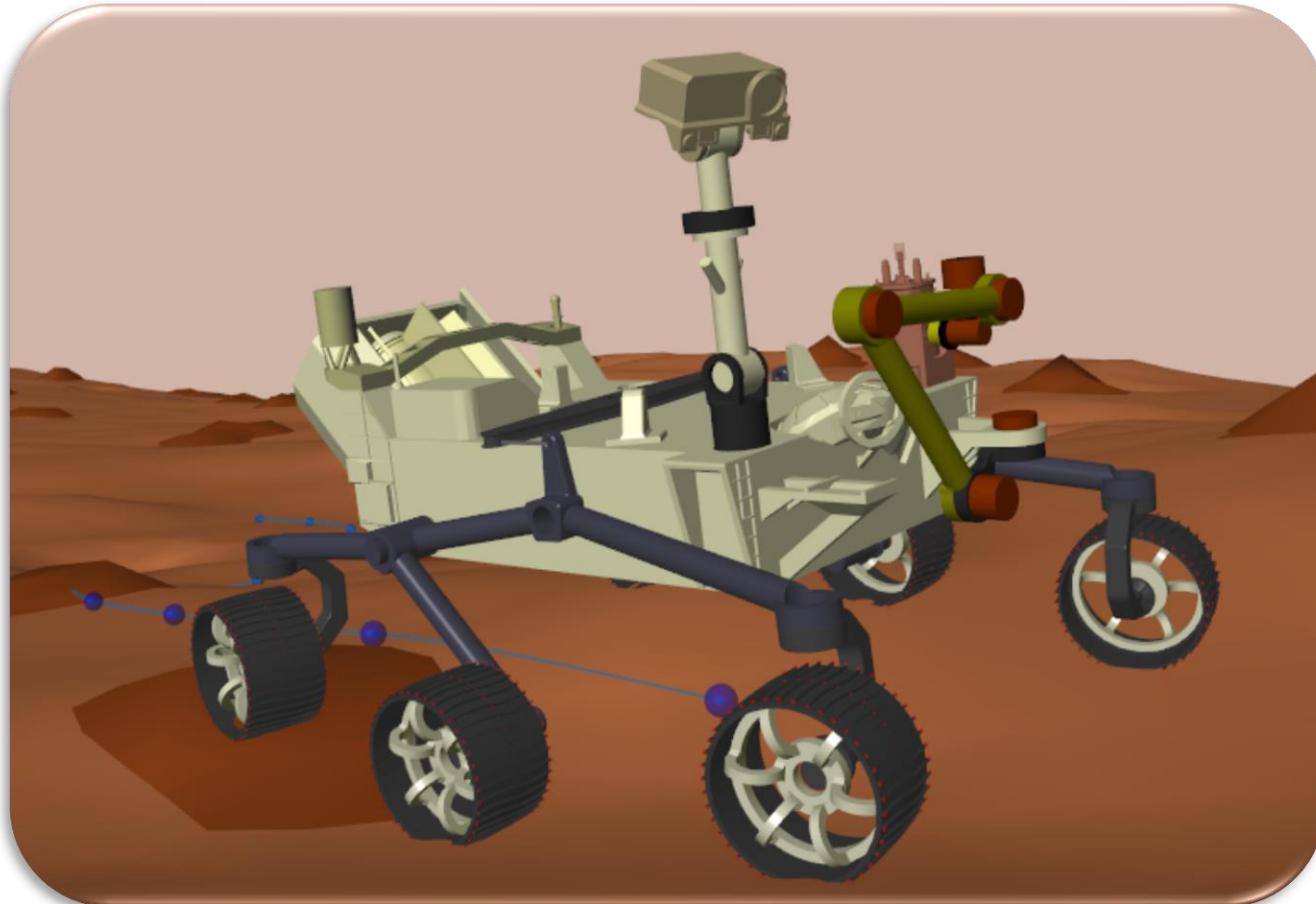


Where can I go?
Navigation

Multiple components are used to construct a physical model



Multiple components are used to construct a physical model



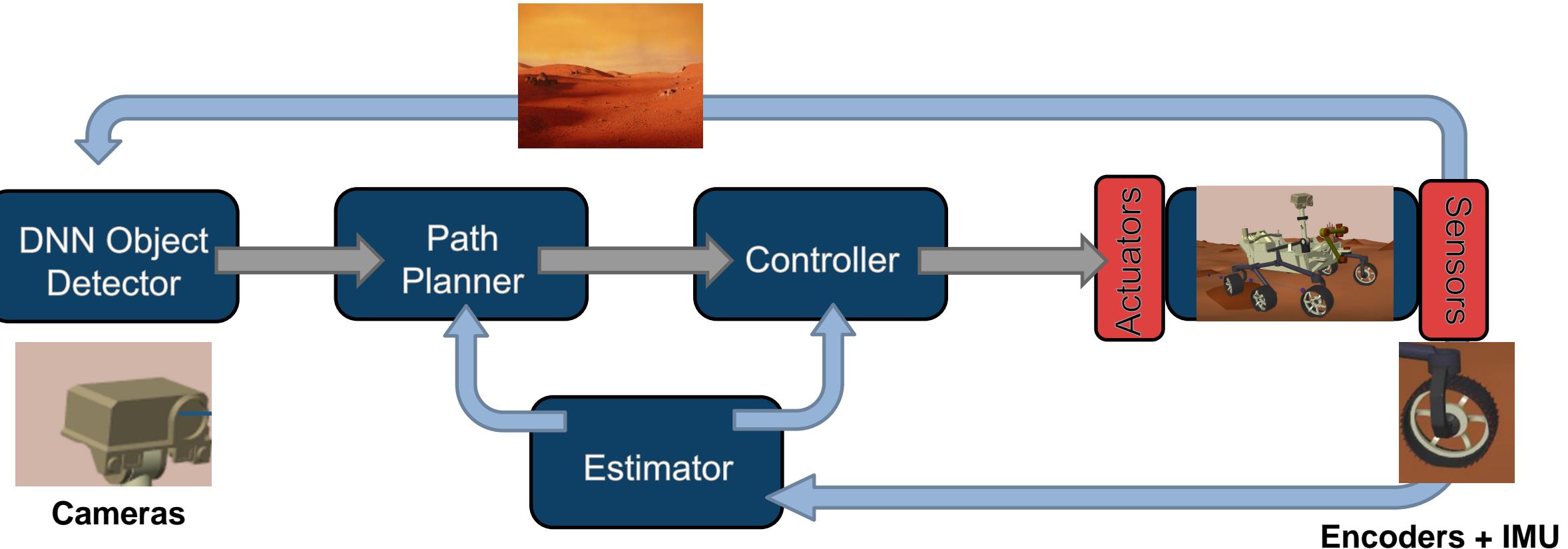
Chassis, Steering

Ground contact forces

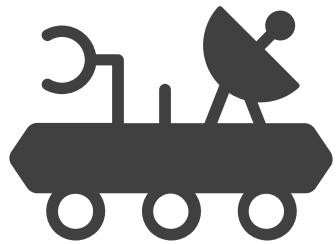
Max velocity (~0.1 mph)

Mapping Environment

Components of an autonomous rover



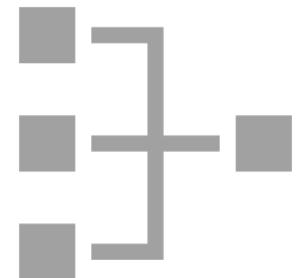
Our workshop today is split into multiple tasks



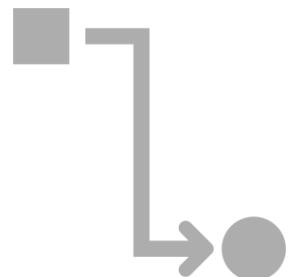
Who am I?
Rover Model



Where am I?
Calibration

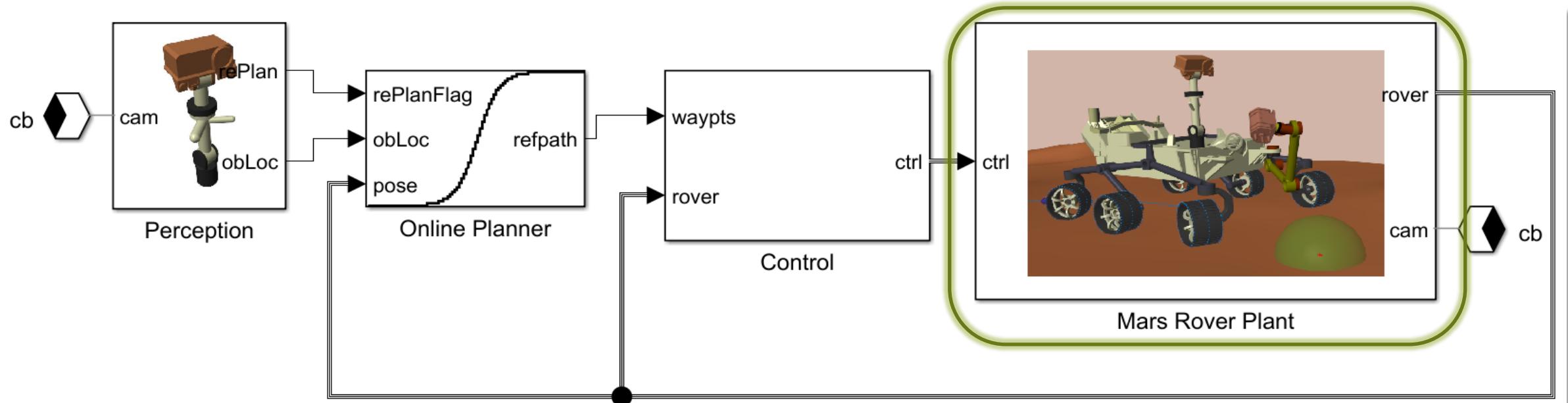
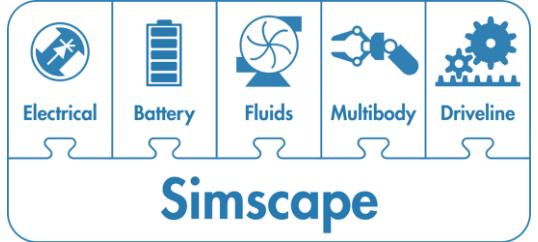


What do I see?
Object Detection



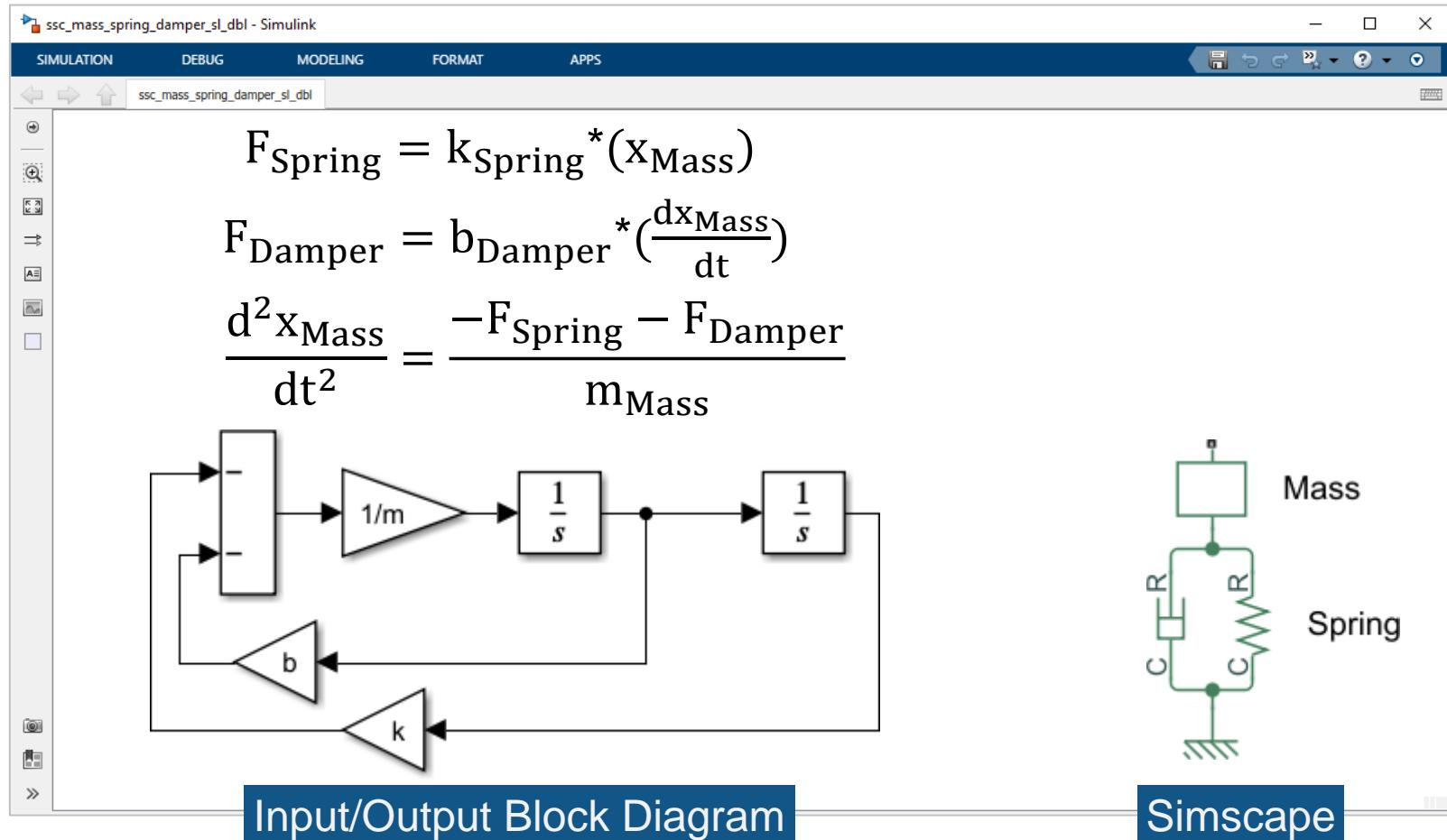
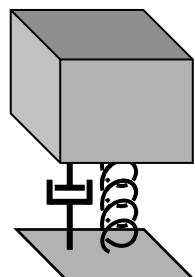
Where can I go?
Navigation

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

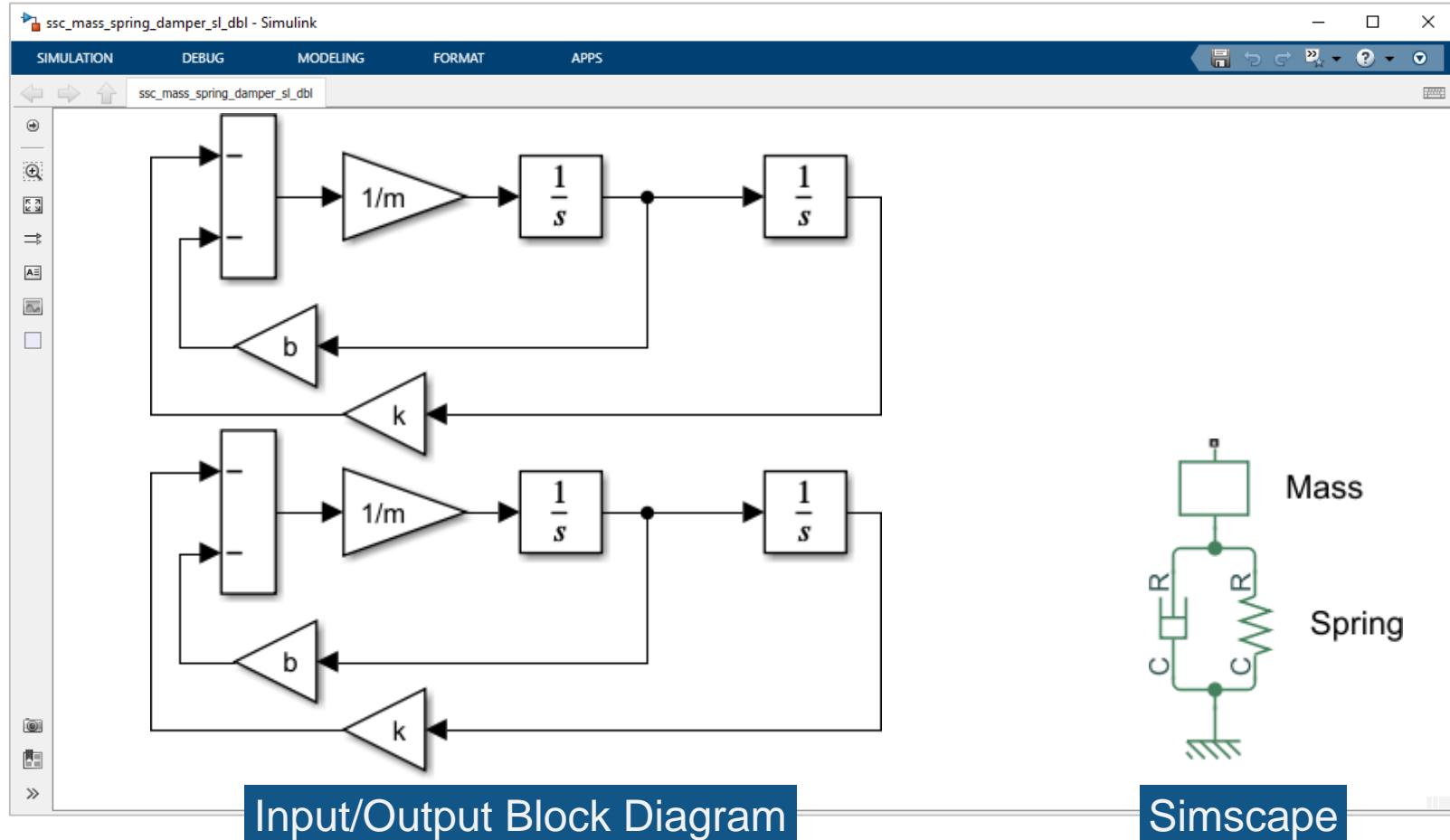
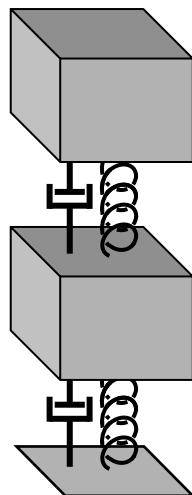


Autonomous Navigation of a Mars Rover

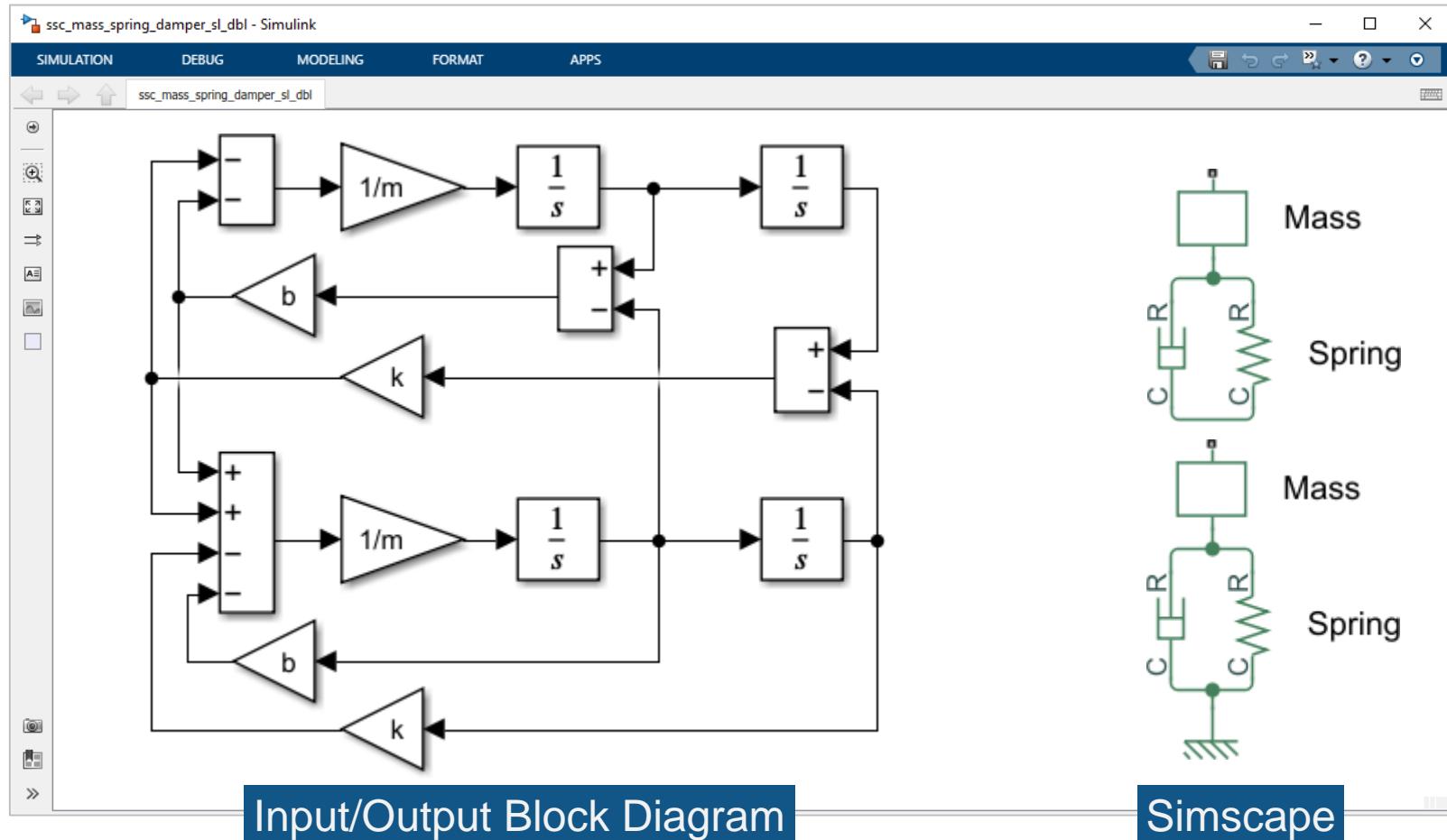
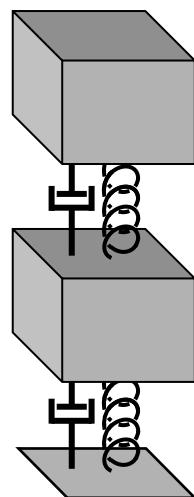
Simscape makes modeling physical systems easy



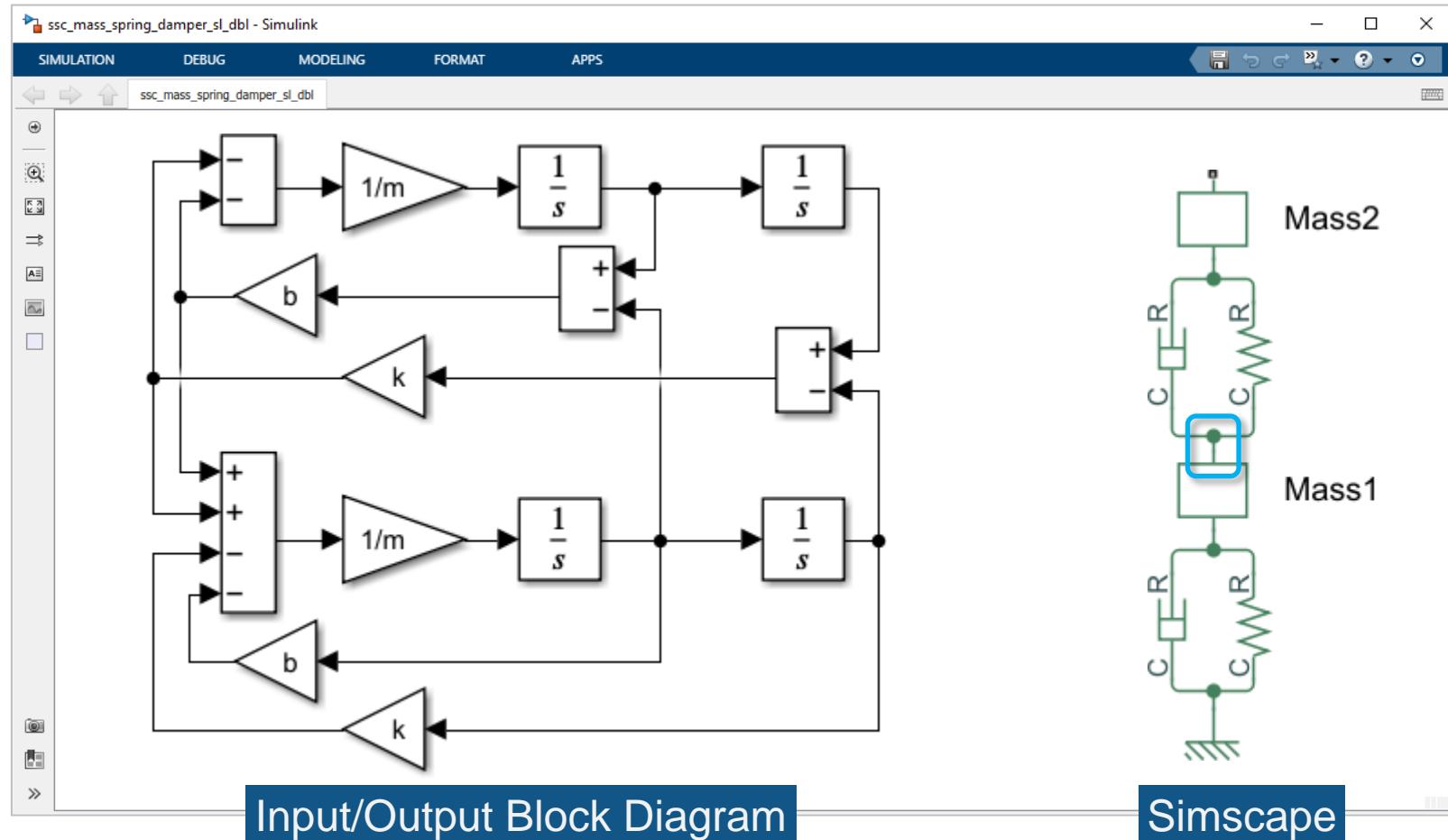
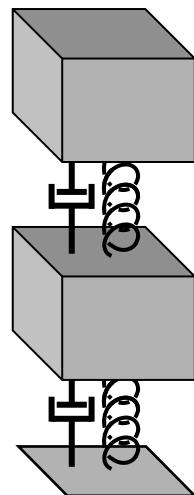
Simscape makes modeling physical systems easy



