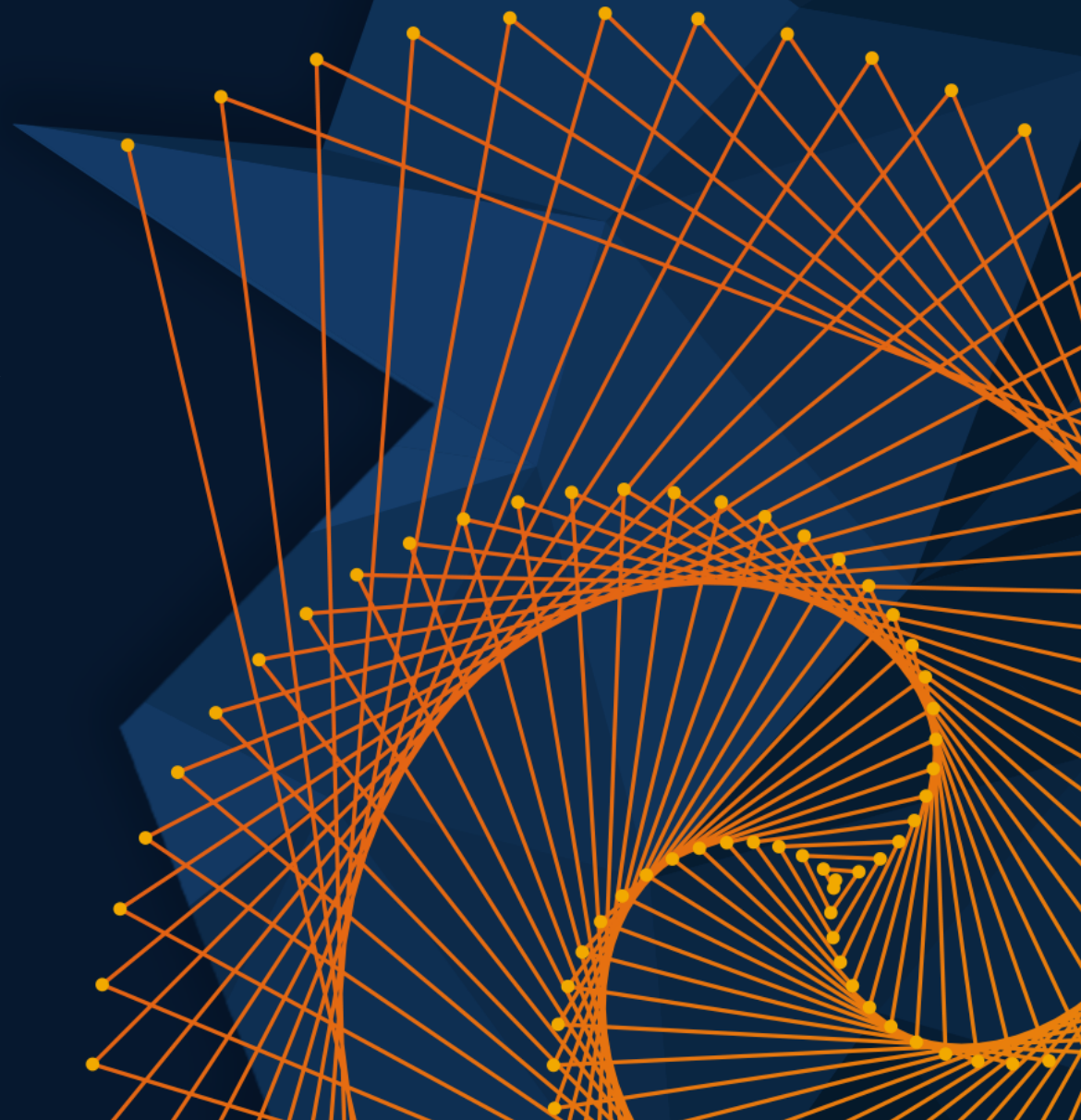
MATLAB EXPO