Simscape makes modeling physical systems easy

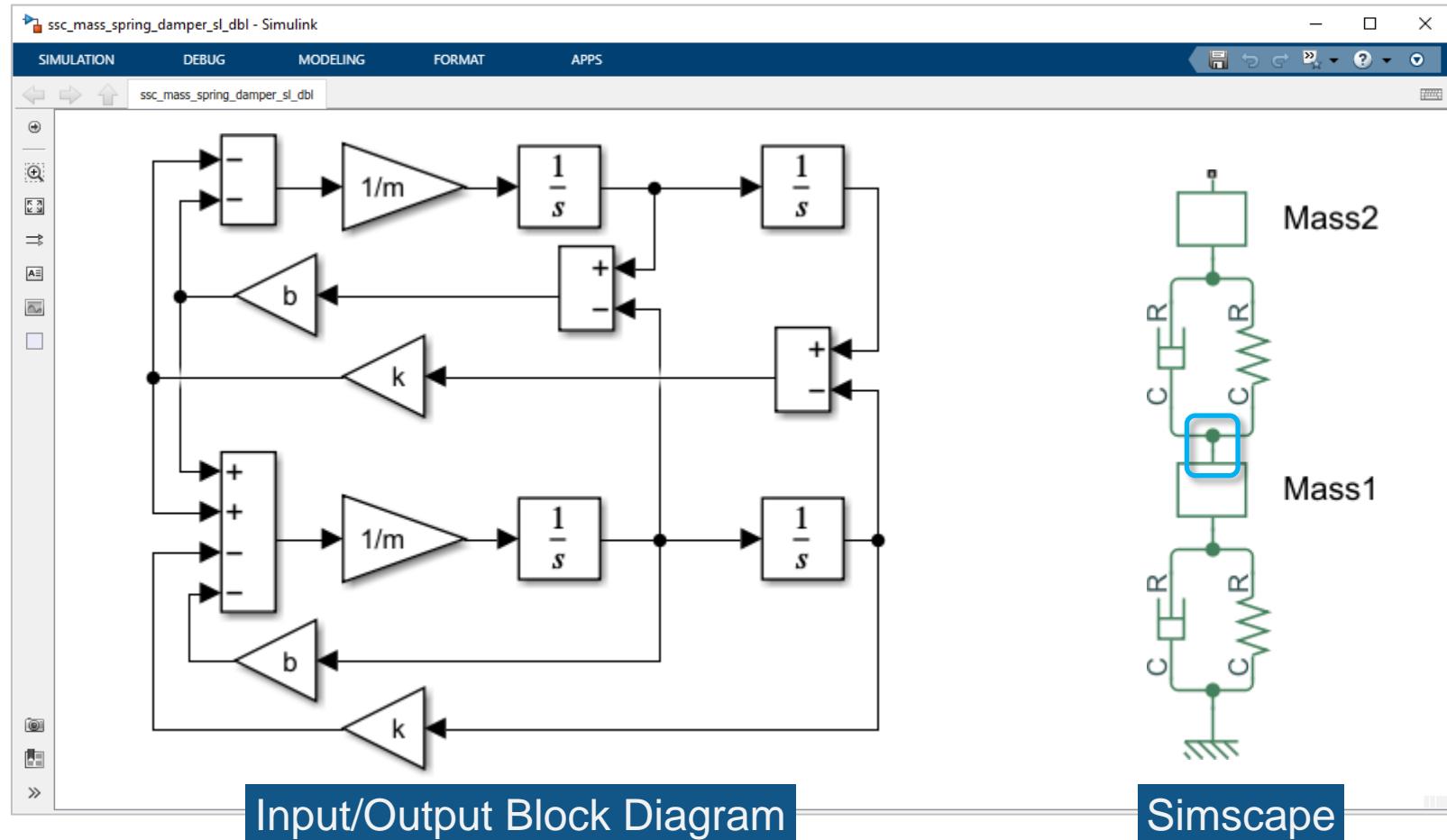


Simscape makes modeling physical systems easy



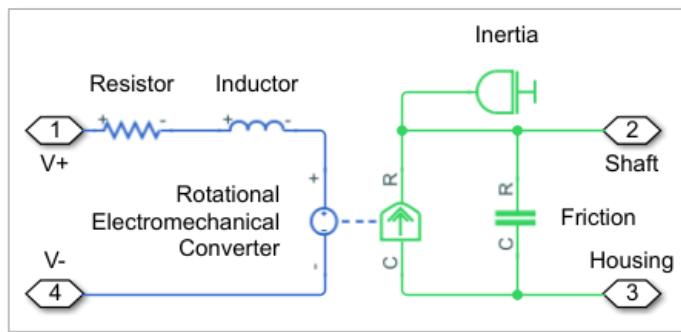
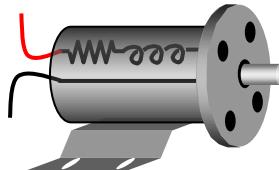
Simscape makes modeling physical systems easy

- 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

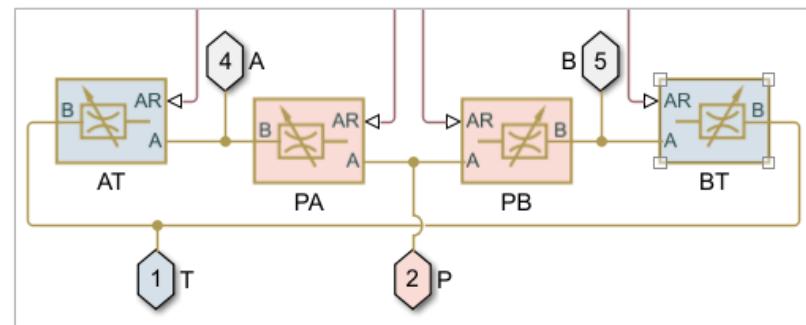
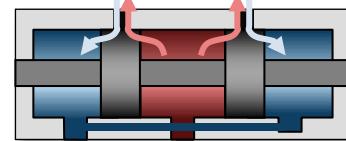


Modeling method applies to many domains

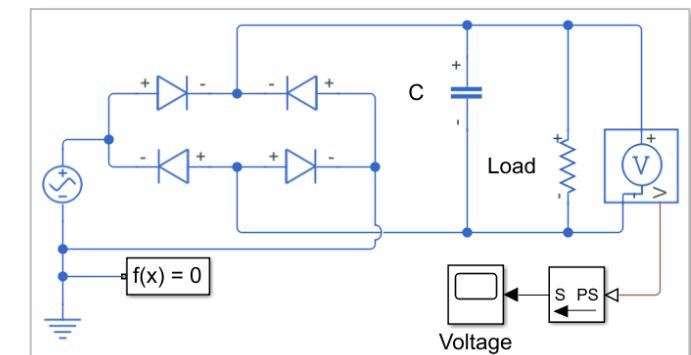
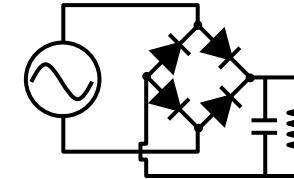
Electromechanical Systems such as motors



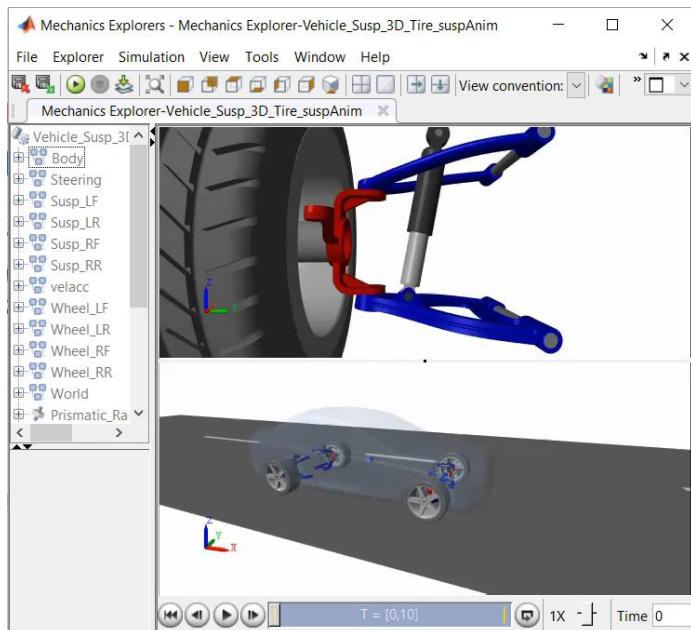
Fluid Systems such as valves and pumps



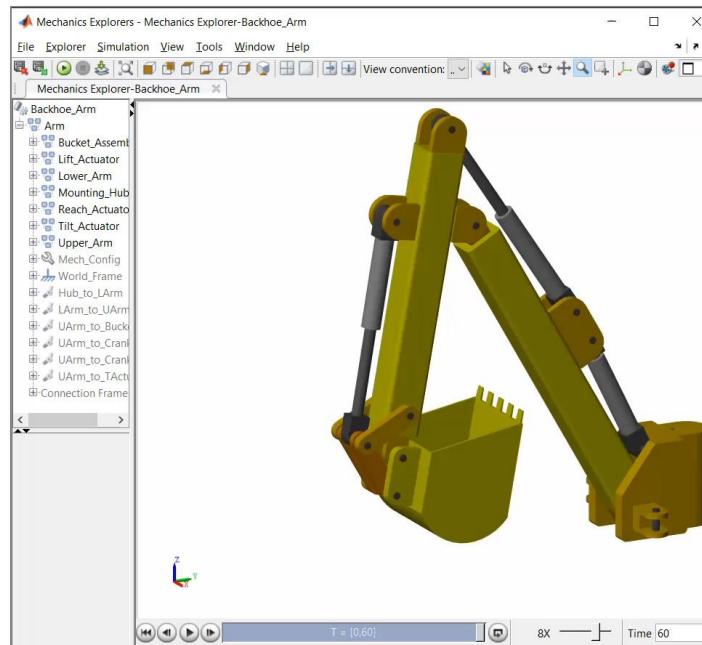
Electronic Systems such as power converters



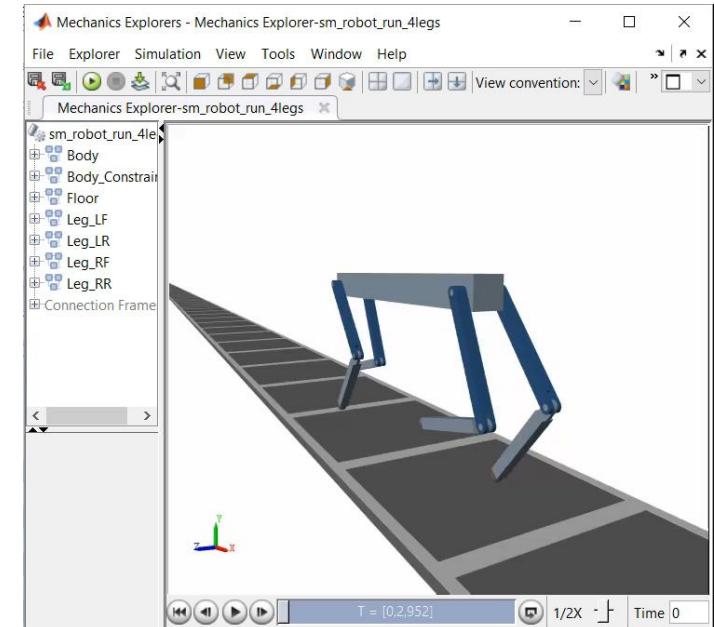
Modeling method scales to complex systems



Automotive Systems
such as vehicle dynamics
and powertrain

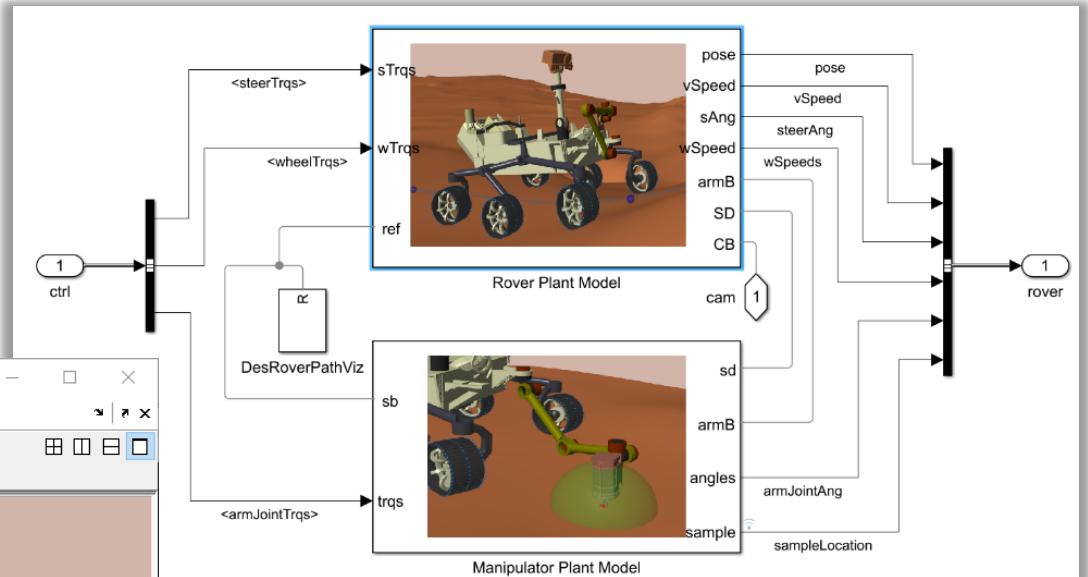
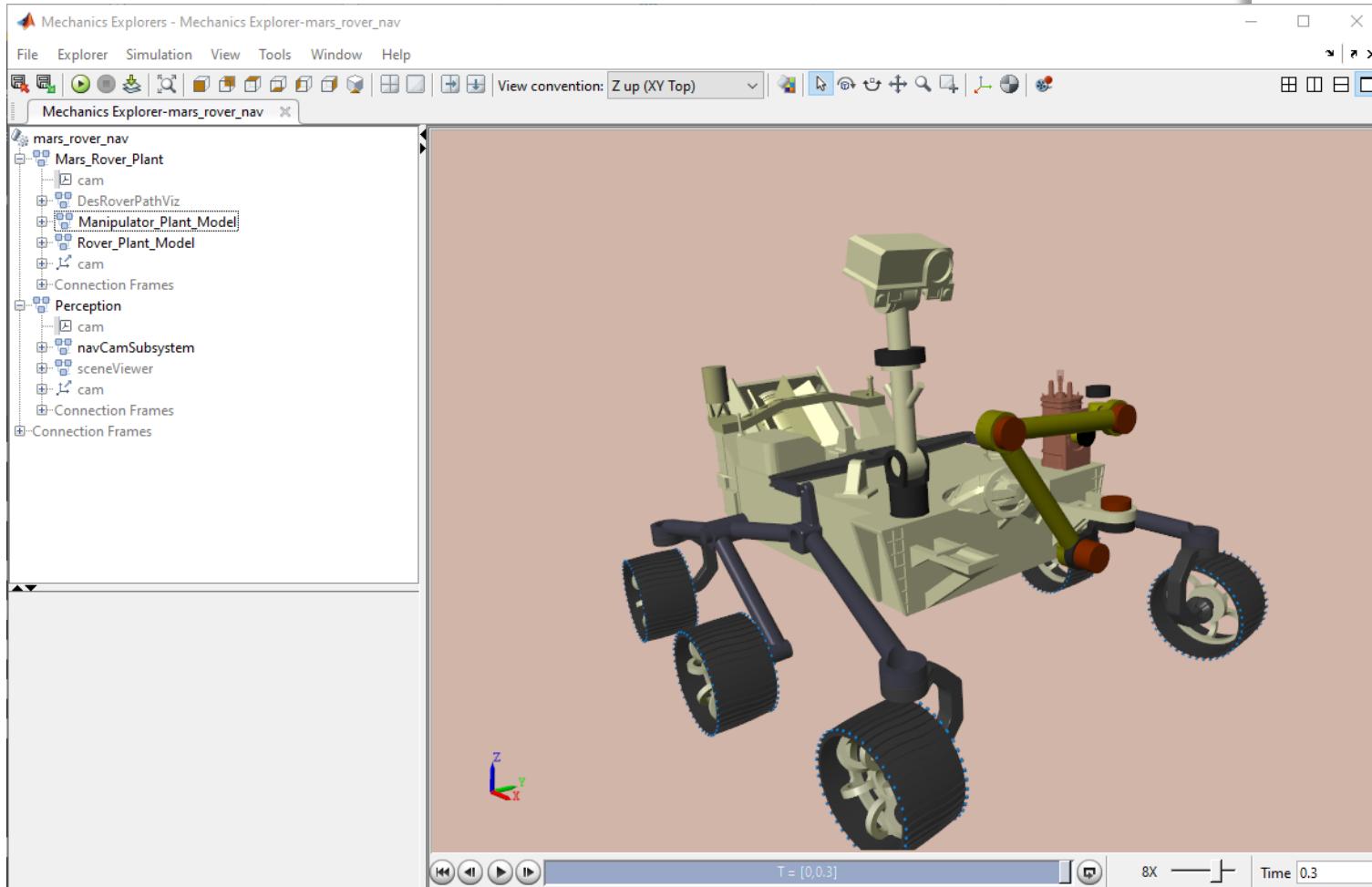


Industrial Equipment
such as backhoes



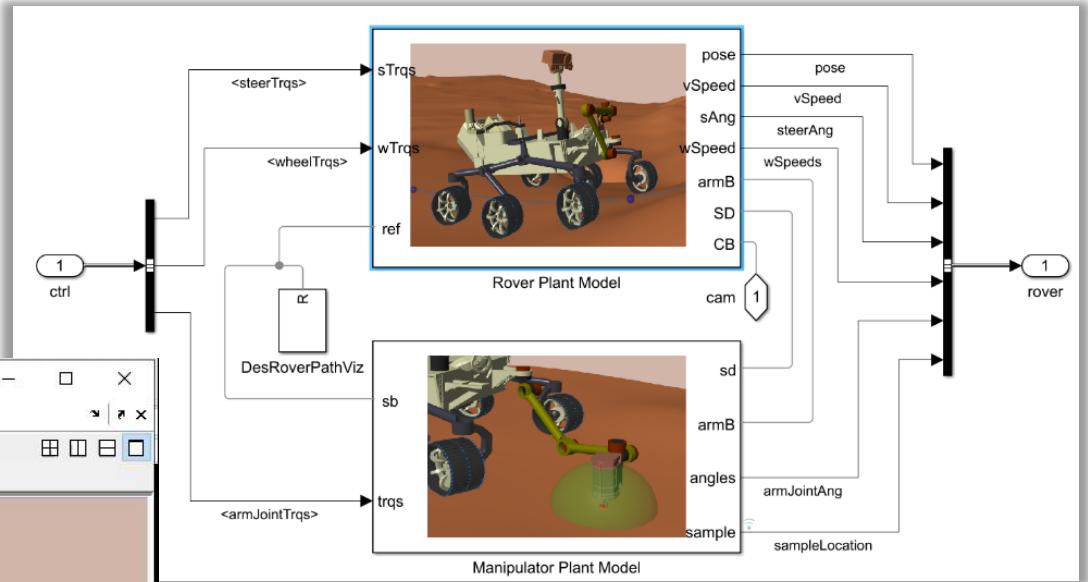
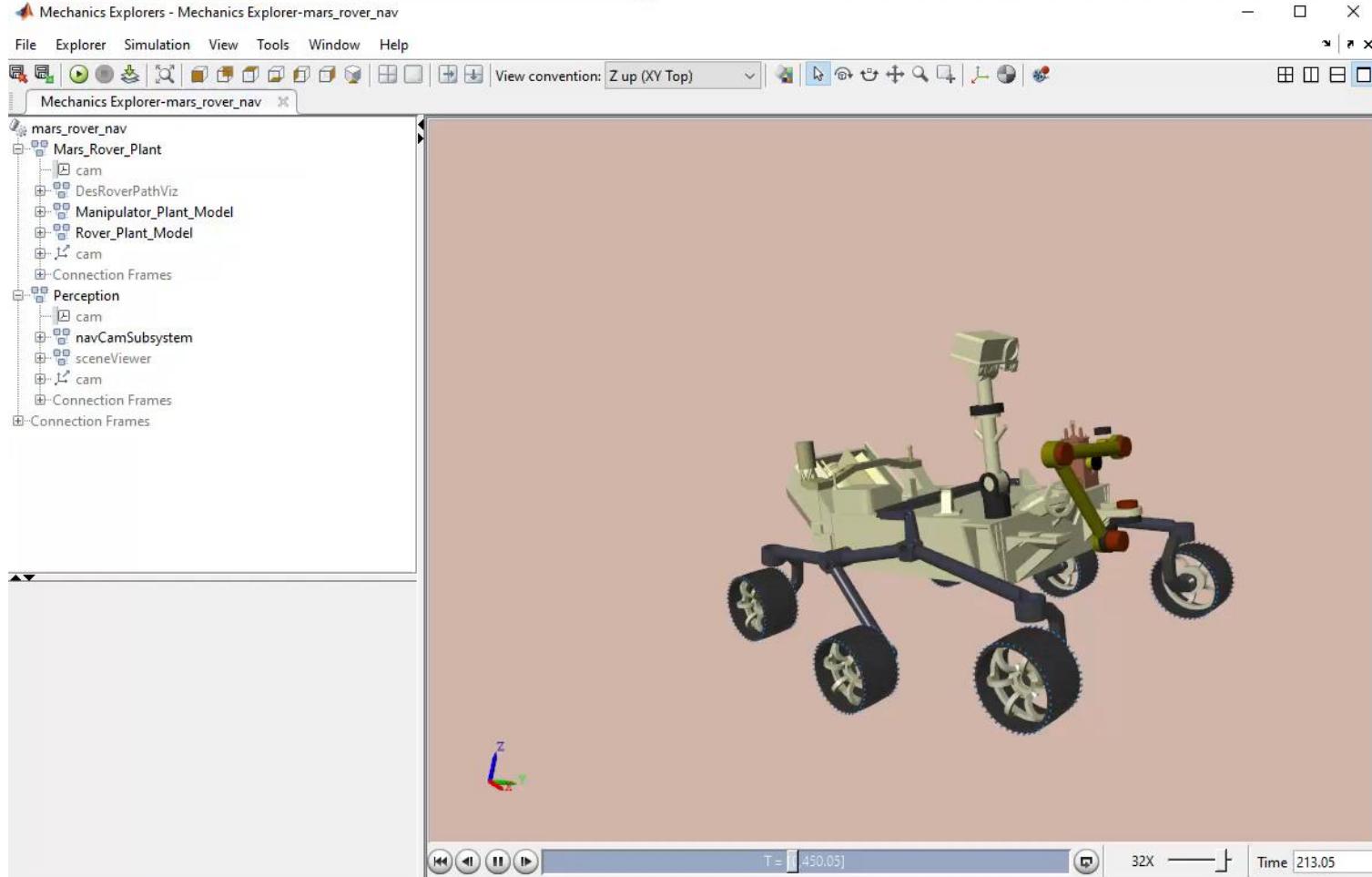
Robotic Systems
such as mobile robotics
and manipulators

Modeling the Rover



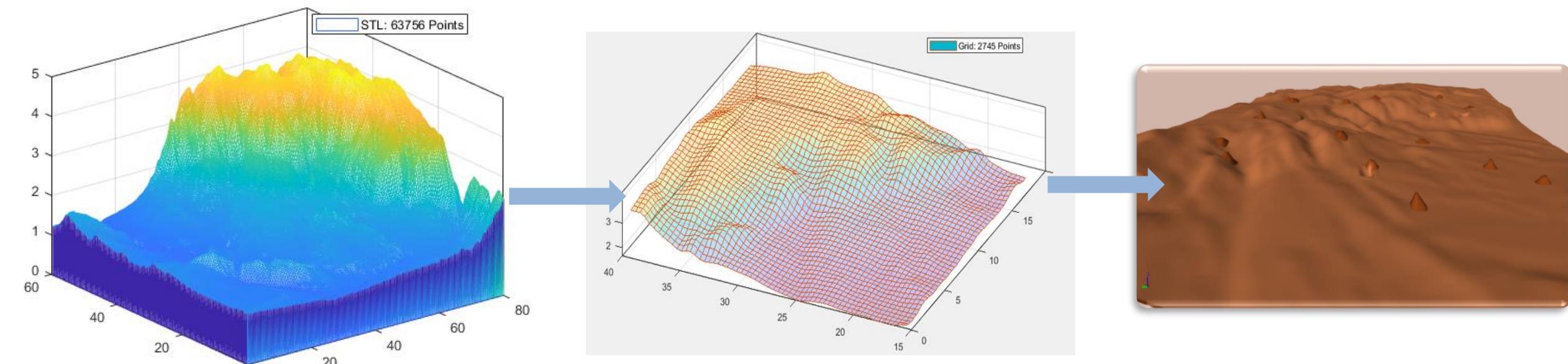
Chassis
Rocker Bogie Suspension
Wheels
Arm

Modeling the Rover



Chassis
Rocker Bogie Suspension
Wheels
Arm
Motion Analysis

Virtual Mars Terrain

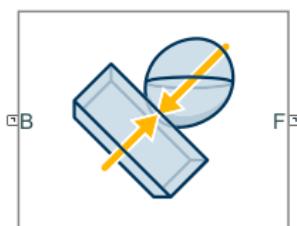
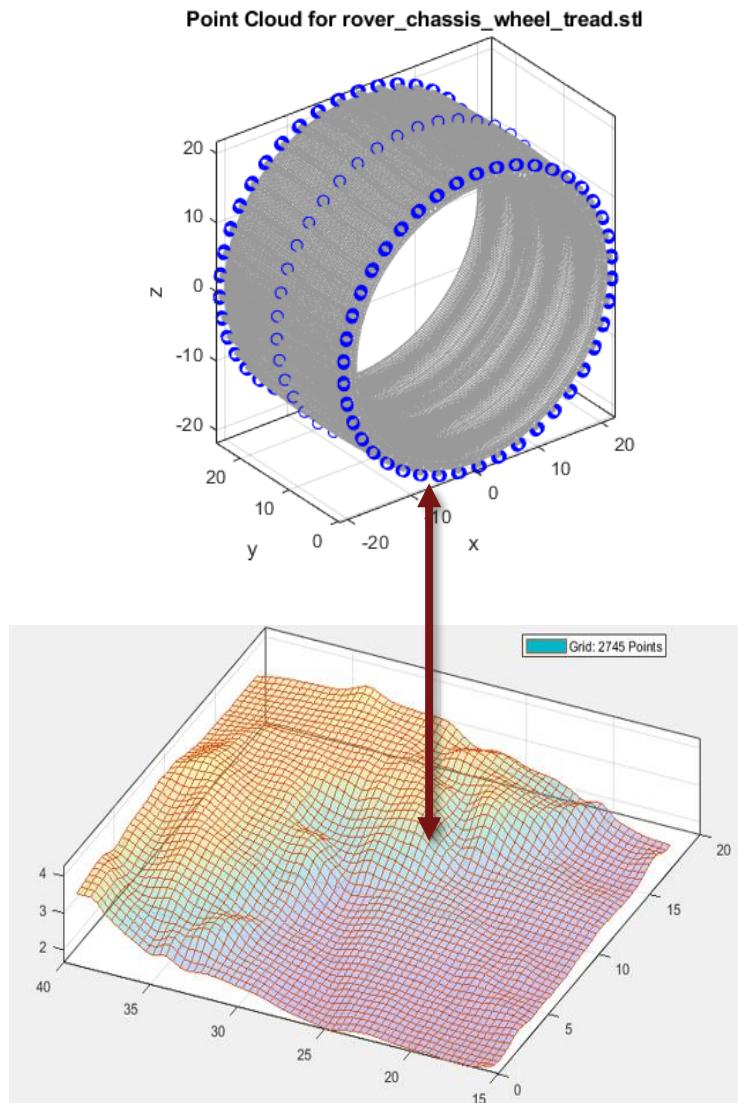


Terrain point cloud constructed using
camera + satellite images

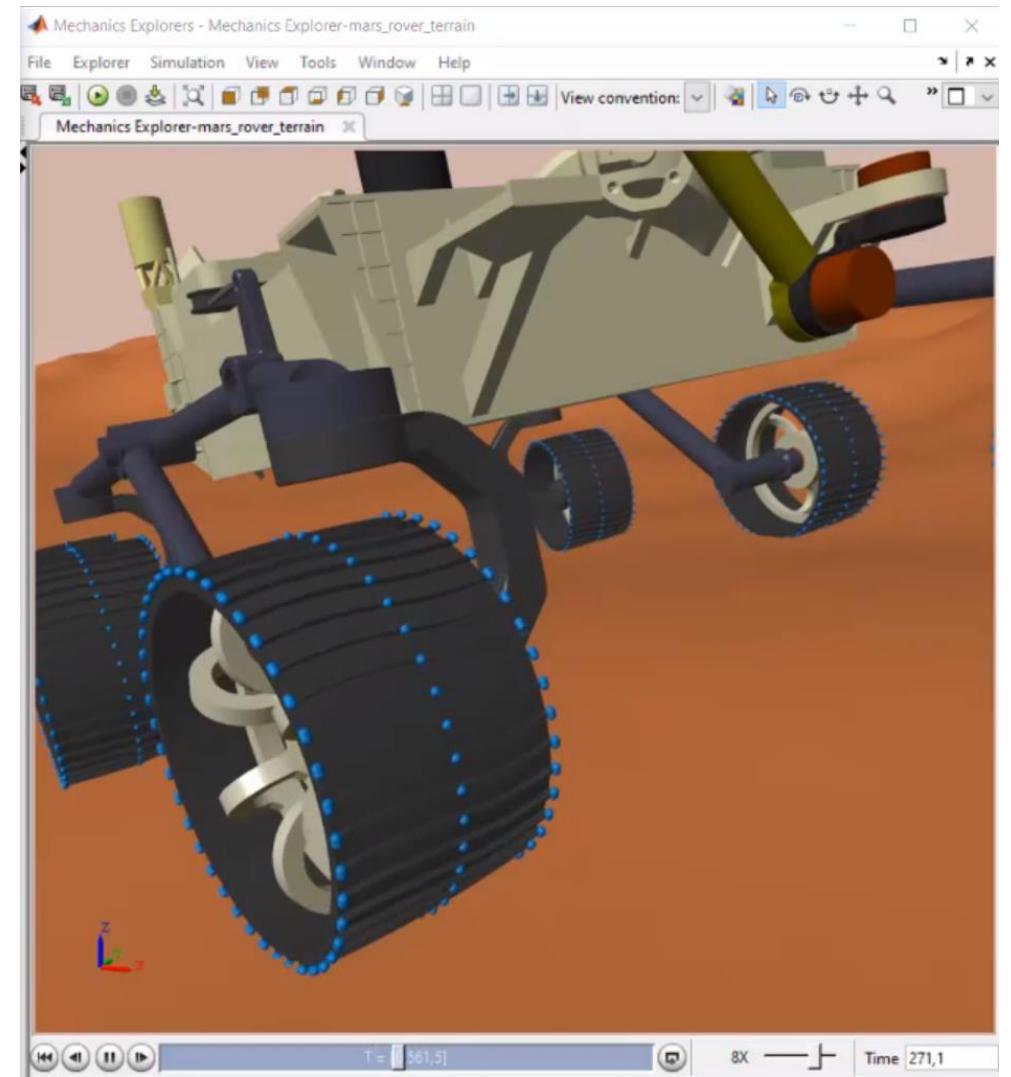
MATLAB to clean up data
+ create grid surface

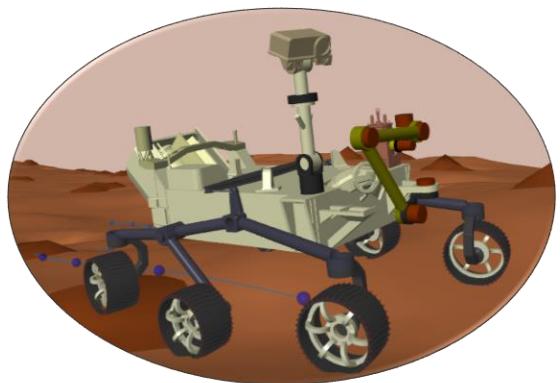
Simscape for terrain visualization
+ physics simulation

Modeling Ground forces



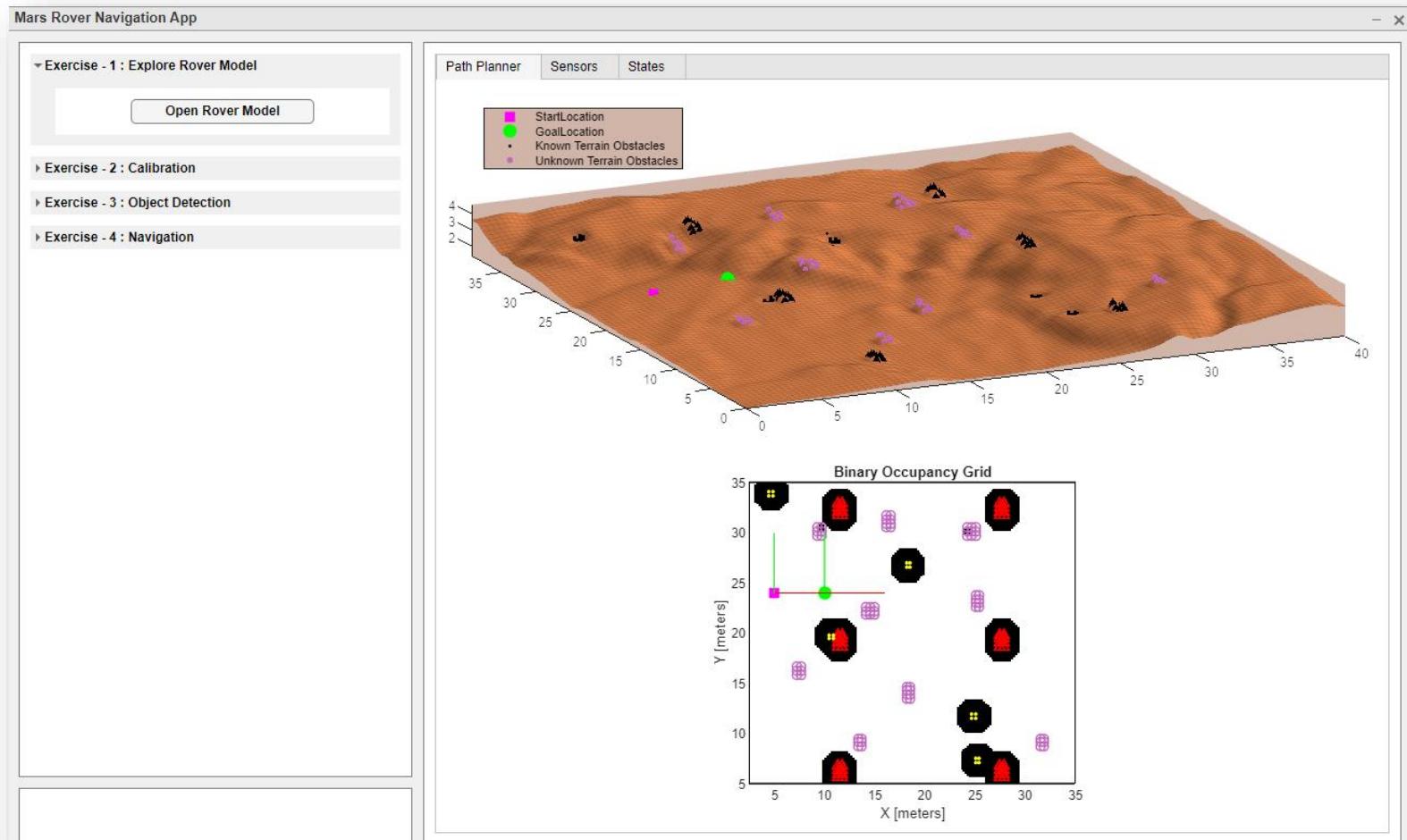
Spatial contact
force block





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

- X

Exercise - 1 : Explore Rover Model

[Open Rover Model](#)

Exercise - 2 : Calibration

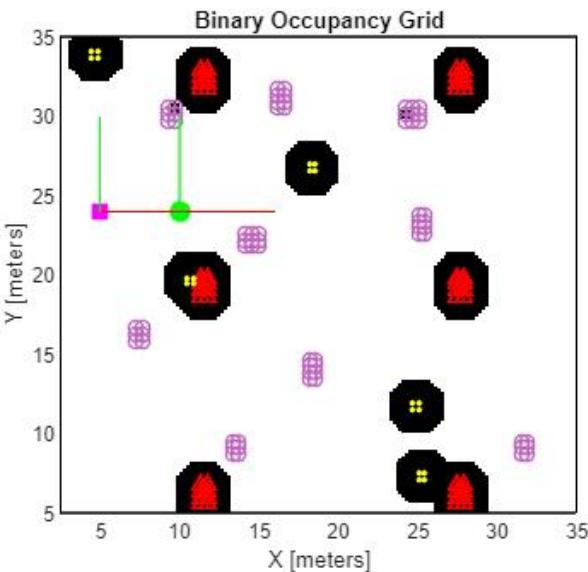
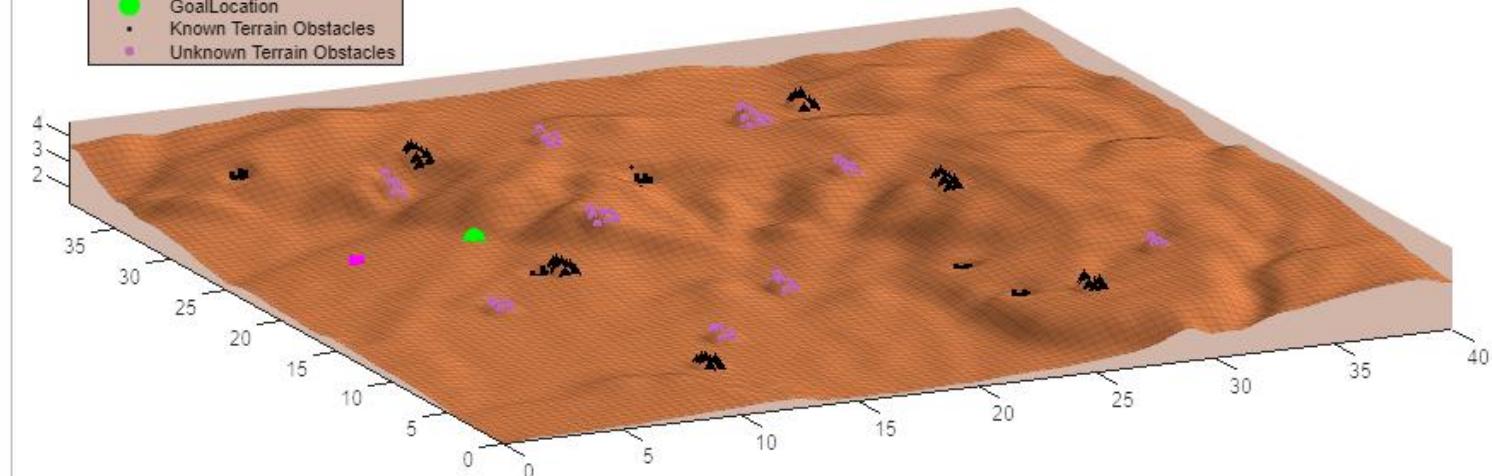
Exercise - 3 : Object Detection

Exercise - 4 : Navigation

Explore Rover Model

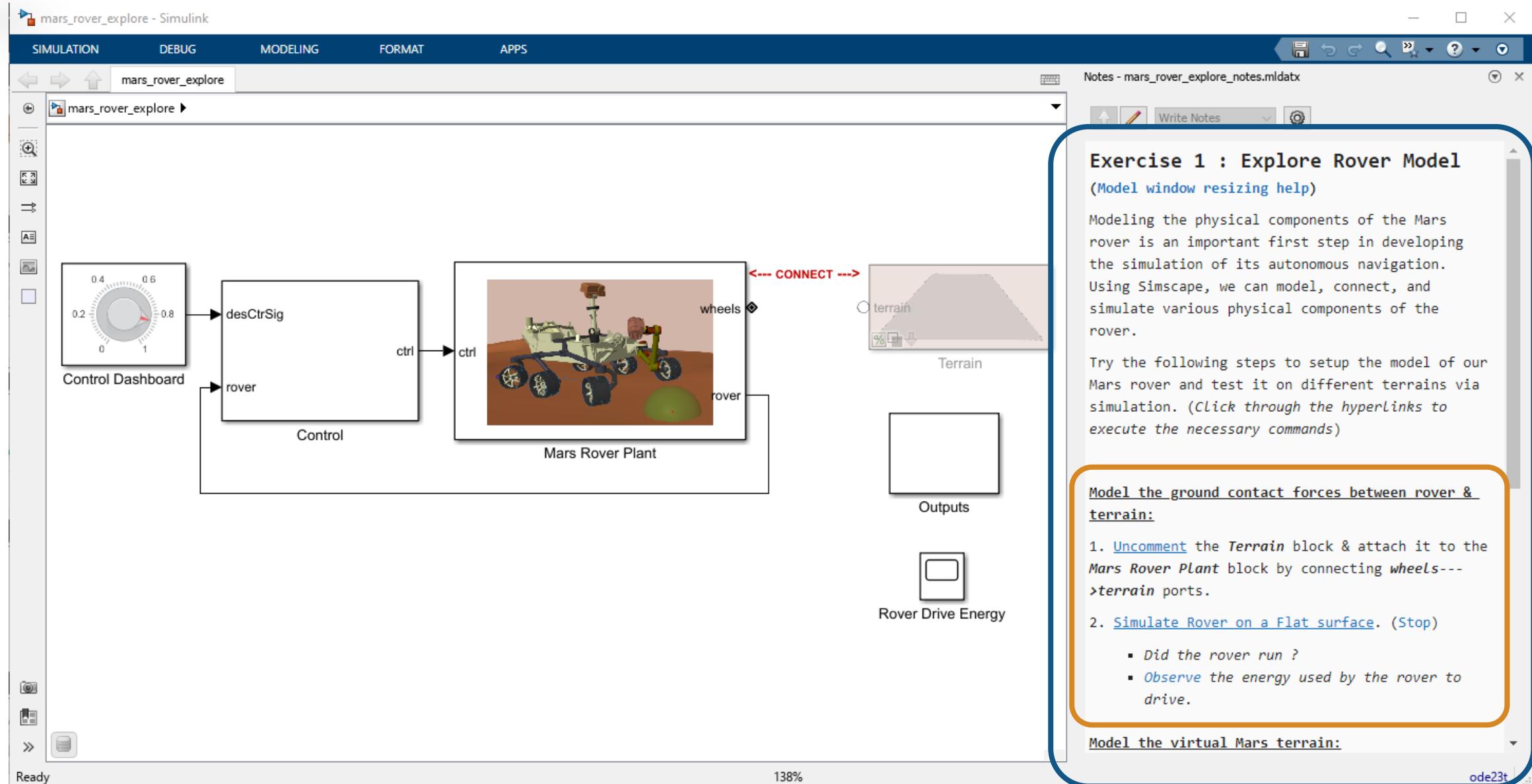
Path Planner Sensors States

- StartLocation
- GoalLocation
- Known Terrain Obstacles
- Unknown Terrain Obstacles





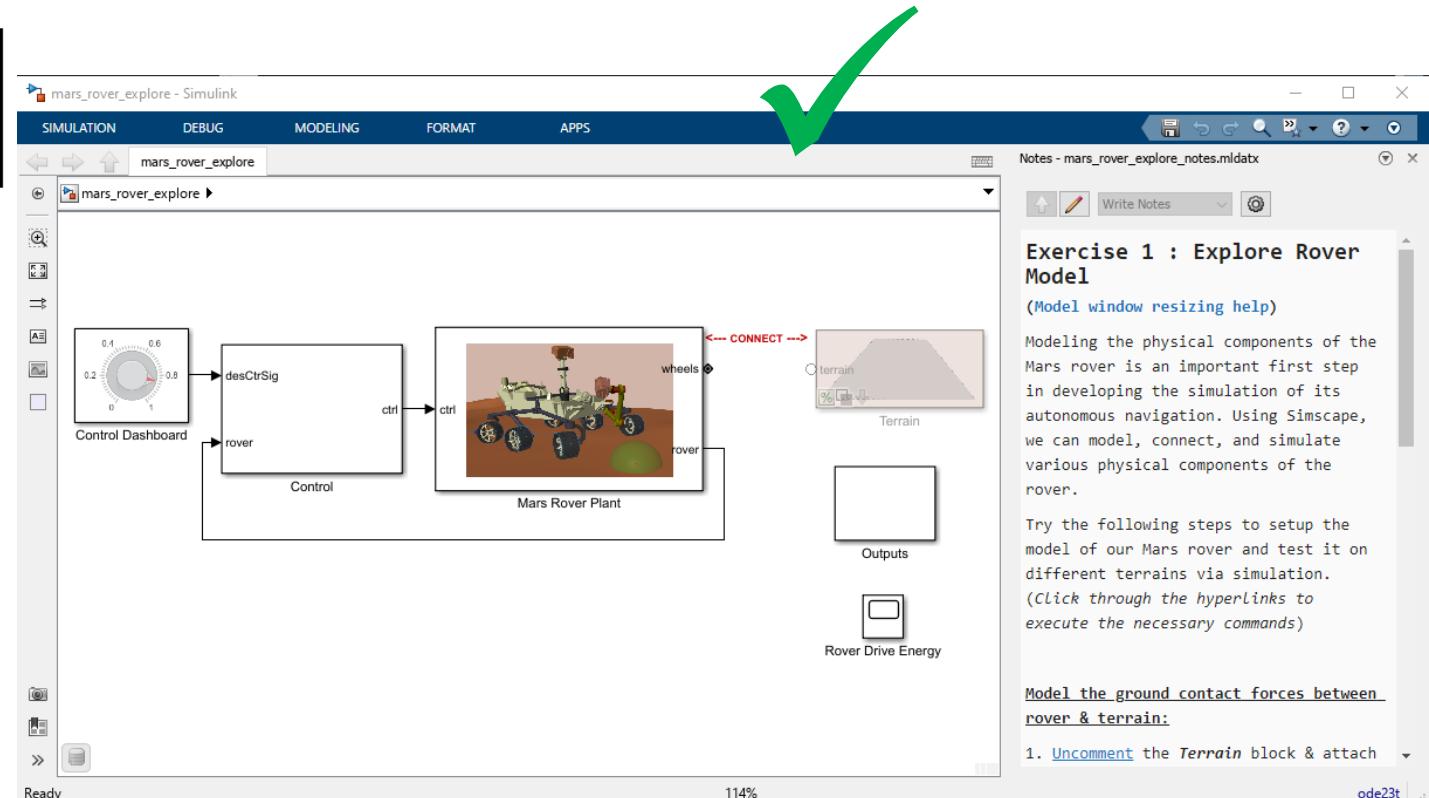
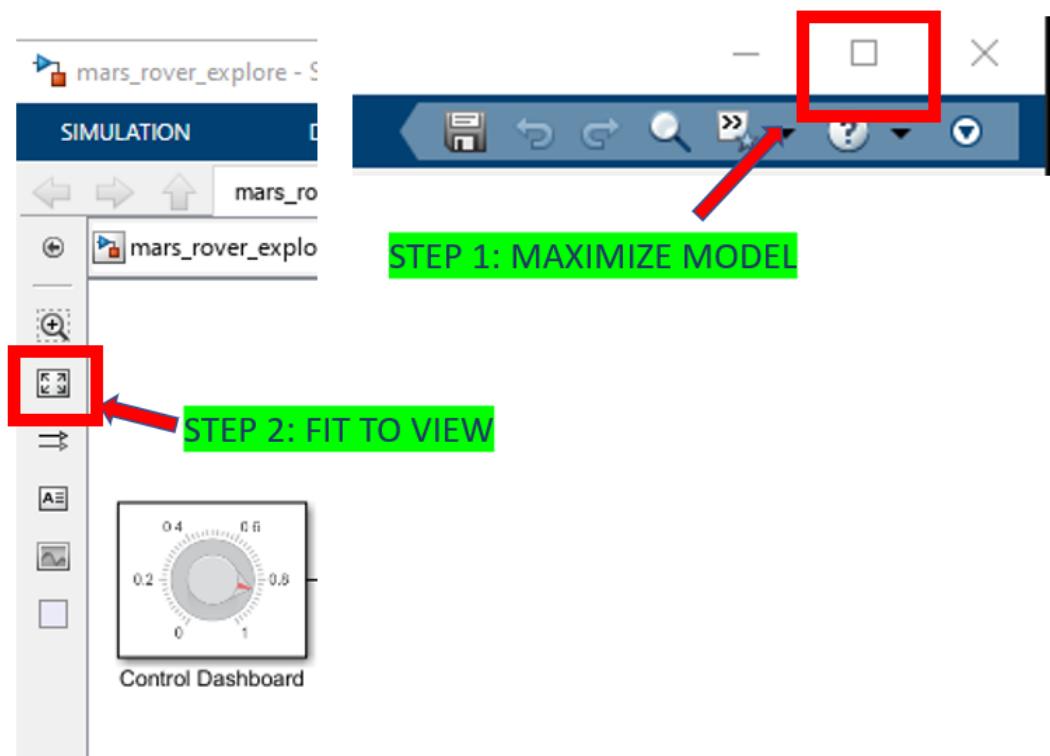
Opens mars_rover_explore.slx





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

Control Dashboard

Control

desCtrSig

wheels

ctrl

rover

terrain

Terrain

Notes - mars_rover_explore_notes.mldatx

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 ?