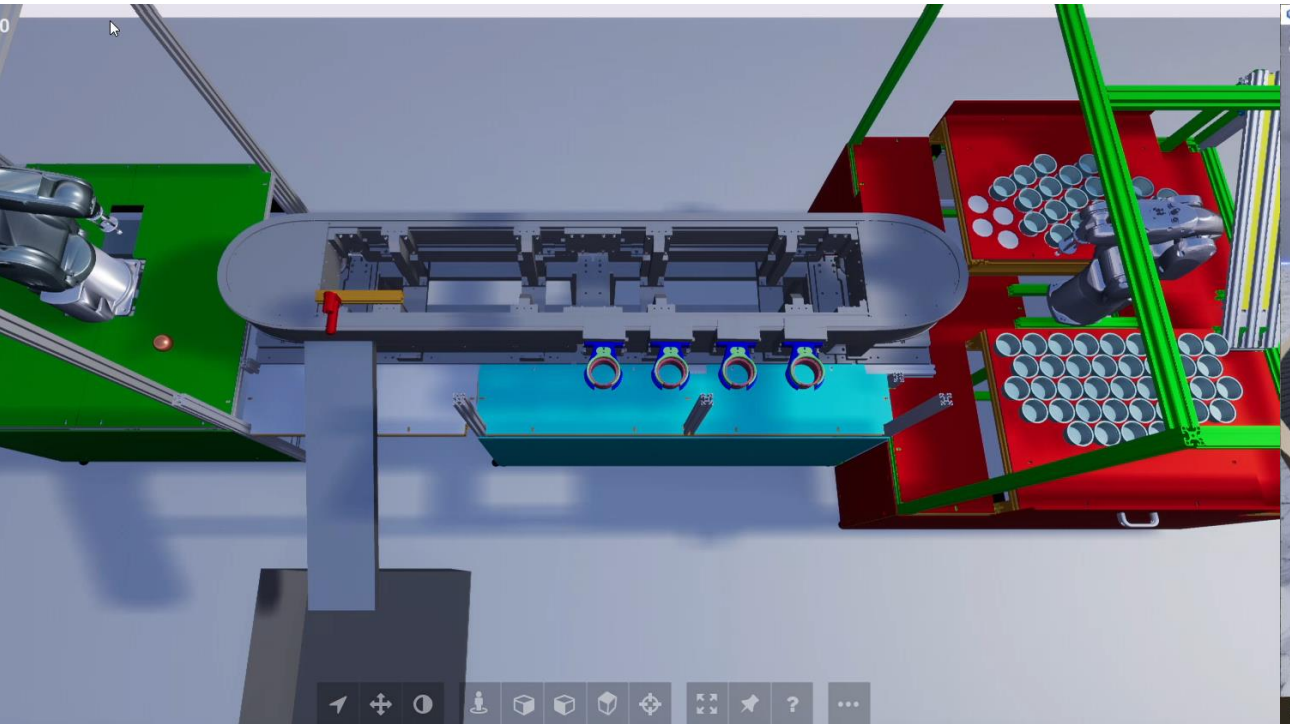
MATLAB/Simulink를 이용한 손쉬운 Unreal 3차원 시뮬레이션 환경 구축