37



Exercise 1 : Model Virtual Mars Terrain

mars_rover_explore * - Simulink

SIMULATION DEBUG MODELING FORMAT APPS

Control Dashboard

desCtrSig → rover

rover → Control

Control → mars_rover_explore

wheels → terrain

Mechanics Explorer-mars_rover_explore_sol

Notes - mars_rover_explore_notes.midatx

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

Ready

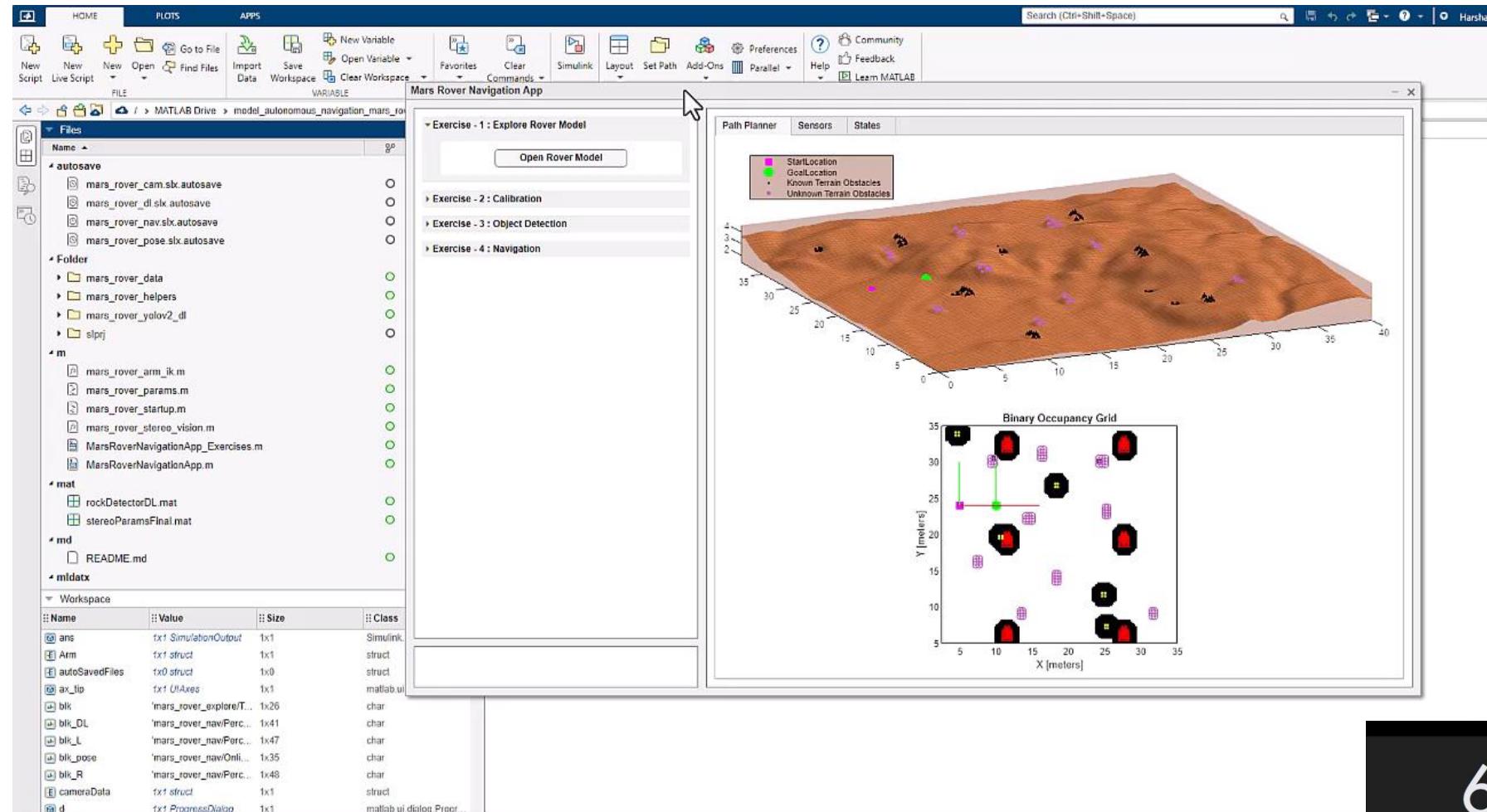
Time

32.70s / 2.0s 42 1x Time



Please take 6 minutes to complete exercise 1

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

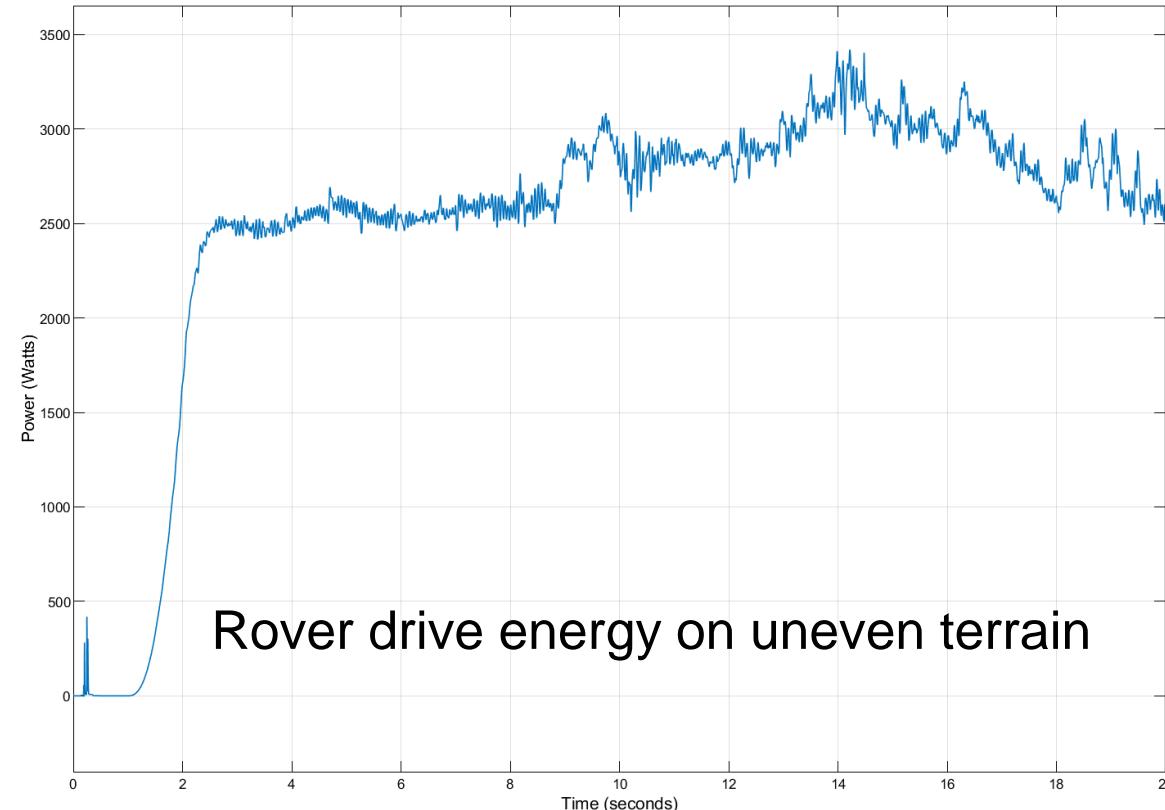
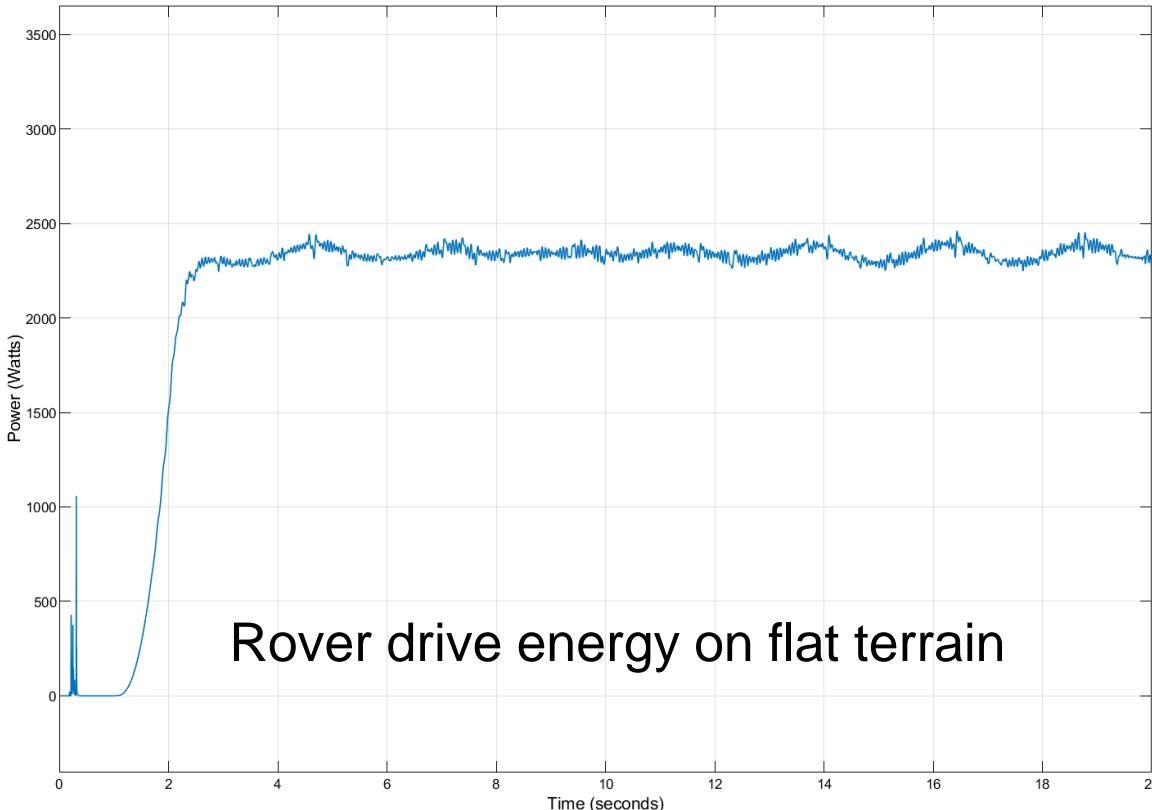
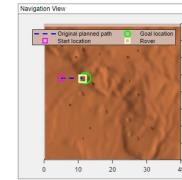


6m 00s



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

Notes - mars_rover_explore_notes.mldatax

[Write Notes](#)

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 ?

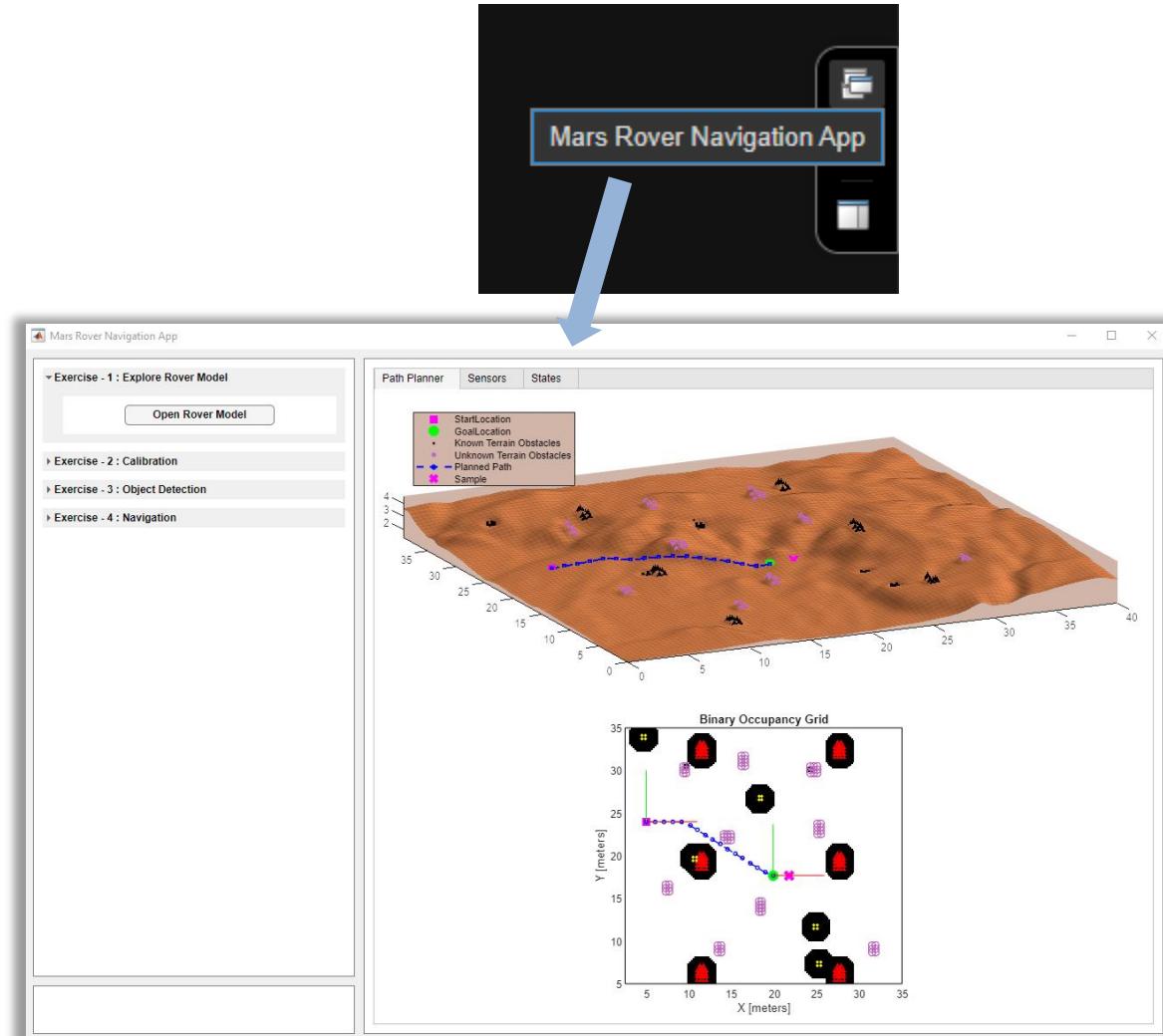
5. Close the opened dialog and Simulate Rover on the virtual Martian terrain. (Stop)

- Did the rover run on an uneven terrain ?
- Observe the energy used by the rover to drive in this case.

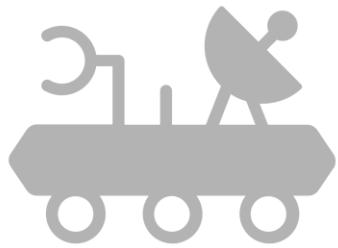
Close the model and go back to the app

ode23t

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



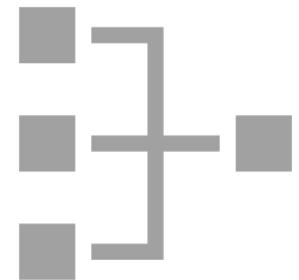
Our workshop today is split into multiple tasks



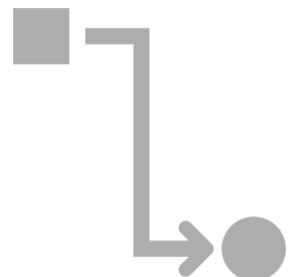
Who am I?
Rover Model



Where am I?
Calibration

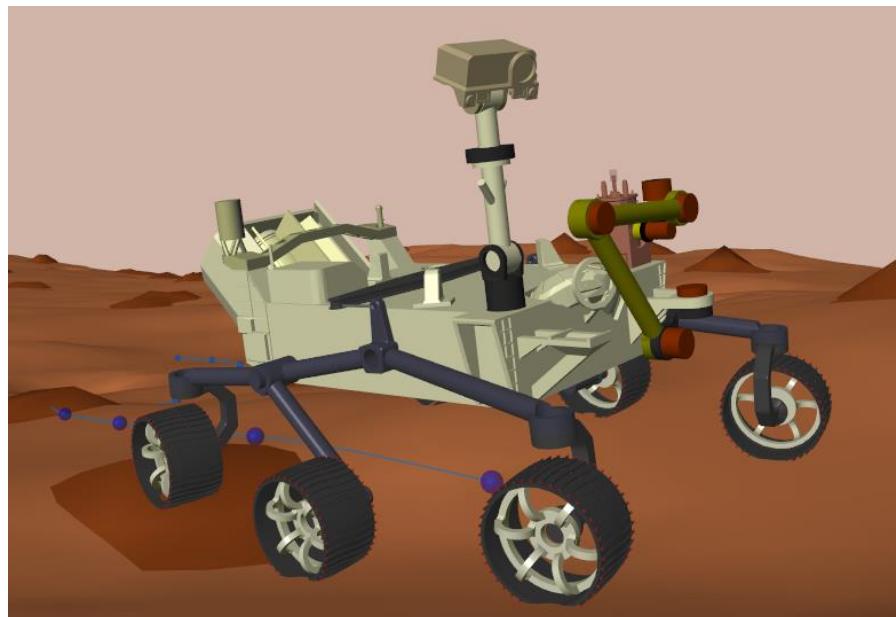


What do I see?
Object Detection



Where can I go?
Navigation

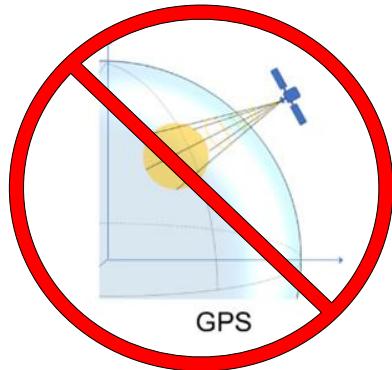
Calibrate and orient the rover using cached maps and on-board instruments



Rover Egress Phase

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

Estimating Rover Position



No GPS on Mars

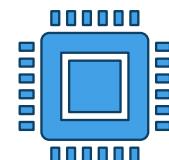


No geolocation information

On-board Sensors



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

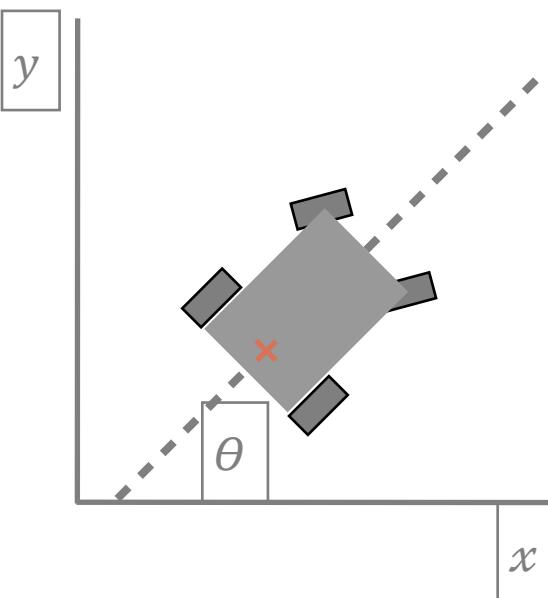
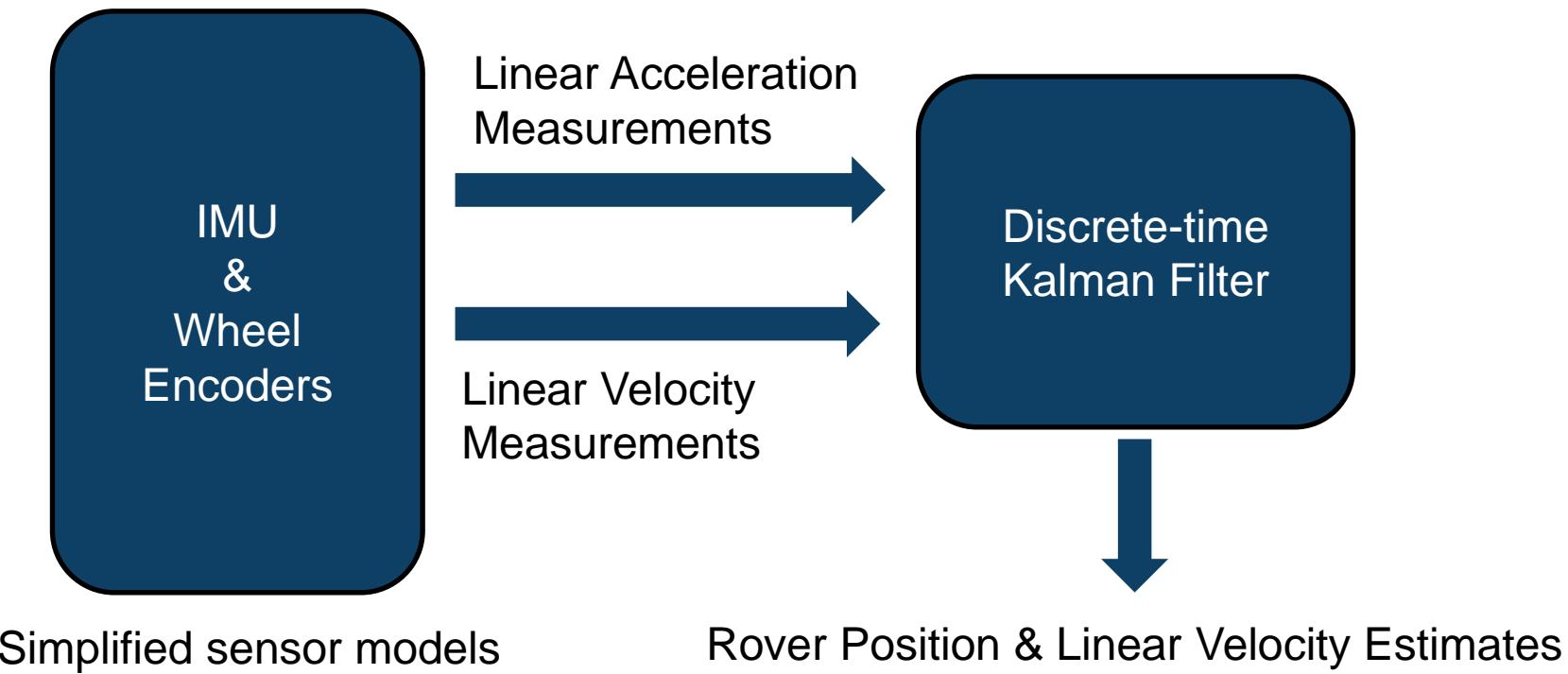


IMU (Inertial Measurement Unit) returns acceleration measurements but results in large drift due to error accumulation

Estimating Rover Position

A Kalman filter is an **optimal estimation algorithm**.