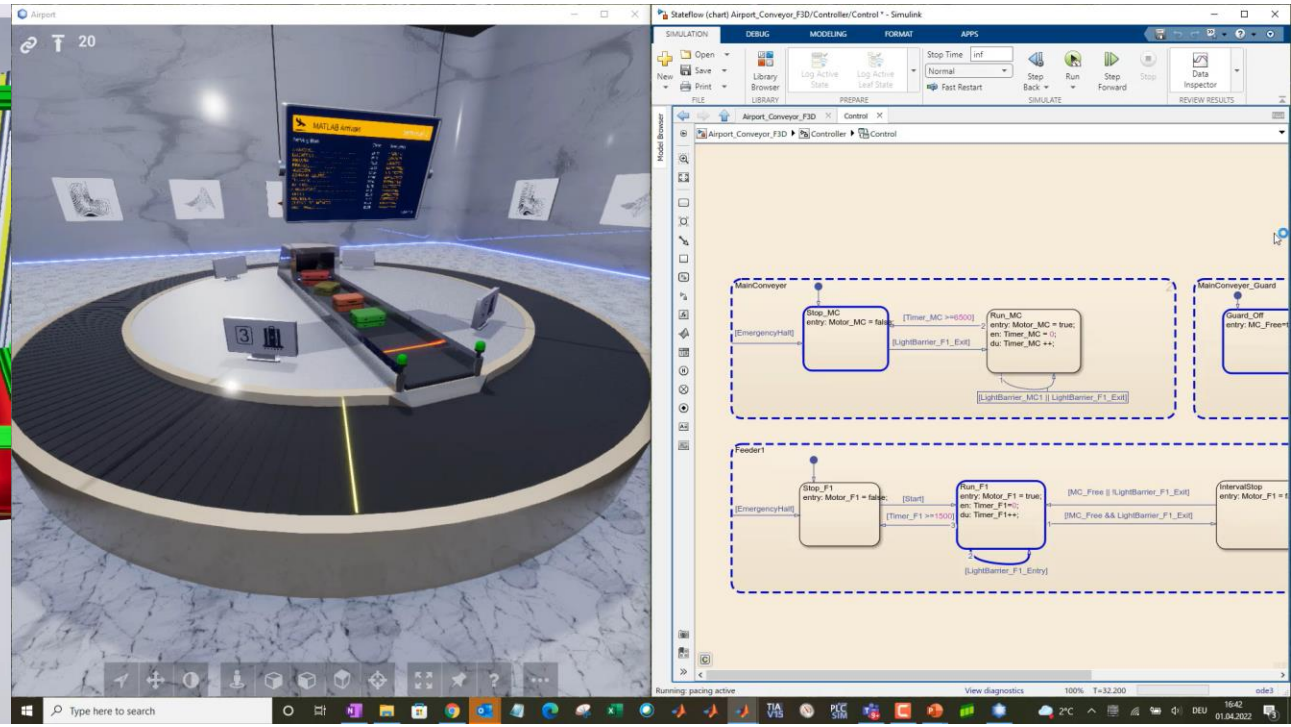


MathWorks®

Industrial Automation & Machinery

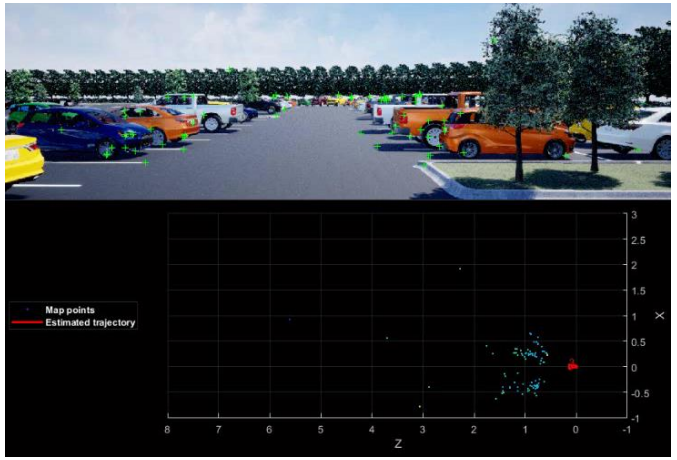
Manufacturing, Virtual Commissioning

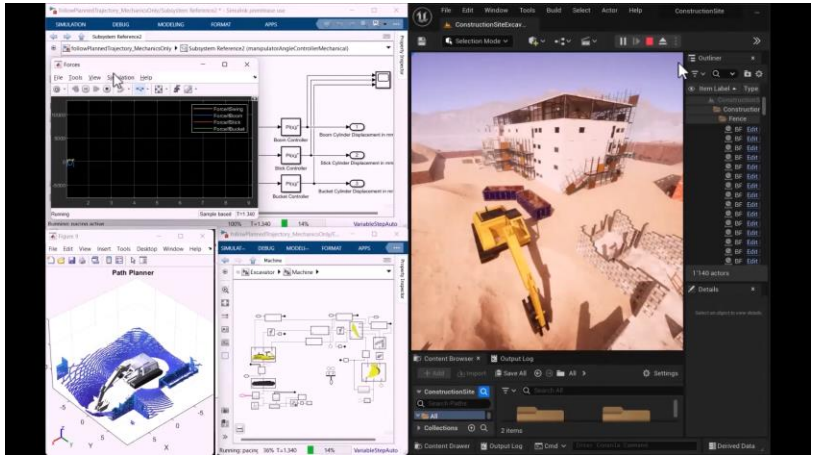


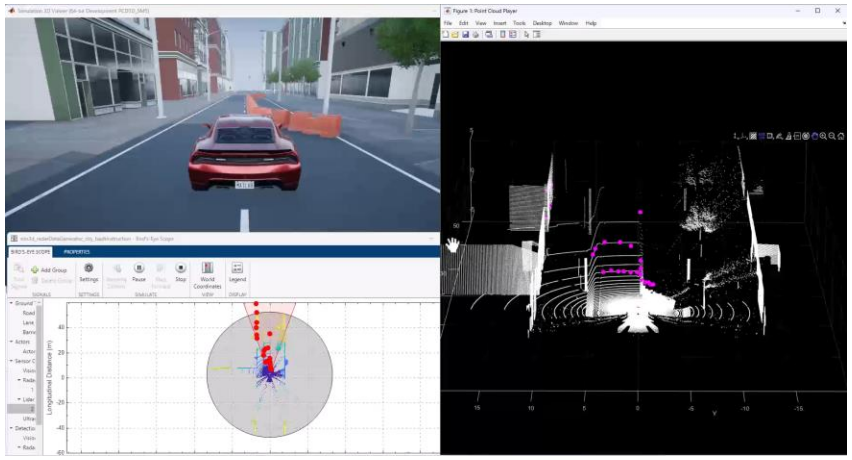


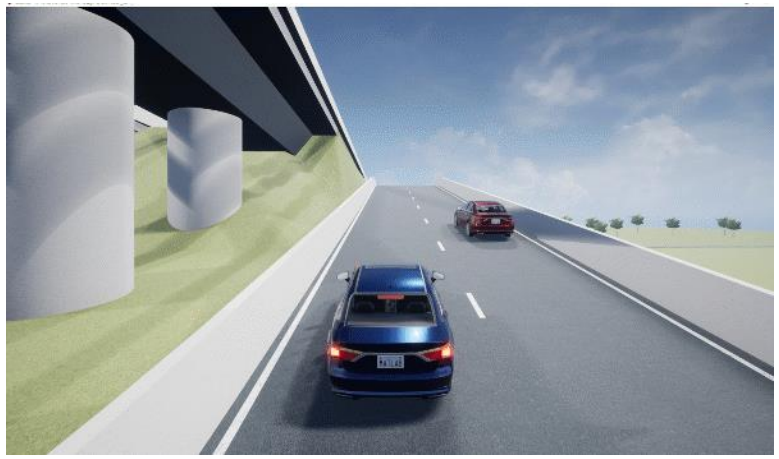
Automotive

Sensing, SLAM, Sensor Fusion, Dynamics and Off-Highway Vehicle



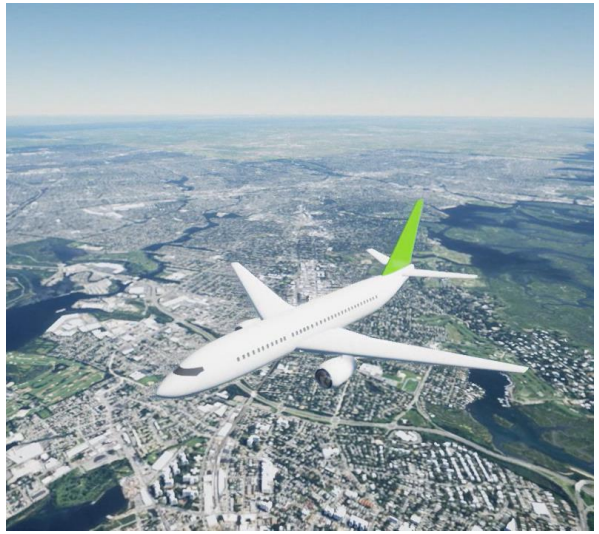


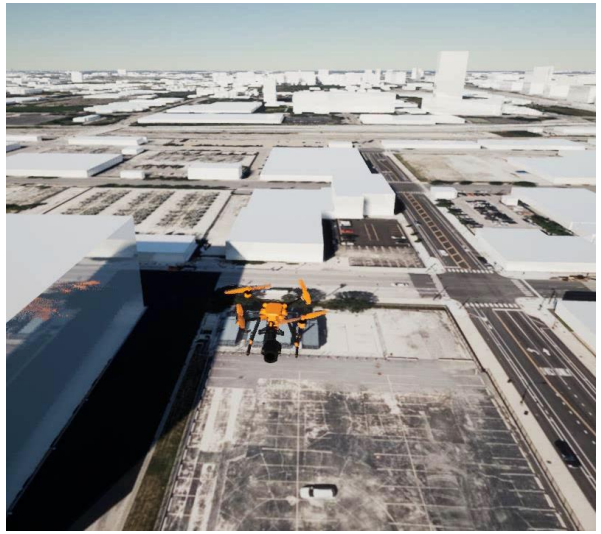