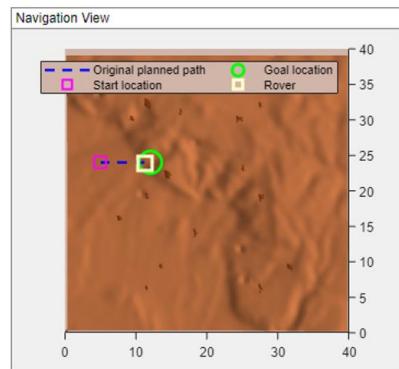
It helps to estimate a system state when it cannot be measured directly



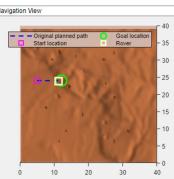
Simplified sensor models

Rover Position & Linear Velocity Estimates

Wheel Encoders + IMU = Better Pose Estimation



Exercise 2: Calibration



Calibrate camera

Mars Rover Navigation App

Exercise - 1 : Explore Rover Model

Exercise - 2 : Calibration

Calibrate Sensors

Camera Pitch (10 to 30 deg) Camera Pan (-120 to 120 deg) to

Calibrate

Calibrate Rover Position

x(m) y(m) yaw(deg)

Start Location
Goal Location Optimal position estimator

Plan an Egress Path **Calibrate**

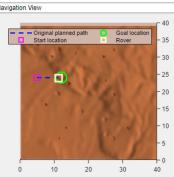
Exercise - 3 : Object Detection

Exercise - 4 : Navigation

Path Planner Sensors States

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

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

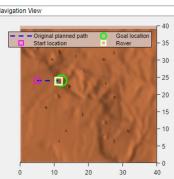
Exercise - 4 : Navigation

Path Planner Sensors States

StartLocation GoalLocation Known Terrain Obstacles Unknown Terrain Obstacles

Binary Occupancy Grid

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



Calibrate linear position

Mars Rover Navigation App

Exercise - 1 : Explore Rover Model

Exercise - 2 : Calibration

Calibrate Sensors

Camera Pitch (10 to 30 deg) Camera Pan (-120 to 120 deg) to

Calibrate Rover Position

x(m) y(m) yaw(deg)

Start Location
 Goal Location Optimal position estimator

Plan an Egress Path **Calibrate**

Exercise - 3 : Object Detection

Exercise - 4 : Navigation

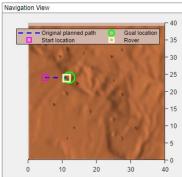
Sensors **Path Planner** **States**

Left Camera **Right Camera**

Navigation View

Original planned path
Start location
Goal location
Rover

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

Calibrate Rover Position

x(m) y(m) yaw(deg)

Start Location

Goal Location Optimal position estimator

Plan an Egress Path **Calibrate**

Exercise - 3 : Object Detection

Exercise - 4 : Navigation

Path Planner **Sensors** **States**

Rover X position

Rover Y position

Rover heading angle

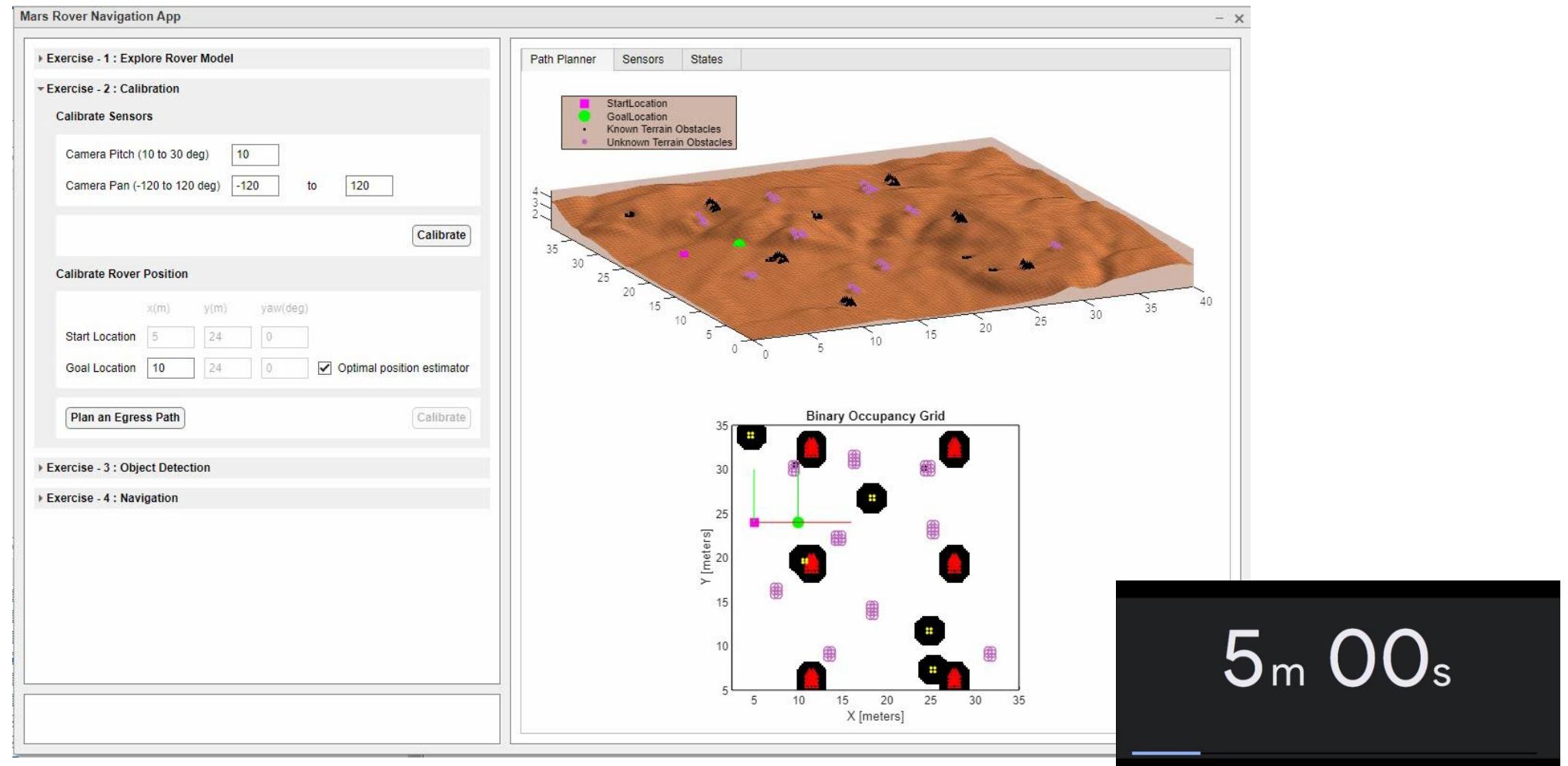
Rover path comparison in XY plane

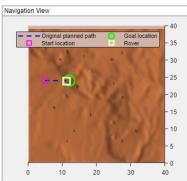
- Observe the difference in the state estimates with and without Optimal position estimator



Please take 5 minutes to complete exercise 2

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

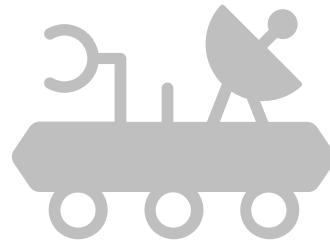




What did you observe?

- 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?

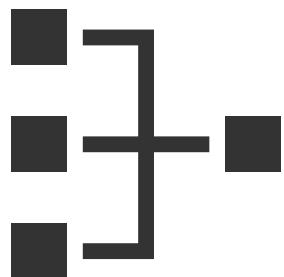
After calibration, the rover is ready to complete it's assigned tasks using AI



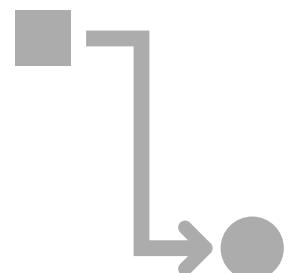
Who am I?
Rover Model



Where am I?
Calibration



What do I see?
Object Detection

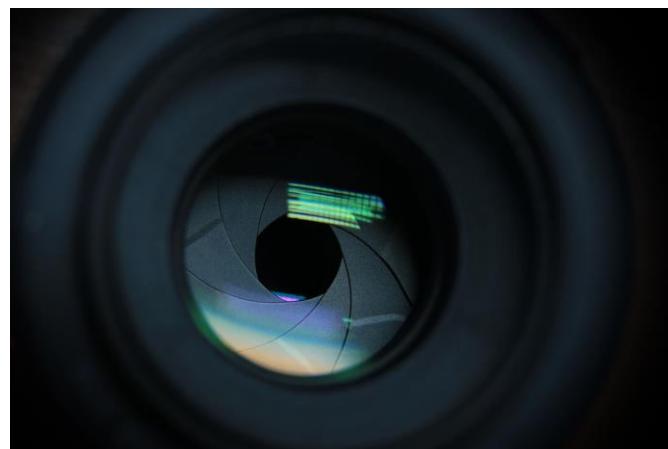


Where can I go?
Navigation

After calibration, the rover needs to traverse the terrain

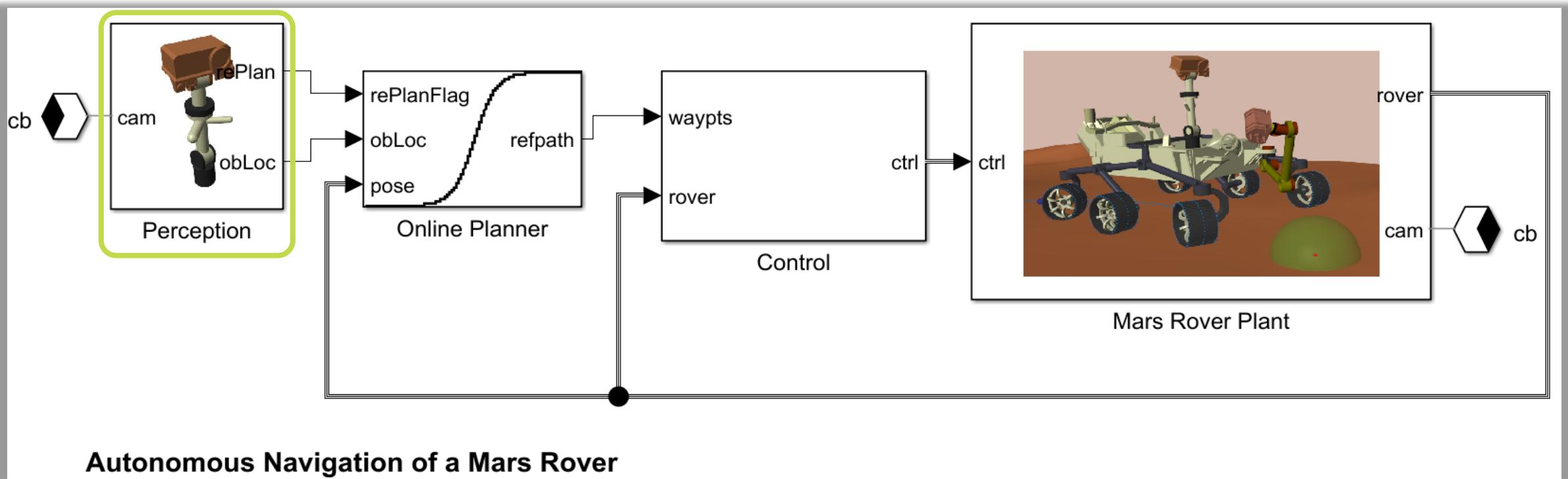


Scan area, eyes closed and walk

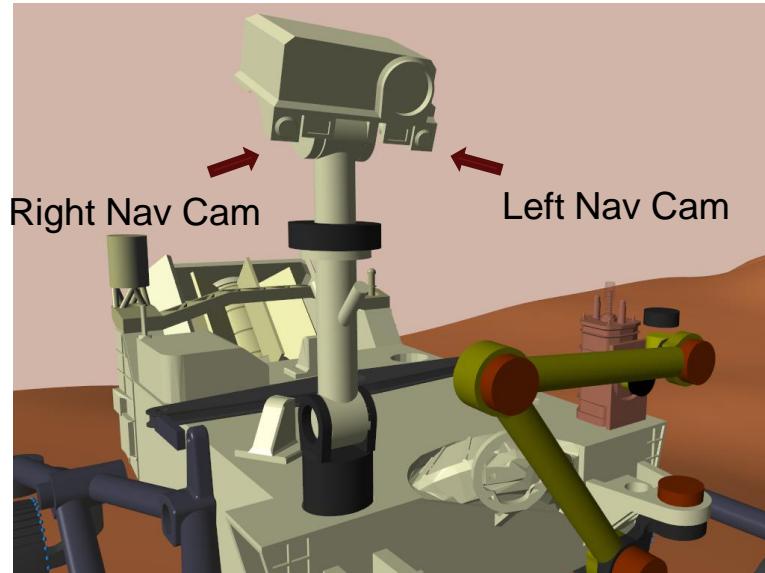


Camera sensors on the calibrated rover serve as the eyes

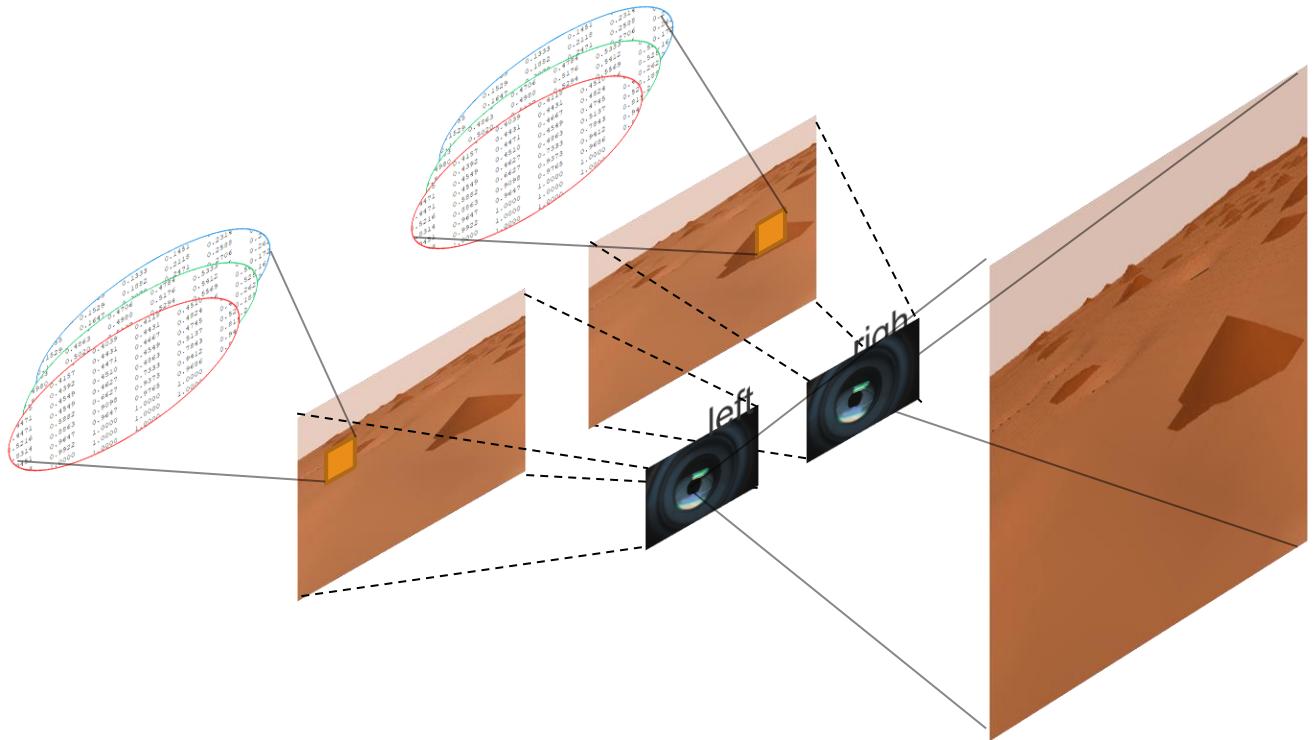
Perception Module



We can identify objects with vision sensors using AI techniques



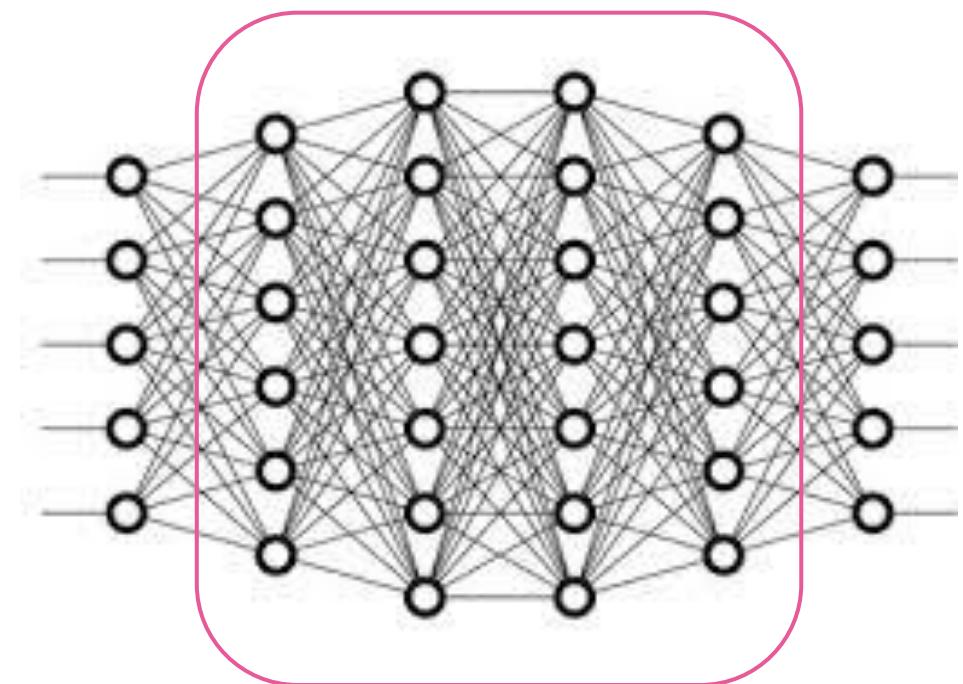
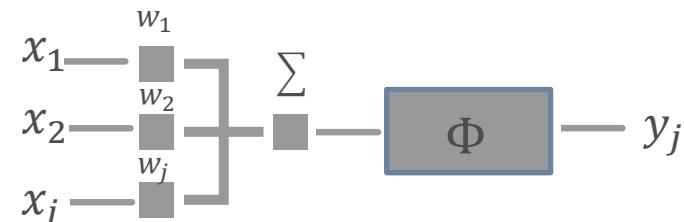
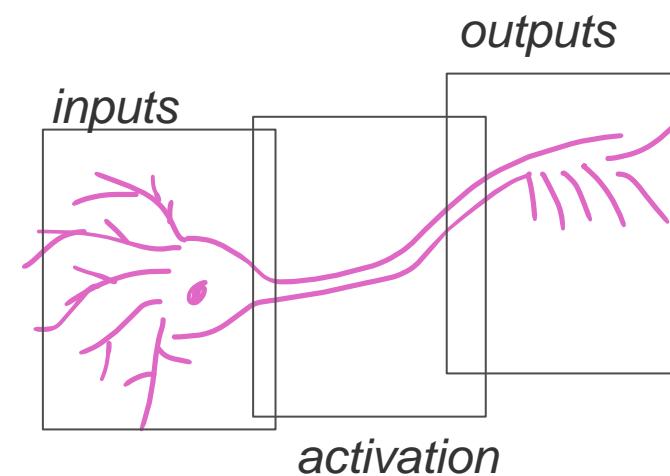
The rover collects image data using both left & right sensors



RGB matrices are fed into a DNN to identify obstacles

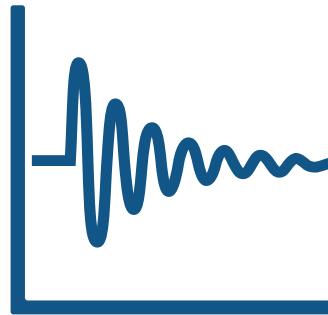
Artificial Neural Network

Deep Neural Networks are stacked neural networks



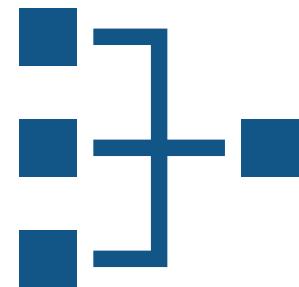
Deep Neural Network

Deep Neural Network plays a pivotal role in identifying objects



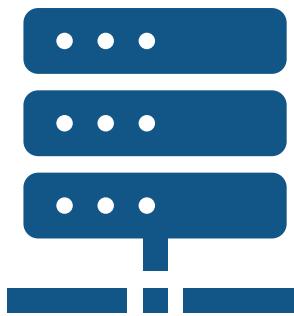
Preprocessing Data

Network is as good as your data



Object Detection Network

Resnet50 – feature extraction



Train Network

60:20:20 ratio



Validate

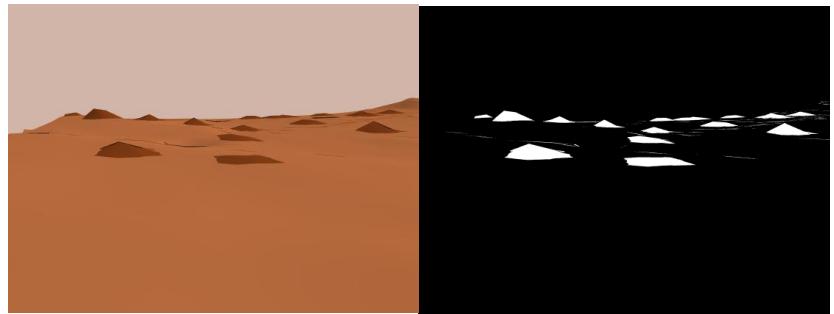
Use the last 20%



Obstacle Detection using Deep Learning

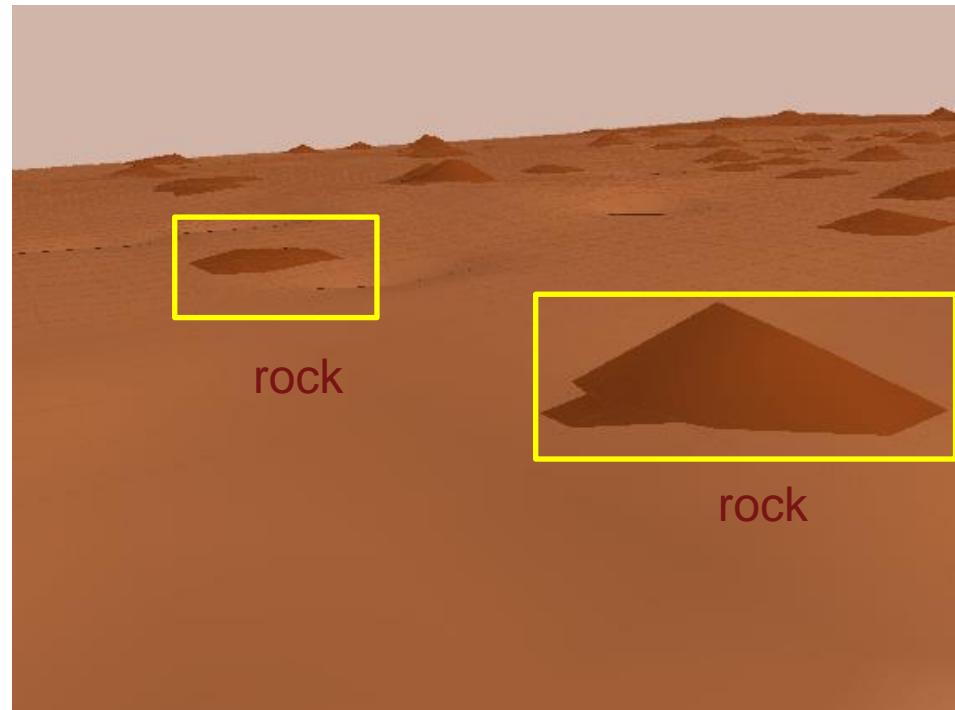


Preprocessing
Data



10K training images

Labeled using binary masking

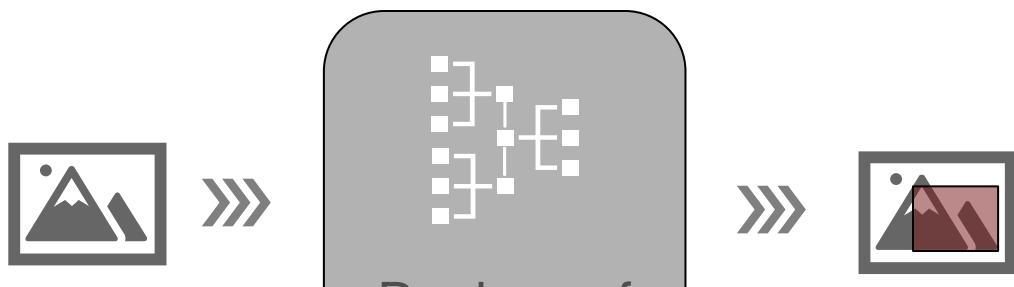




Yolov2 Object Detector

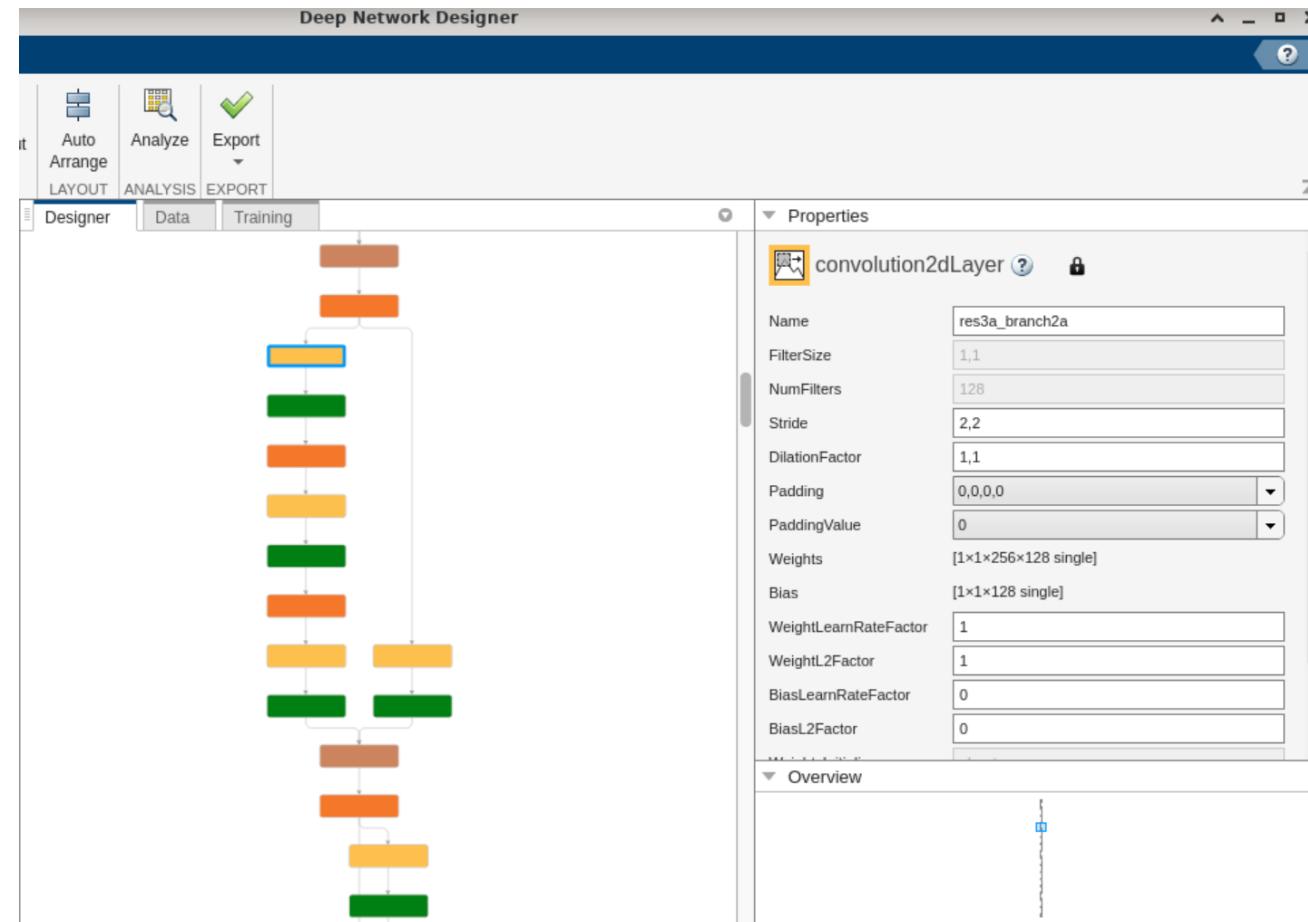
Object Detection
Network

```
>> trainYOLOv2ObjectDetector()
```

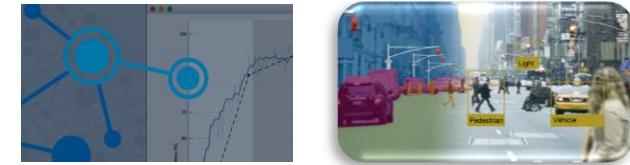


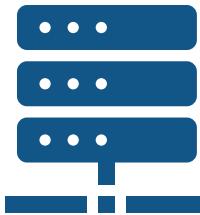
Classification
+ localization

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](#)

Import Image Classification Data

TRAINING
Import image classification data for training.
Data source: **Folder**
Select a folder with subfolders of images for each class.
LabeledBinaryRockData

AUGMENTATION OPTIONS

Random reflection axis	X: <input type="checkbox"/>	Y: <input type="checkbox"/>
Random rotation (degrees)	Min: <input type="text" value="0"/>	Max: <input type="text" value="0"/>
Random rescaling	Min: <input type="text" value="1"/>	Max: <input type="text" value="1"/>
Random horizontal translation (pixels)	Min: <input type="text" value="0"/>	Max: <input type="text" value="0"/>
Random vertical translation (pixels)	Min: <input type="text" value="0"/>	Max: <input type="text" value="0"/>

VALIDATION
Import validation data to help prevent overfitting.
Data source: **Split from training data**
Specify amount of training data to use for validation.
Percentage: Randomize

Info: Images will be resized during training to match network input size.