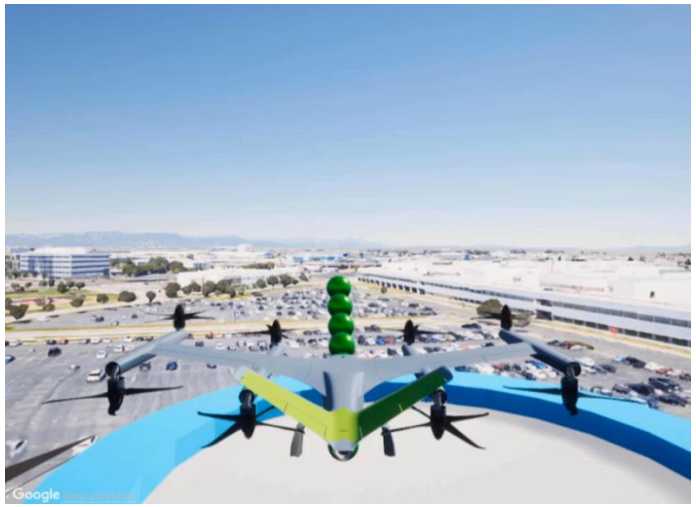


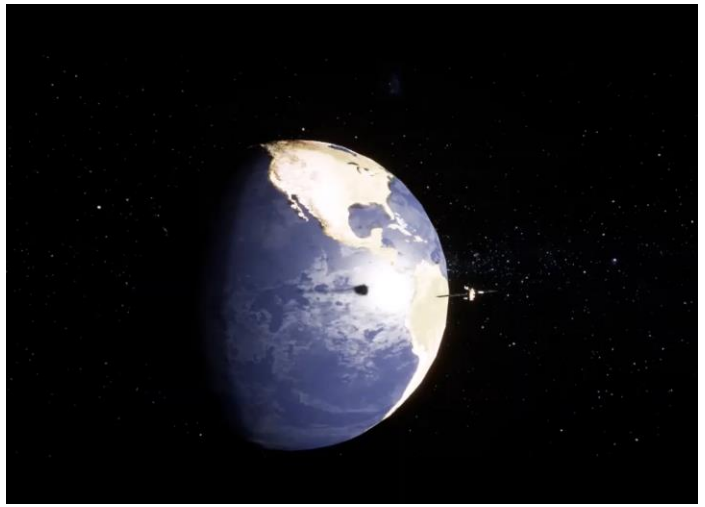
Aerospace & Defense

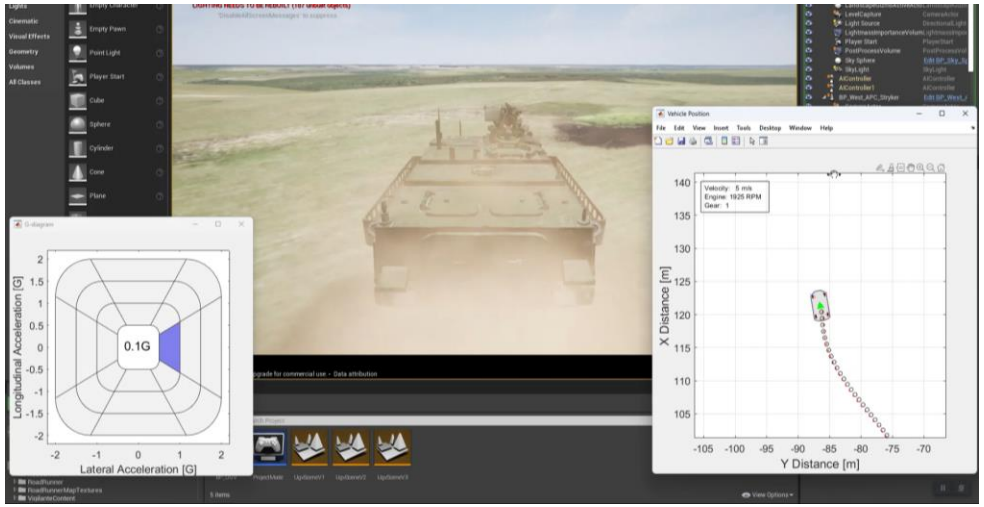
Airplane, Drone, UAM, Satellite, Jet Fighter & UGV





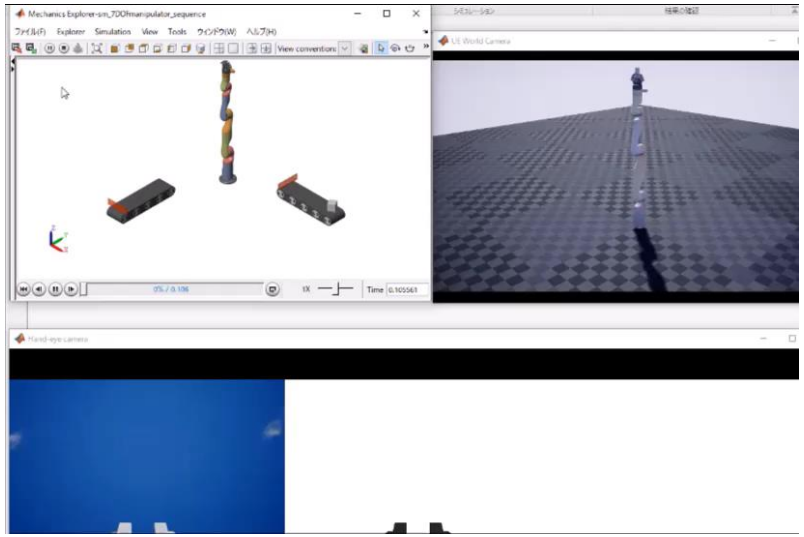


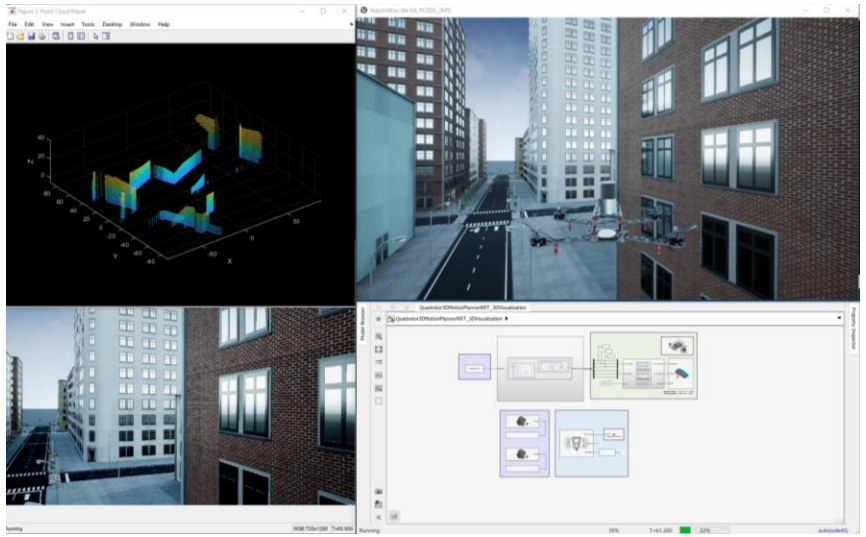


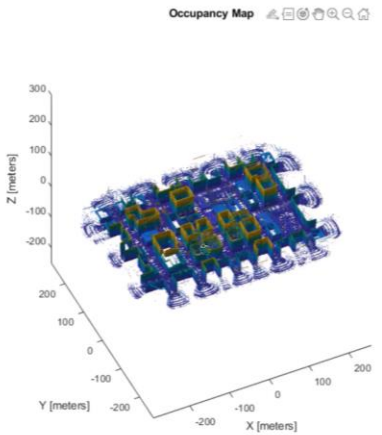


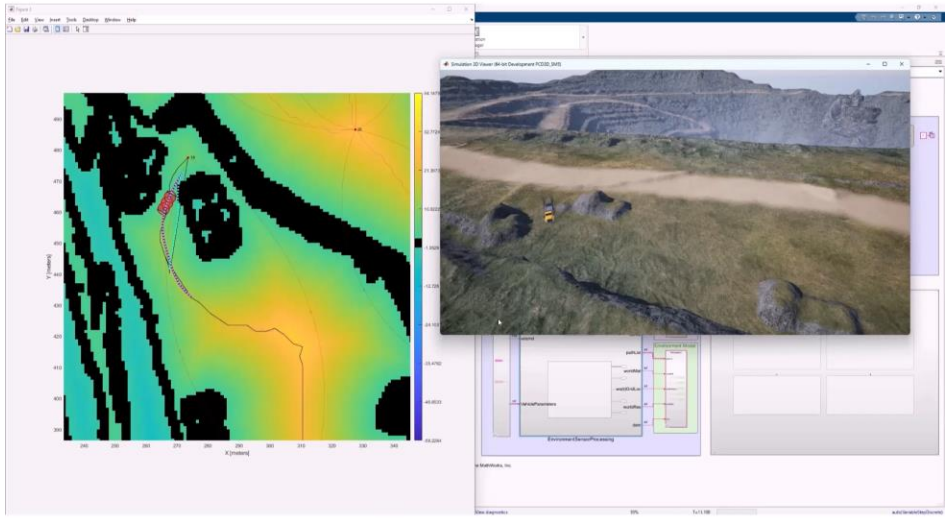
Robotics

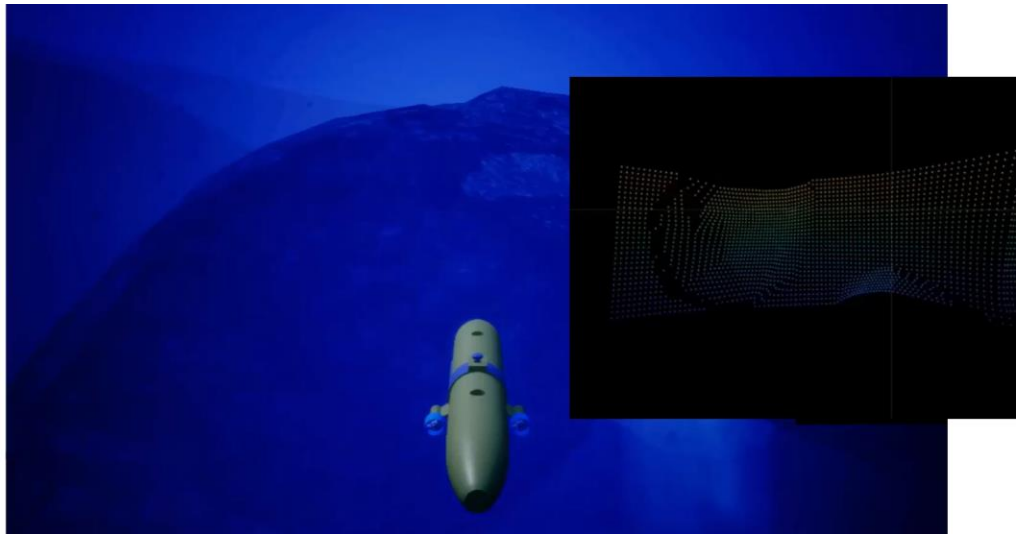
Manipulators, UAV, Mobile Robots & Marine Robots











Simulink 3D Animation

Perform closed-loop, deterministic simulations with Unreal Engine