Obstacle Depth Estimation

Image
Processing
Toolbox

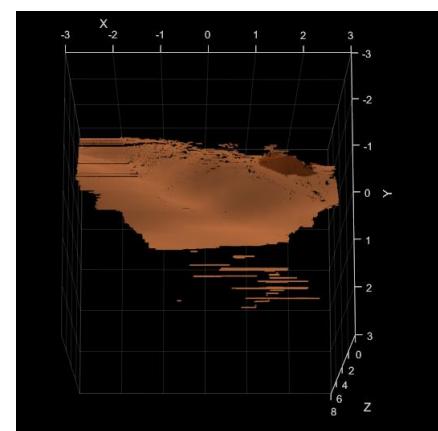
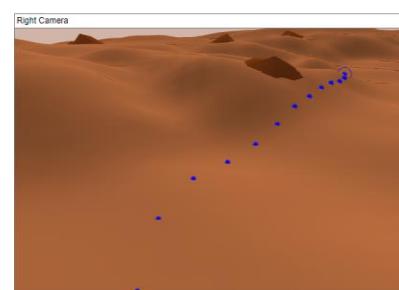
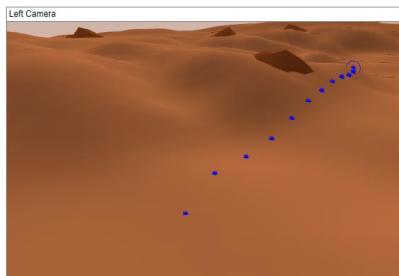


Stereo Camera
Calibration

Reconstructed
3D scene

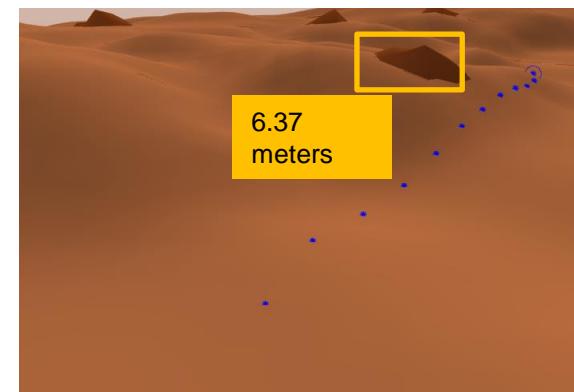
Disparity Map

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

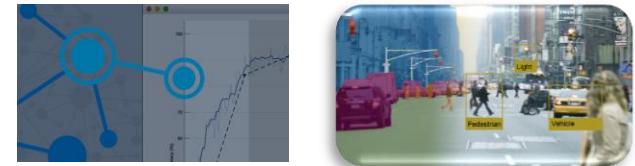


3D point cloud

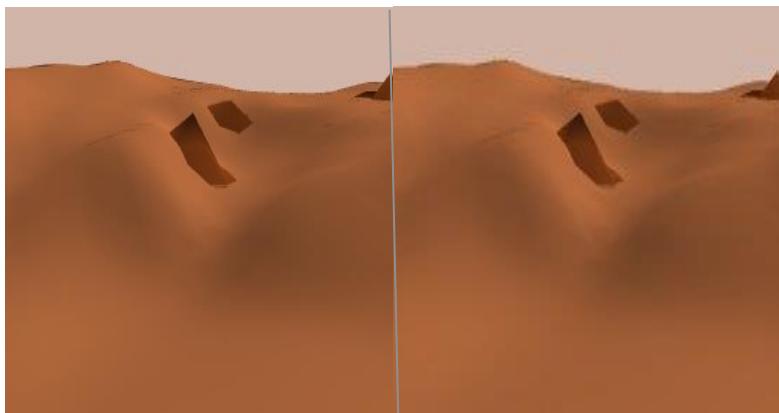
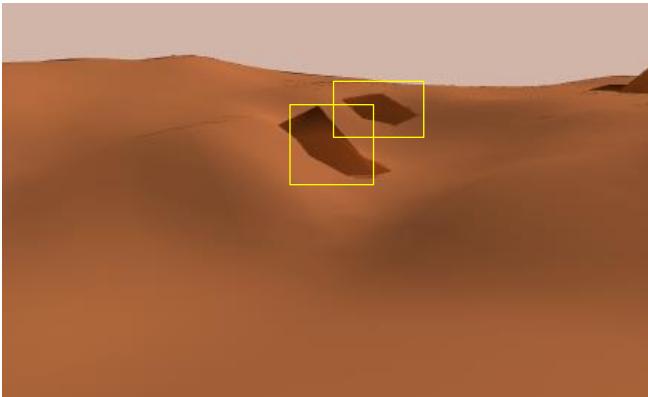
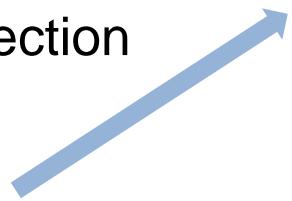
Reconstructed scene



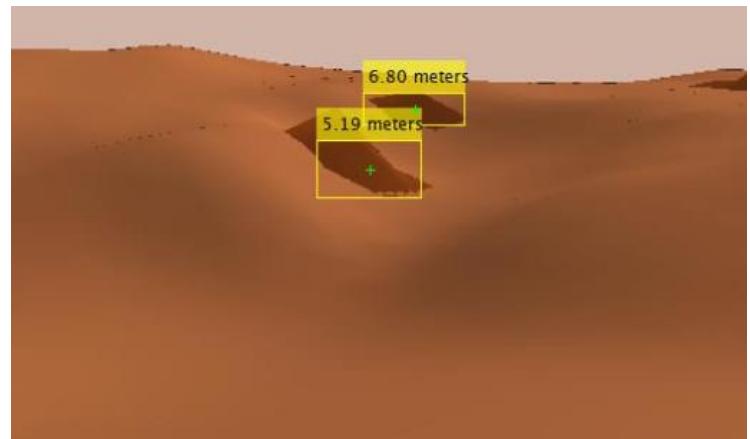
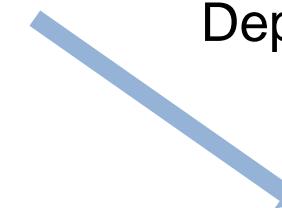
Obstacle detection

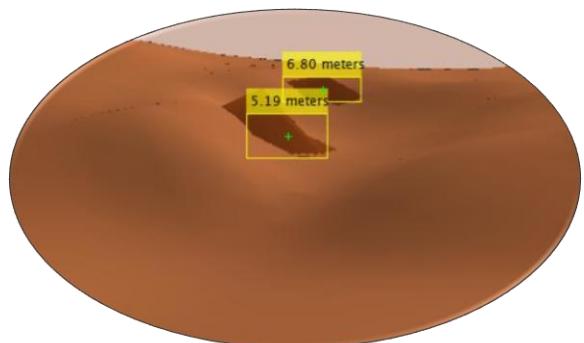


Object detection

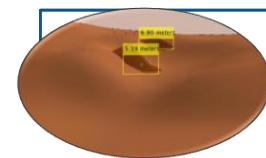


Depth Estimation





Exercise 3 : Obstacle Detection



Mars Rover Navigation App

Path Planner Sensors States

Exercise - 1 : Explore Rover Model

Exercise - 2 : Calibration

Exercise - 3 : Object Detection

Camera Pan (-120 to 120 deg) to

Detection Threshold (0 to 1)

Detect

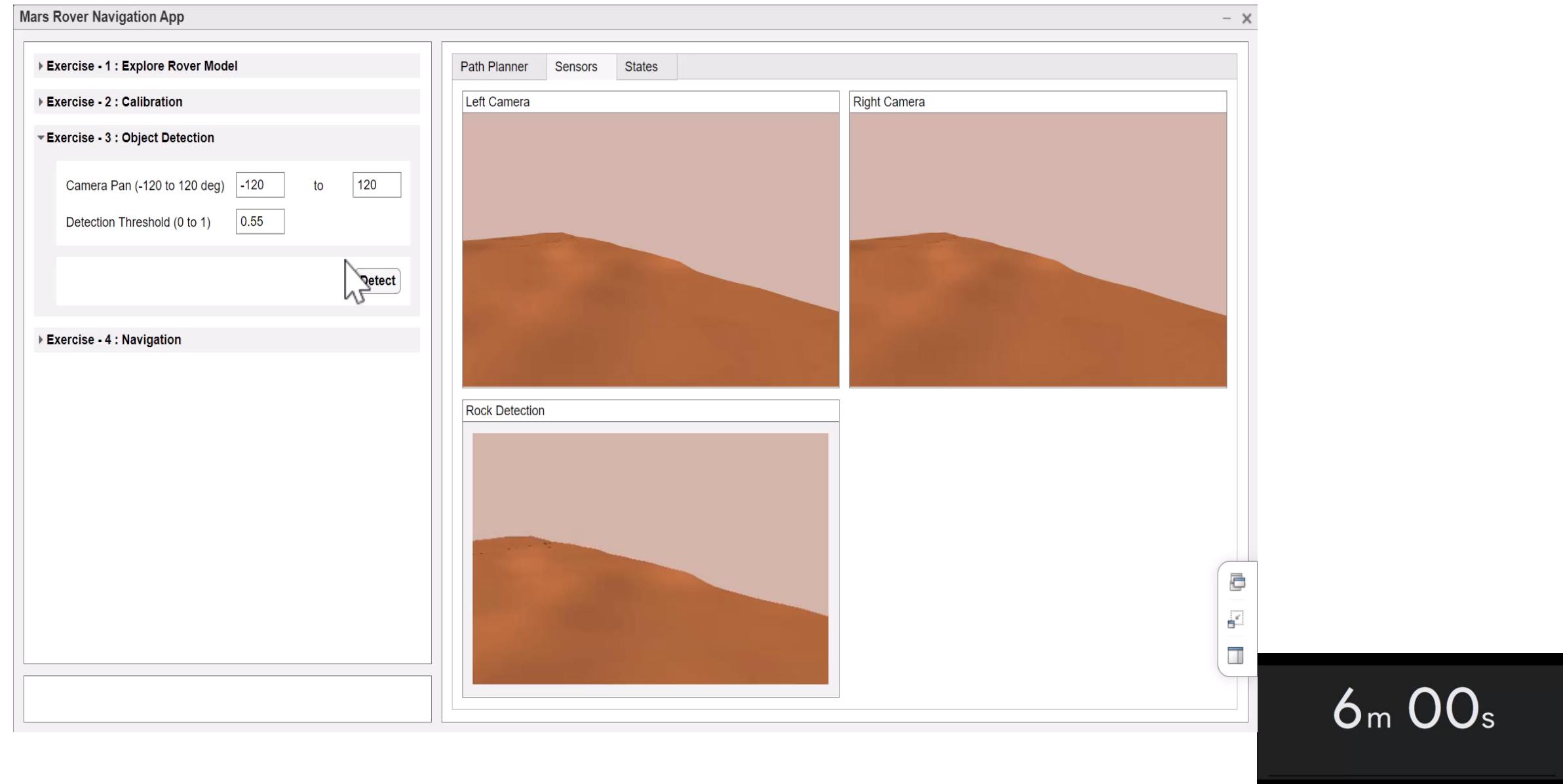
Left Camera

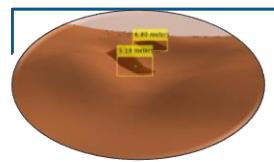
Right Camera

Rock Detection

The application interface for the Mars Rover Navigation App. On the left, a sidebar lists exercises: 'Exercise - 1 : Explore Rover Model', 'Exercise - 2 : Calibration', 'Exercise - 3 : Object Detection' (which is currently selected and highlighted with a green border), and 'Exercise - 4 : Navigation'. The main area contains three camera feeds: 'Left Camera' and 'Right Camera' under the 'Sensors' tab, and a 'Rock Detection' feed under the 'States' tab. The 'Sensors' tab is active, indicated by a blue border around its tab header. In the 'Object Detection' section, there are input fields for 'Camera Pan' (ranging from -120 to 120 degrees) and 'Detection Threshold' (ranging from 0 to 1), with values set to -120, 120, and 0.55 respectively. A yellow-bordered 'Detect' button is located below these inputs. The 'Left Camera' and 'Right Camera' feeds show a 3D rendering of a reddish-brown Mars surface with a prominent rocky outcrop. The 'Rock Detection' feed shows the same scene with a dashed white outline highlighting the detected rock feature.

Please take 6 minutes to complete exercise 3

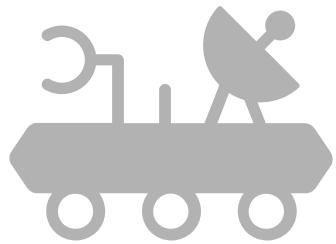




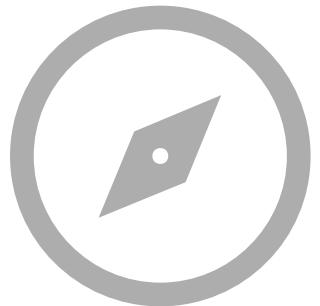
What did you observe?

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

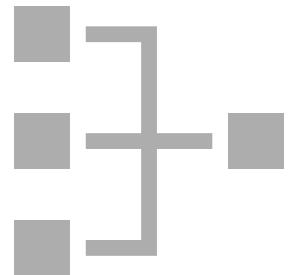
Onto the next task...



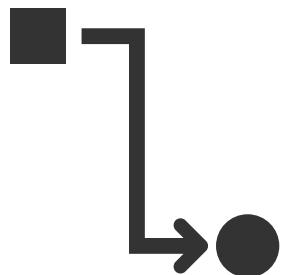
Who am I?
Rover Model



Where am I?
Calibration

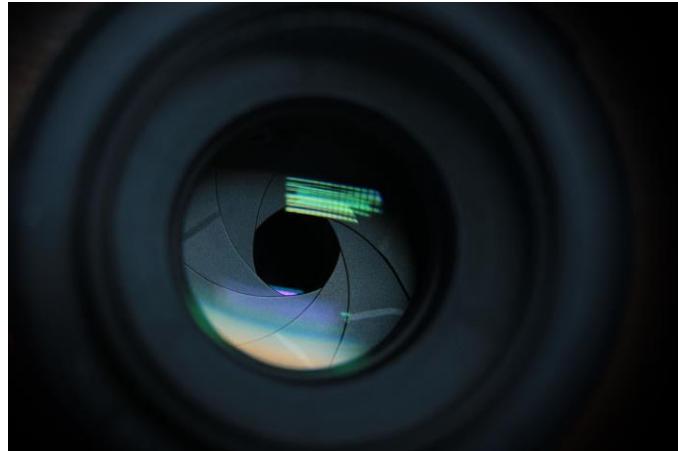


What do I see?
Object Detection

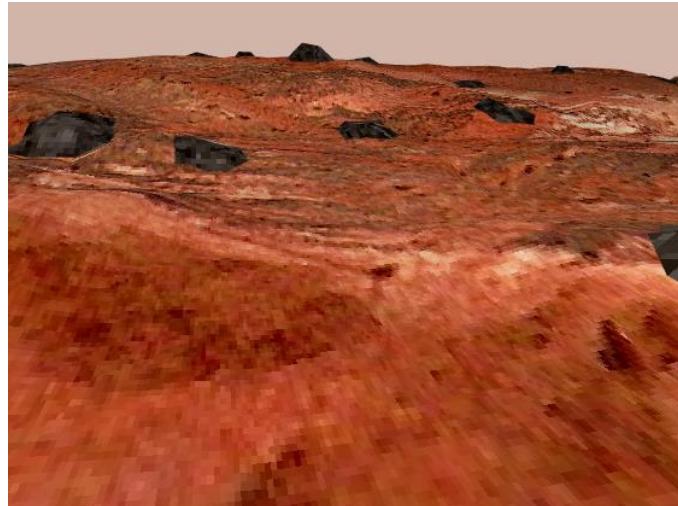


Where can I go?
Navigation

After correctly identifying objects, we need to traverse



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



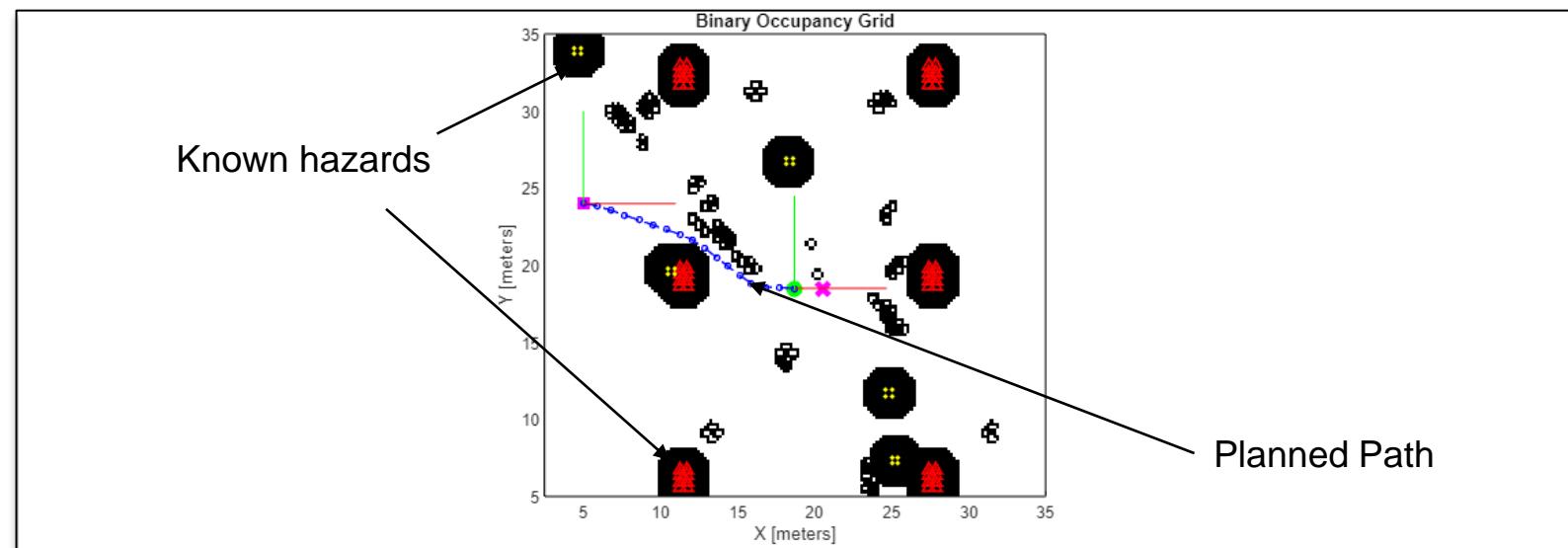
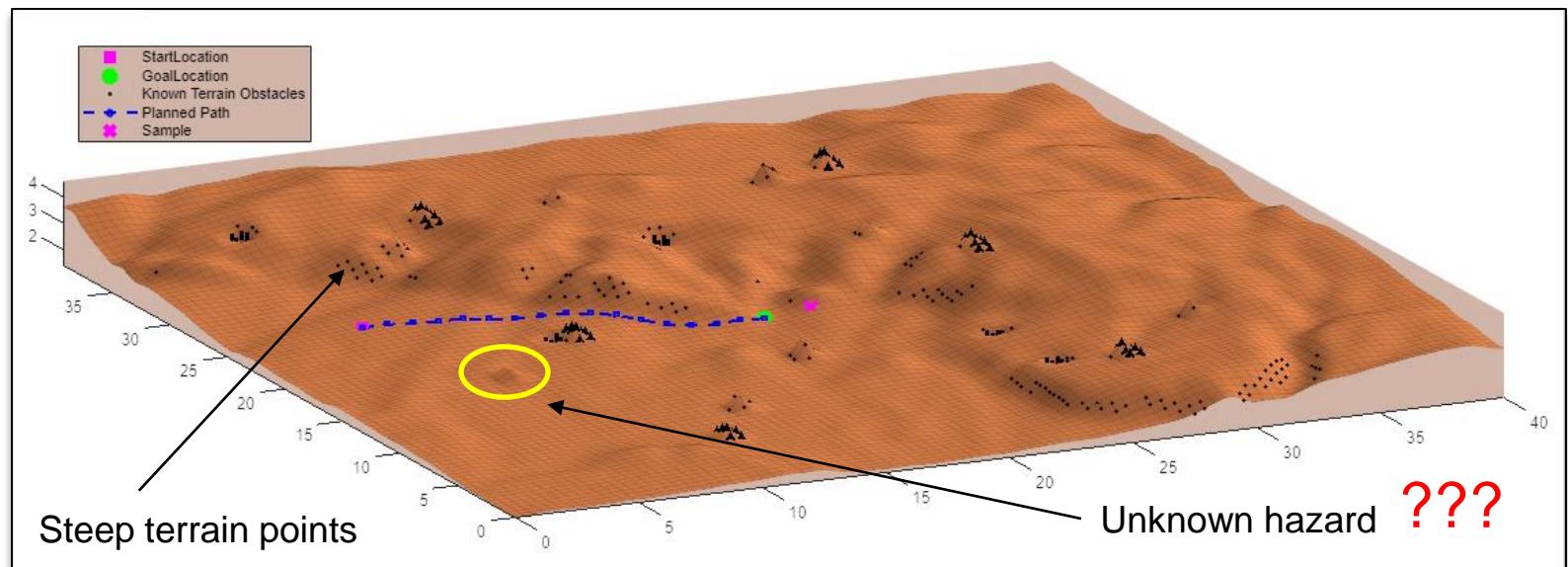
With predetermined maps and updates from DNN, the rover should be able to carry out its mission successfully

Offline Path Planning

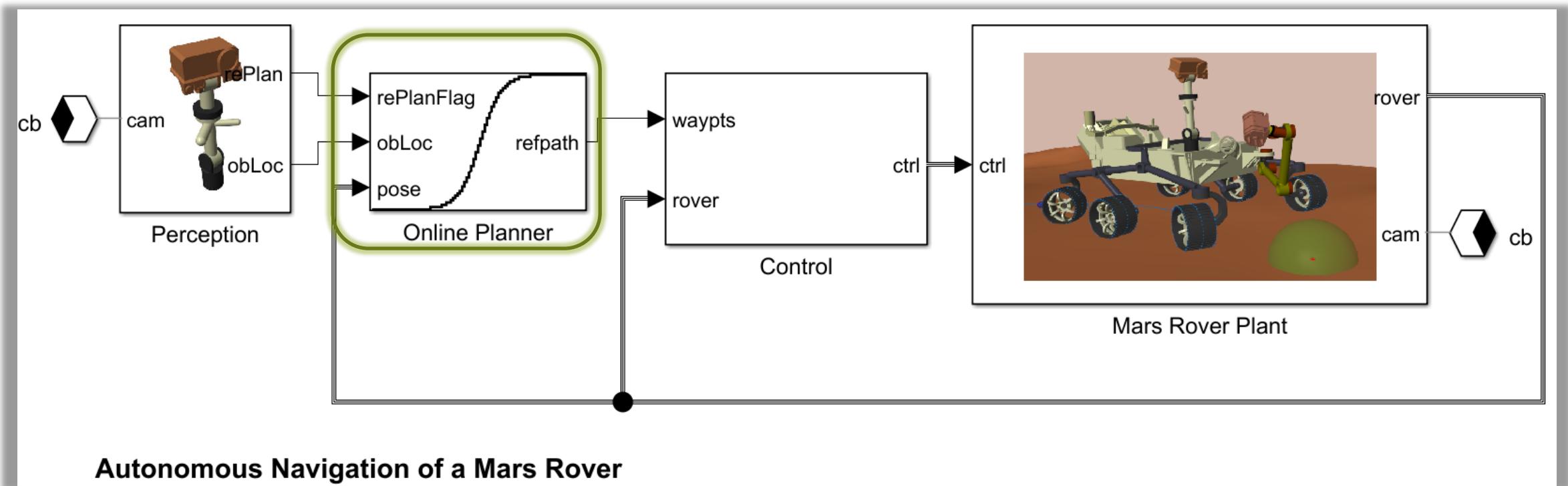
3D terrain map known using point cloud reconstruction from multiple satellite / stereo vision images.