MATLAB & Simulink

- **Physics of main actor(s)**
- **Perception, Planning, Control Algorithms**

actor position

sensor location

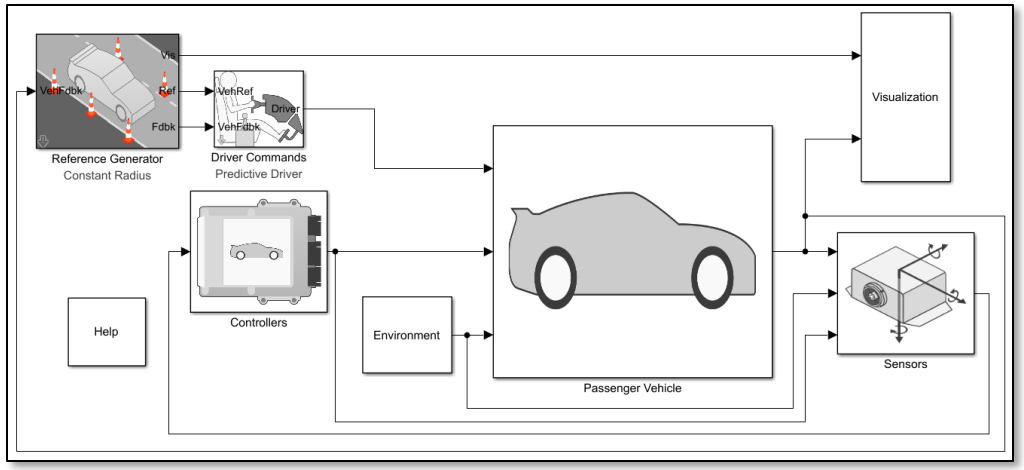
sensor information

collision information

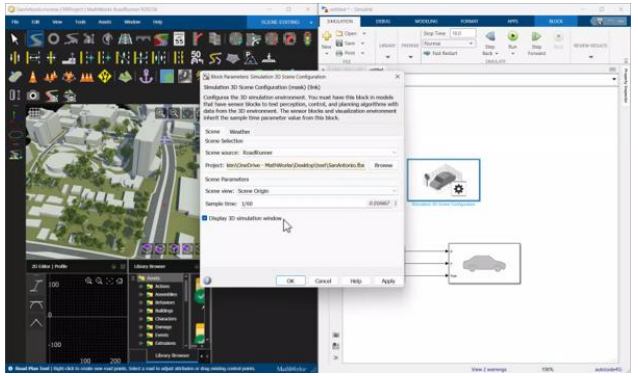
Unreal Engine

- **Rendering / lighting**
- **Physics of non-Simulink actors**
- **Collision detection**