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

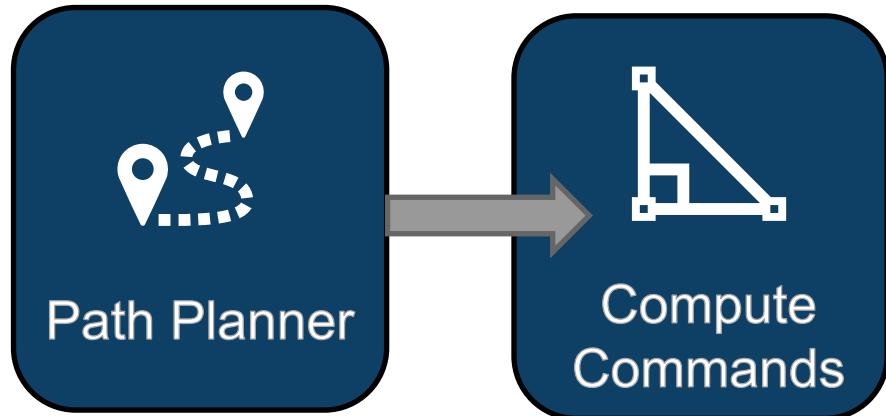
Hybrid A star algorithm is used to plan paths avoiding the known obstacles.



Online Planner / Obstacle Avoidance



Obstacle Avoidance helps navigate in unknown terrains



A to B?

Find an optimal
path using A*

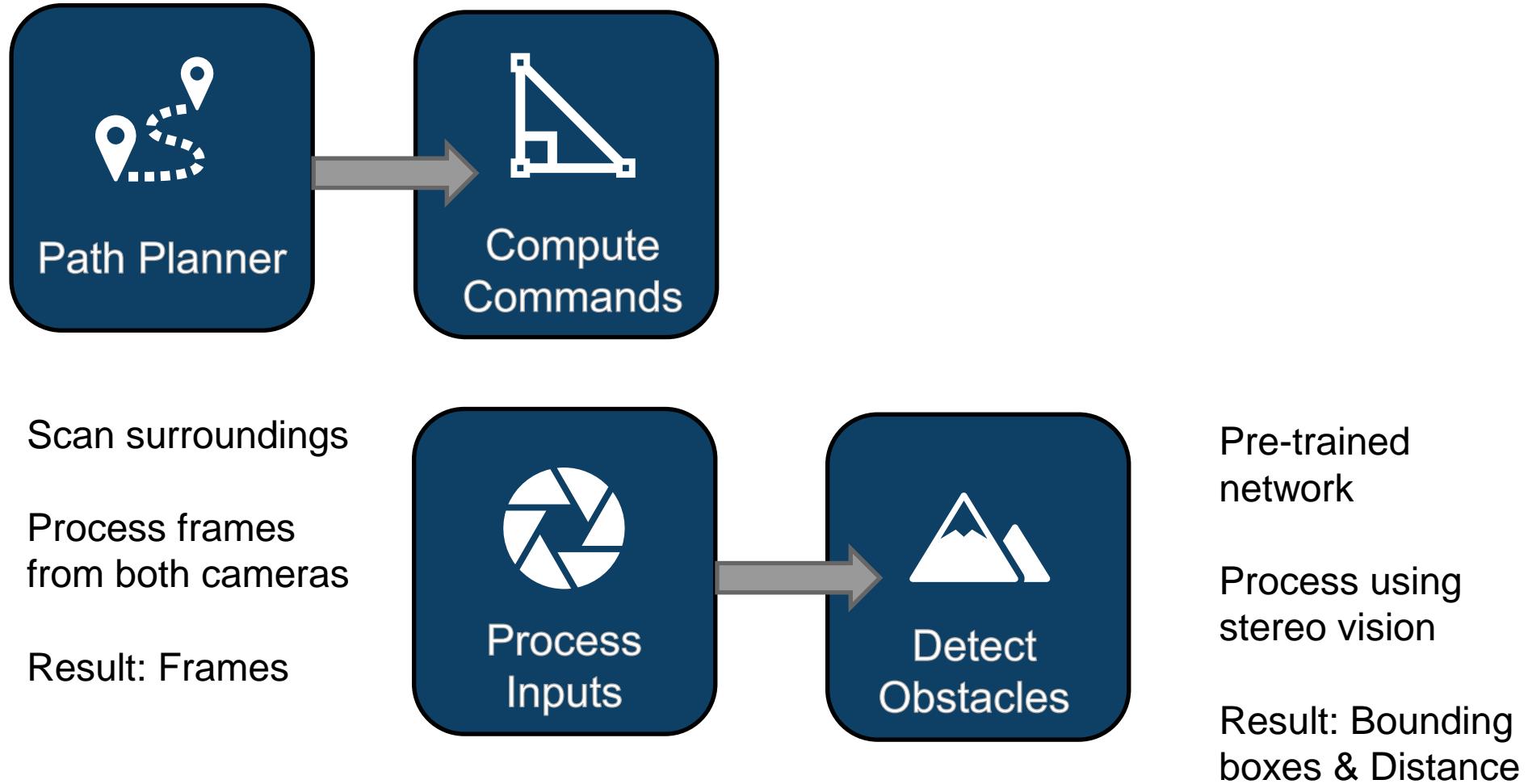
Result: Waypoints

Waypoints are a set of
coordinates

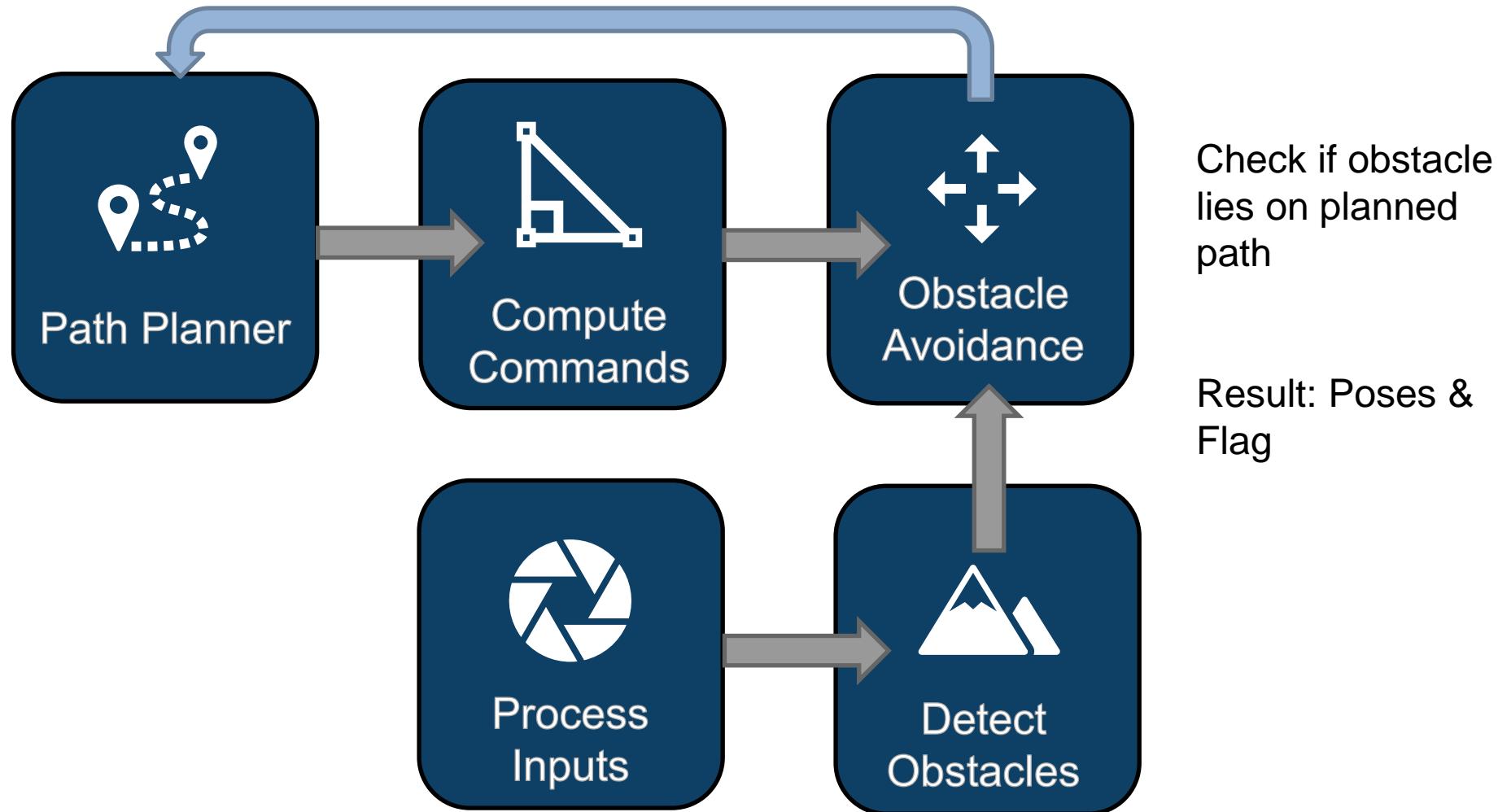
Compute commands
using Pure Pursuit

Result: Velocity and
Theta

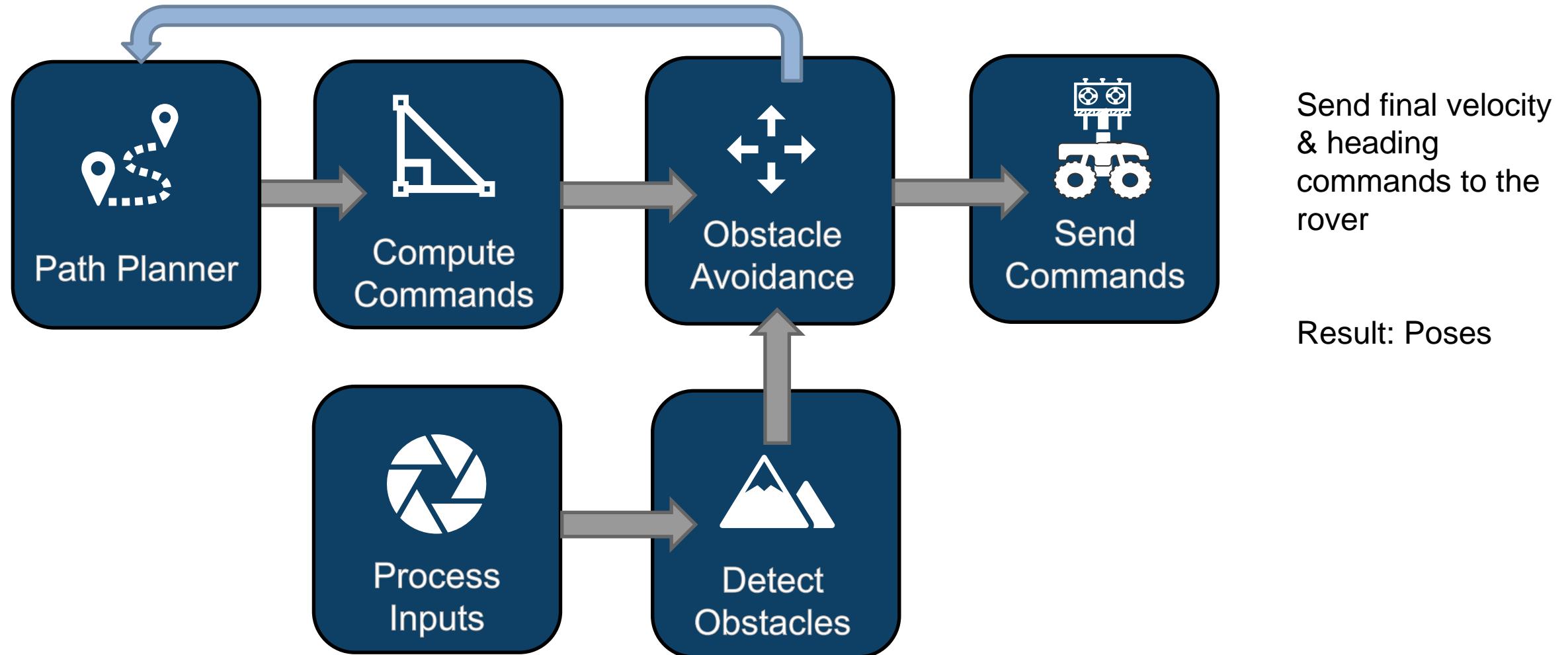
Obstacle Avoidance helps navigate in unknown terrains



Obstacle Avoidance helps navigate in unknown terrains

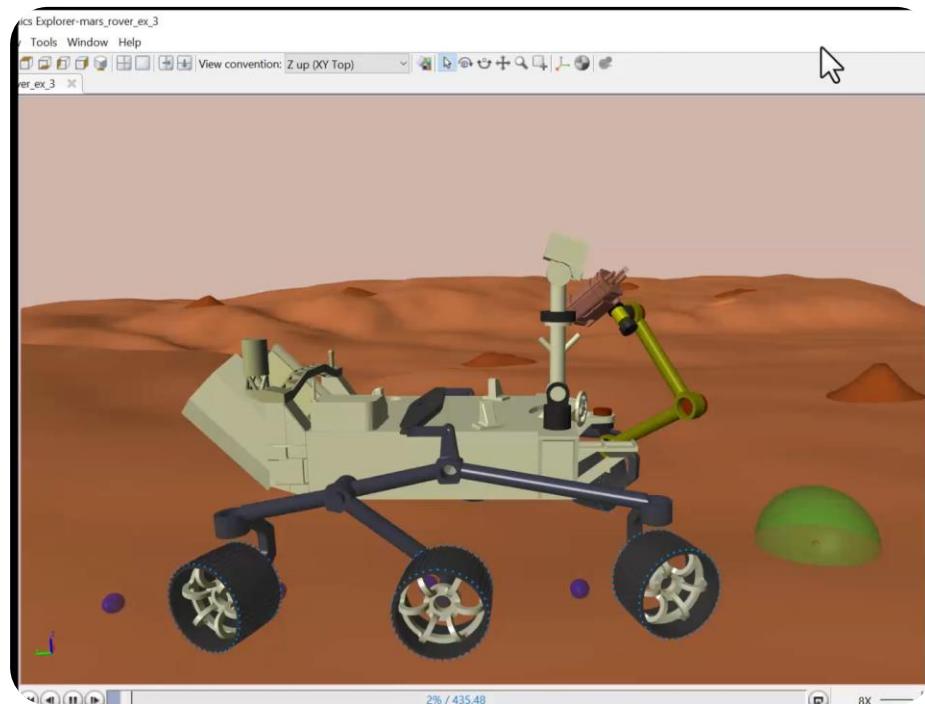


Obstacle Avoidance helps navigate in unknown terrains

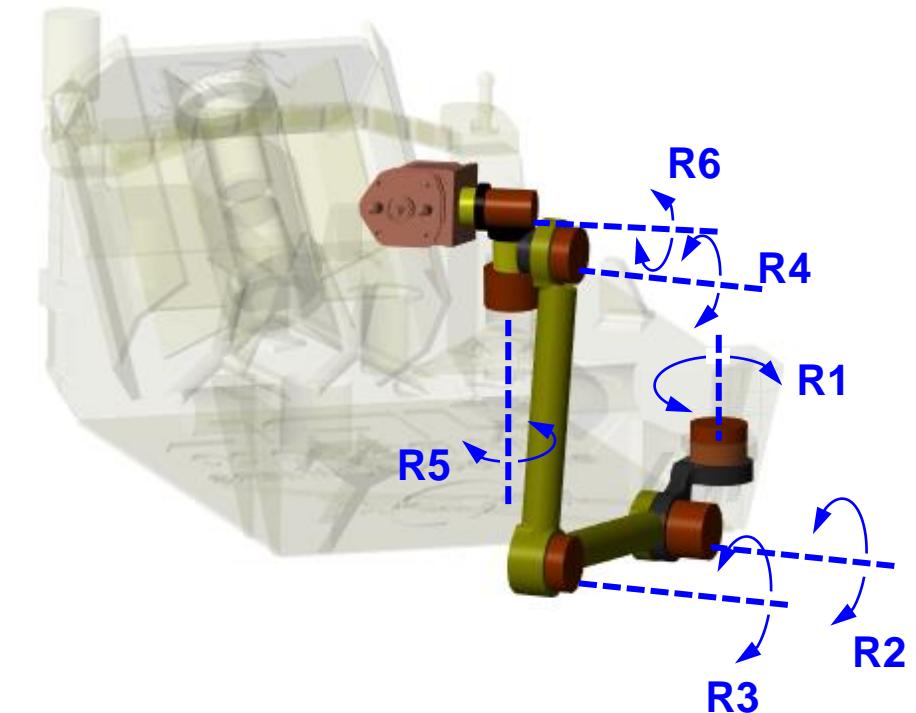


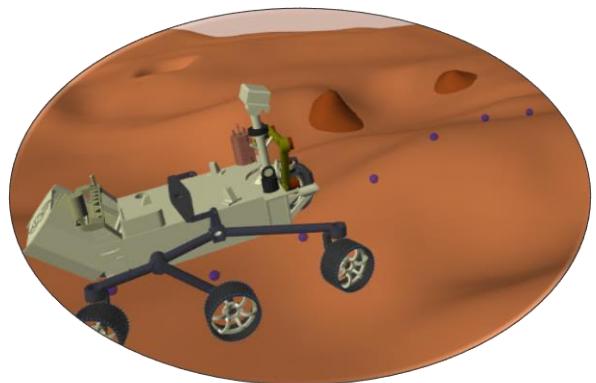
Mars Rover Arm

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



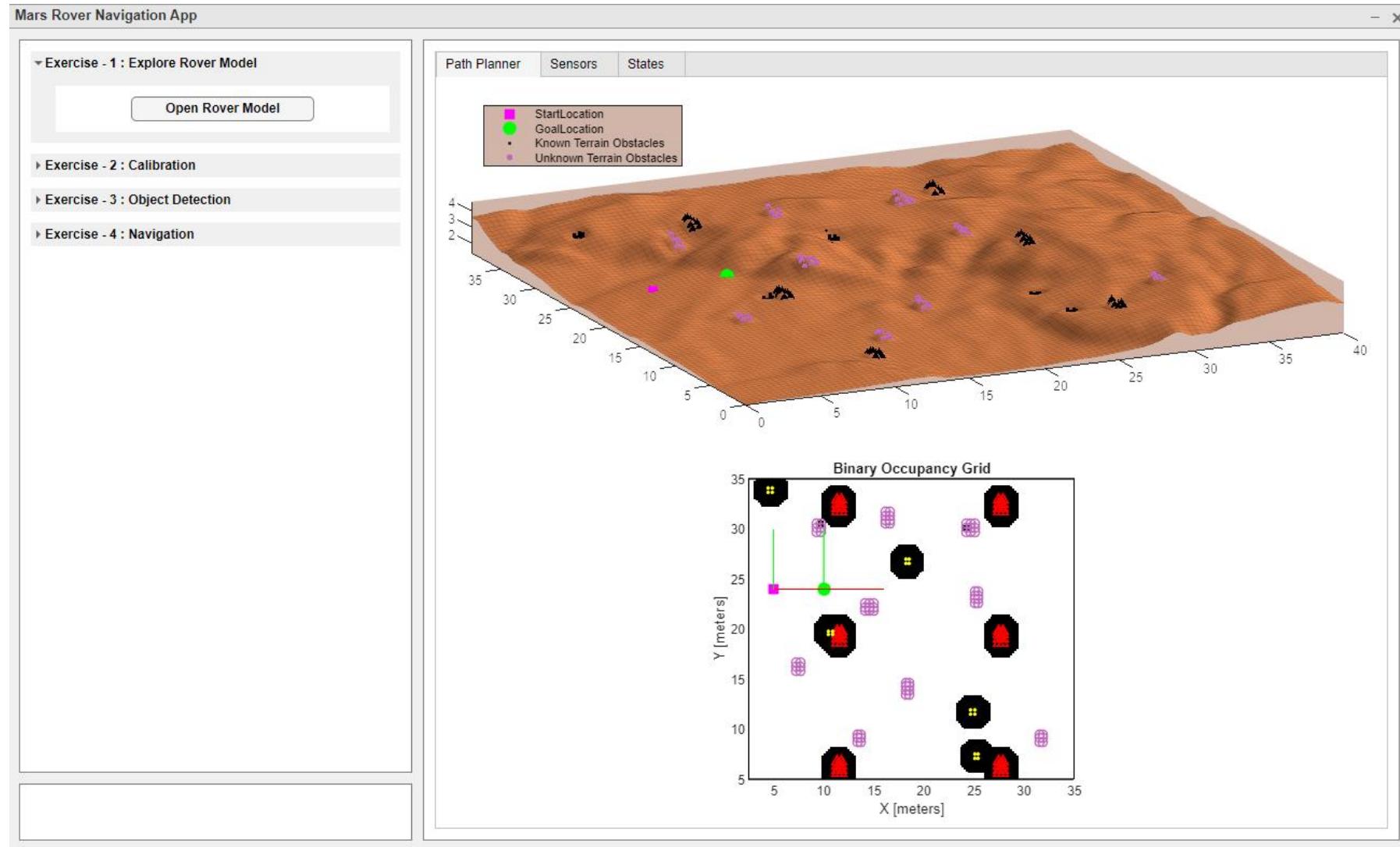
Sample





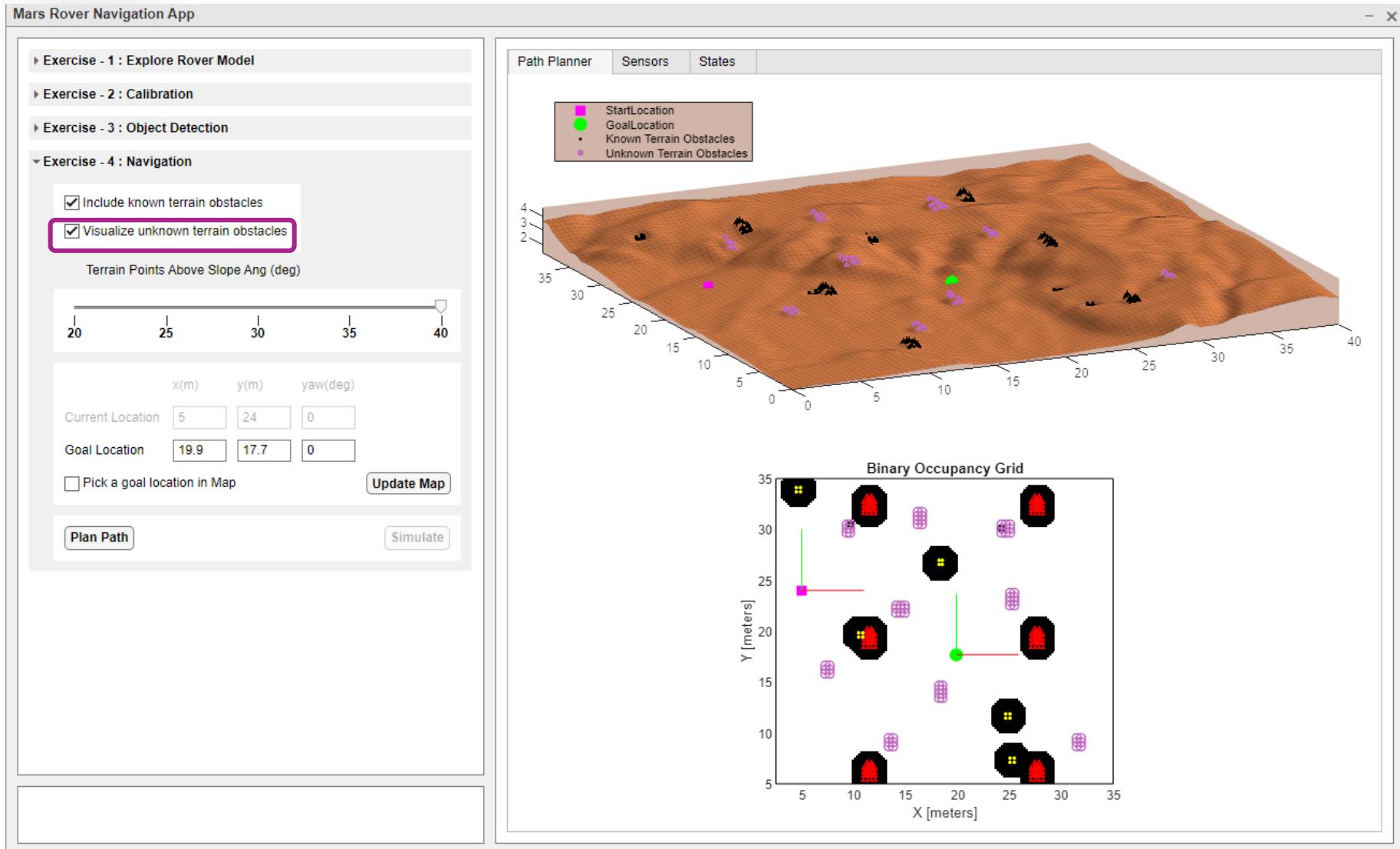
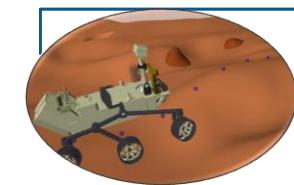
Exercise 4: Navigation

Let's dive into our hands-on exercise



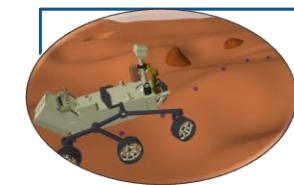
Navigate to a specific goal location

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

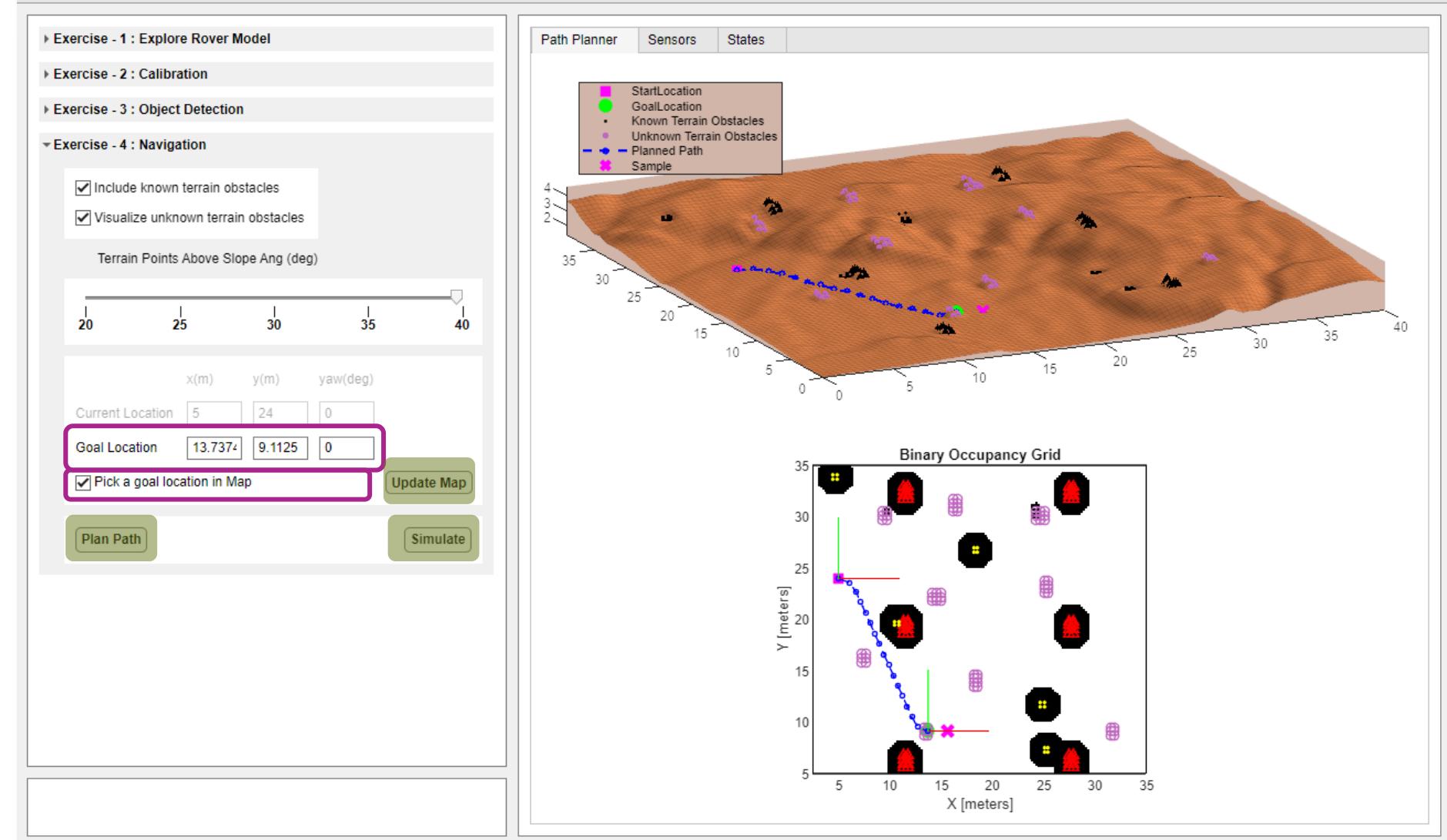


Choose to include/exclude unknown rocks

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

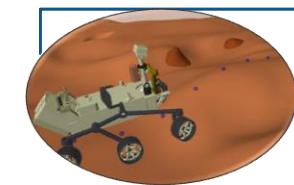


Mars Rover Navigation App



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

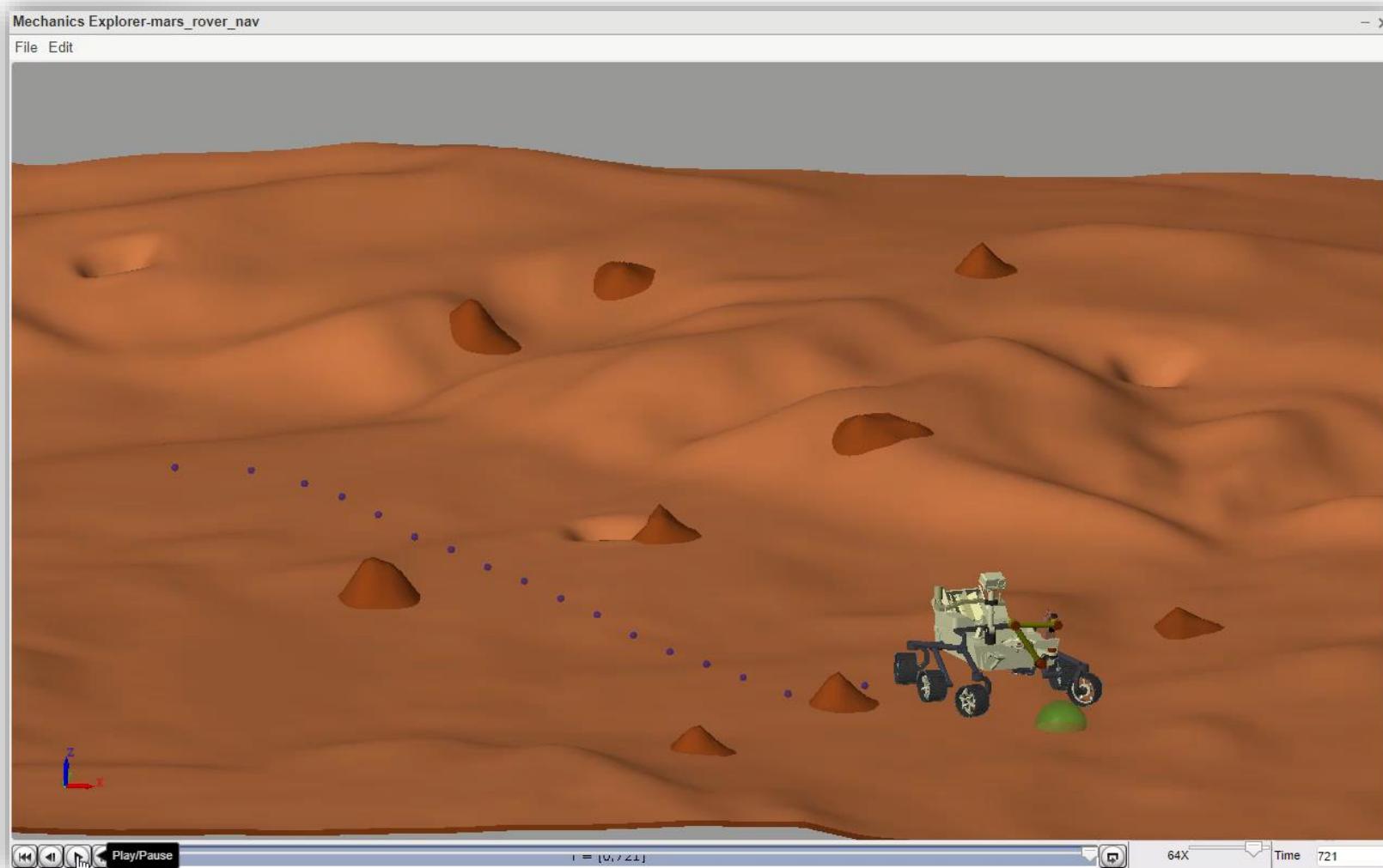
Need a hint? A video showing the solution will play here.

Screenshot of the MATLAB interface during Exercise 4:

- Top Bar:** HOME, PLOTS, APPS, FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, RESOURCES.
- Left Sidebar:**
 - Files:** Shows a folder structure for 'autonomous-mars-rover-main' containing 'autosave', 'Folder', 'm', and 'mat' subfolders, along with several MATLAB files like 'mars_rover_arm_ik.m' and 'MarsRoverNavigationApp_Exercises.m'.
 - Workspace:** Displays variables such as 'ans', 'Arm', 'autoSavedFi...', 'ax', 'blk', 'blk_DL', 'blk_L', 'blk_pose', 'blk_R', 'cameraData', 'd', 'eeConfig', 'Exercise', 'F', 'fig', 'flag_dl', 'flag_explore', 'flag_nav', etc.
- Command Window:** Titled 'Mars Rover Navigation App'. It lists exercises: 'Exercise - 1 : Explore Rover Model', 'Exercise - 2 : Calibration', 'Exercise - 3 : Object Detection', and 'Exercise - 4 : Navigation'. The 'Exercise - 4 : Navigation' tab is selected.
- 3D Plot:** Shows a 3D terrain model of Mars with a grid. It includes a legend for 'StartLocation' (pink square), 'GoalLocation' (green circle), 'Known Terrain Obstacles' (black dots), and 'Unknown Terrain Obstacles' (purple dots).
- Binary Occupancy Grid:** A 2D plot titled 'Binary Occupancy Grid' showing the same terrain as the 3D plot, but with binary obstacle representation. It shows the start location at (5, 25) and goal location at (20, 18).
- Timer:** A black box in the bottom right corner displays '6 m 00 s'.

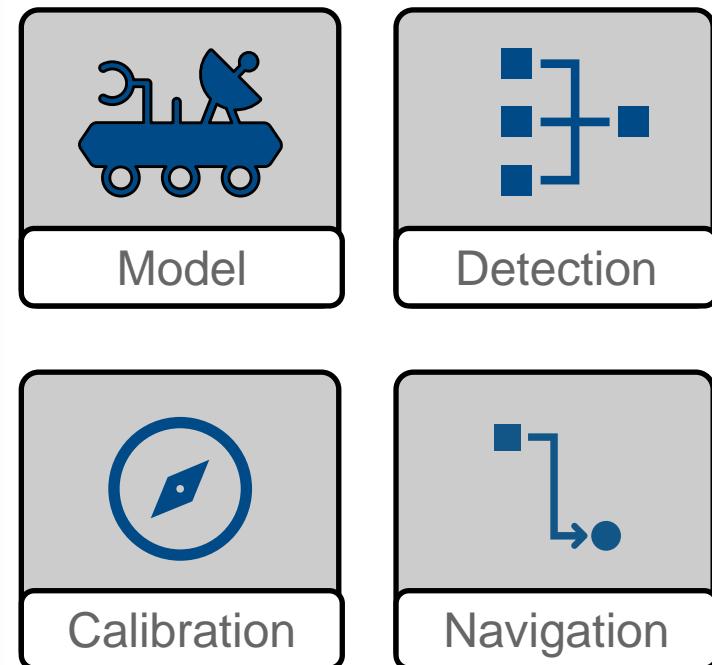
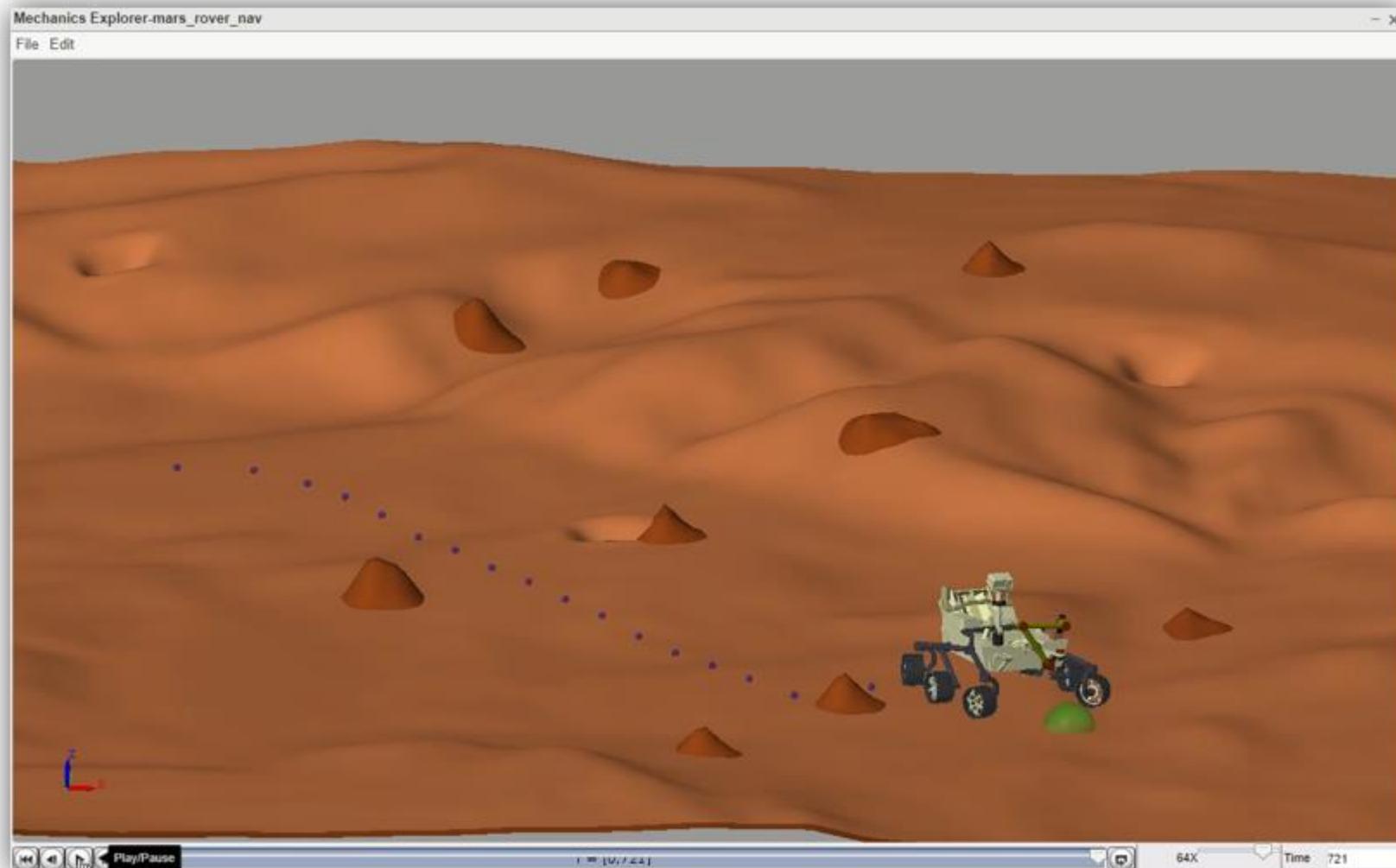


How did your rover perform?

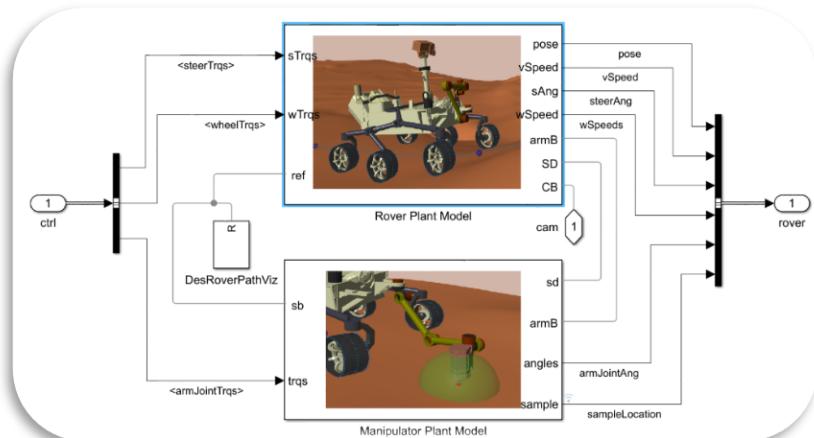


- What objects did you detect?
- Did you avoid obstacles?
- Did rover arm pick up a sample ?

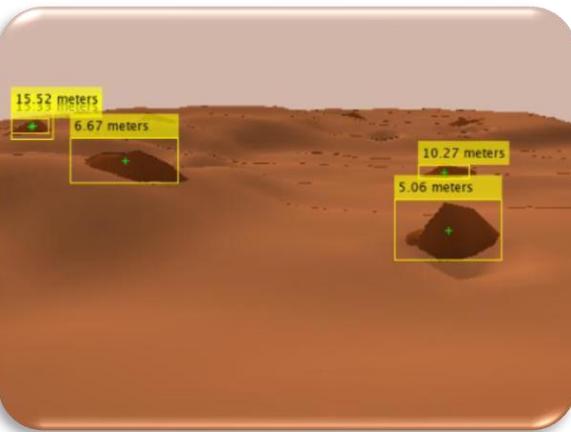
Congratulations ! You've just planned an effective mission using simulation.



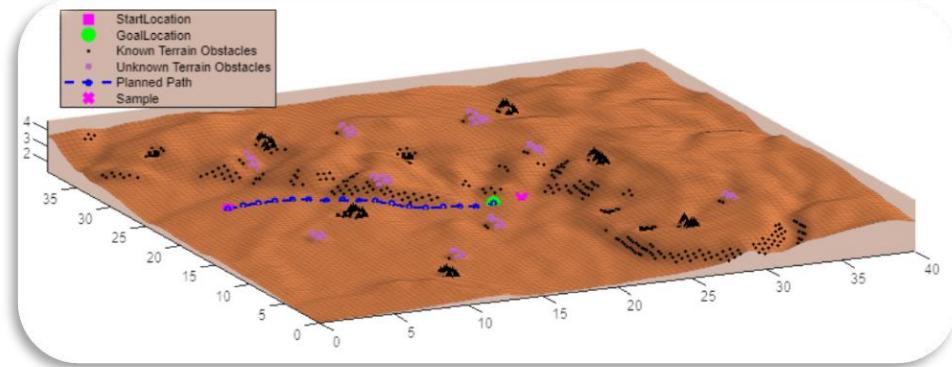
Key takeaways



Physical Modeling
(Rover Model)



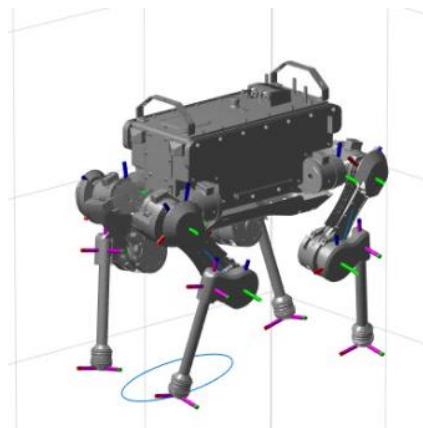
Deep learning for
object detection



Path planning in
autonomous systems

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

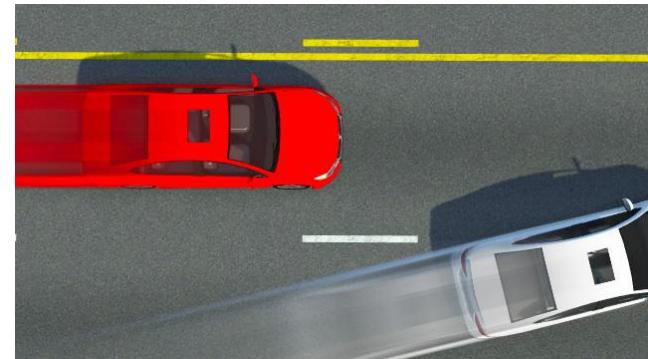
Design complex physical systems



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>

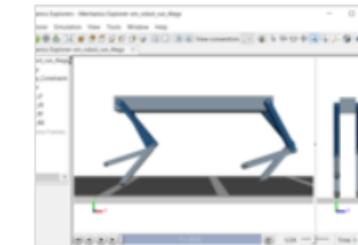
Deep Learning for Image Analysis

<https://www.mathworks.com/solutions/deep-learning.html>

Find More Examples on File Exchange

<https://www.mathworks.com/matlabcentral/fileexchange/?q=Simscape&sort=relevancy>

File Exchange



Running Robot Model in Simscape version 21.2.2

Quadruped robot model with electrical actuation, including

: <https://github.com/mathworks/Simscape-Robot-4Legs/a>
<https://github.com/mathworks/Simscape-Robot-4Legs/ar>

 Using the **Simscape Multibody Contact Forces Library**



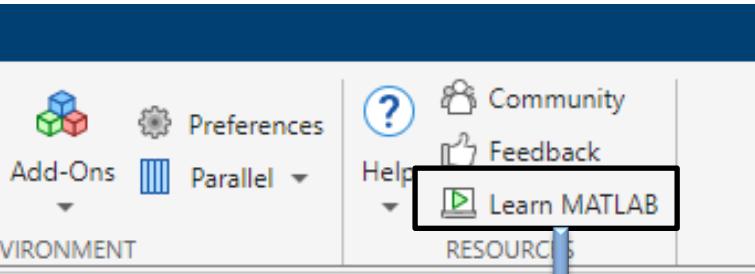
Mars Rover Model in Simscape version 21.2.1.0 by

Six-wheeled electrically-driven rover that navigates uneven

Mars Rover Model in Simscape™ Copyright 2021-2022 The

Collection

Take the Next Step



Self-Paced Online Courses

Home | My Courses

- Getting Started (16)
- MATLAB (5)
- Simulink (7)**
- AI, Machine Learning, and Deep Learning (6)
- Math and Optimization (6)
- Image and Signal Processing (6)
- Explore over 50 virtual and in-person [classroom courses](#)

Simulink

Simulink Onramp
14 modules | 2 hours | Languages
Get started quickly with the basics of Simulink.

Simulink Fundamentals
10 modules | 8 hours | Languages
Learn how to use Simulink, a graphical simulation tool for modeling dynamic physical systems.

Circuit Simulation Onramp
7 modules | 2 hours | Languages
Learn the basics of simulating electrical circuits in Simscape.

Simscape Onramp
9 modules | 1.5 hours | Languages
Learn the basics of simulating physical systems in Simscape.

 MathWorks® Products Solutions Academia Support Community Events

MATLAB & Simulink

Access MATLAB for your Hands on Workshop

MathWorks is pleased to provide a special license to you as a course participant to use for your Hands on Workshop. This is a limited license for the duration of your course and is intended to be used only for course work and not for government, research, commercial, or other organization use.

Course Name:	Model Autonomous Navigation of a Mars Rover
Organization:	MathWorks
Starting:	10 May 2023
Ending:	11 May 2023

[Access MATLAB Online](#)

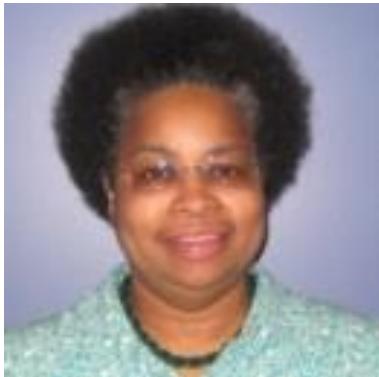
<https://tinyurl.com/ExpoMarsRover>

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



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc.
See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be
trademarks or registered trademarks of their respective holders.