Lock-step co-simulation (solvers take turns) provides **deterministic results**



Benefits of Simulink 3D Animation

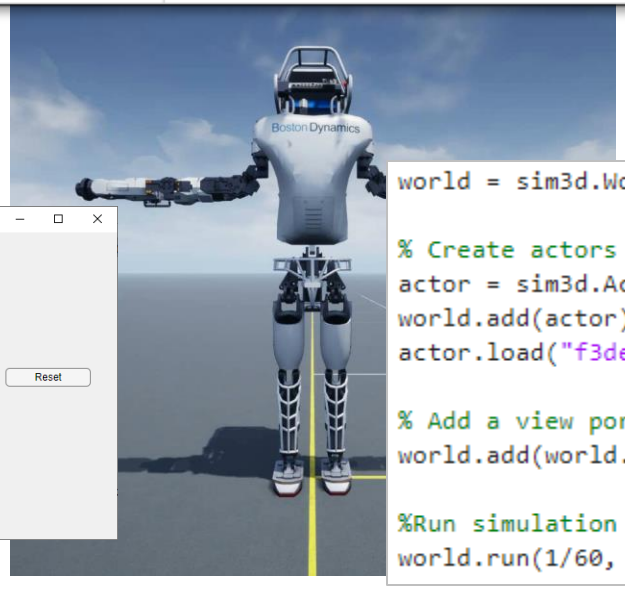


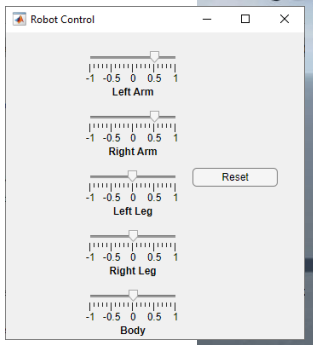
- Easy workflow to build a custom simulation scene and connect with MATLAB/Simulink
  - Unreal Editor
  - **RoadRunner scene import** in Sim3d Configuration Block
  - Programmatic creation using **MATLAB API**

MATLAB API and Simulink Blocks to communicate with Unreal Engine

MATLAB API
run-time and edit-time import of 3D content

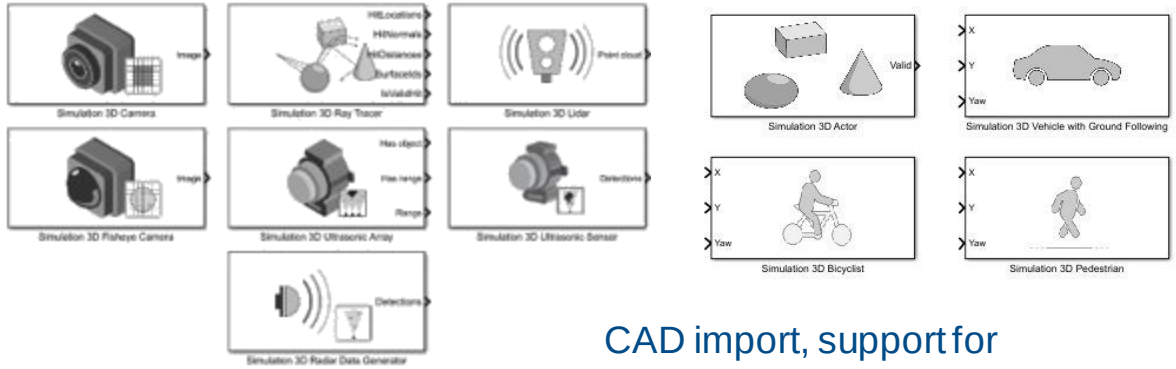
copy	Copy all properties from another actor
propagate	Propagate value of selected property to actor and its children
gather	Return values of selected property from all objects in selected branch
findBy	Find all actors that match specified criteria
restoreSnapshot	Restore actor to state of properties saved in specified snapshot
takeSnapshot	Take snapshot of selected properties
createMesh	Create new mesh with specified values
addMesh	Add mesh on top of current mesh
load	Load or import 3D file
save	Save actor and children to a MAT file
createShape	Create geometry for basic primitives



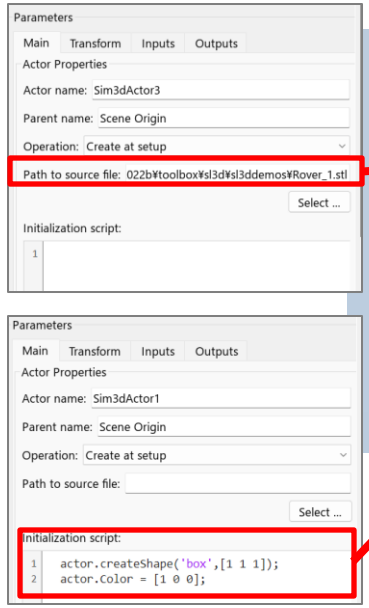


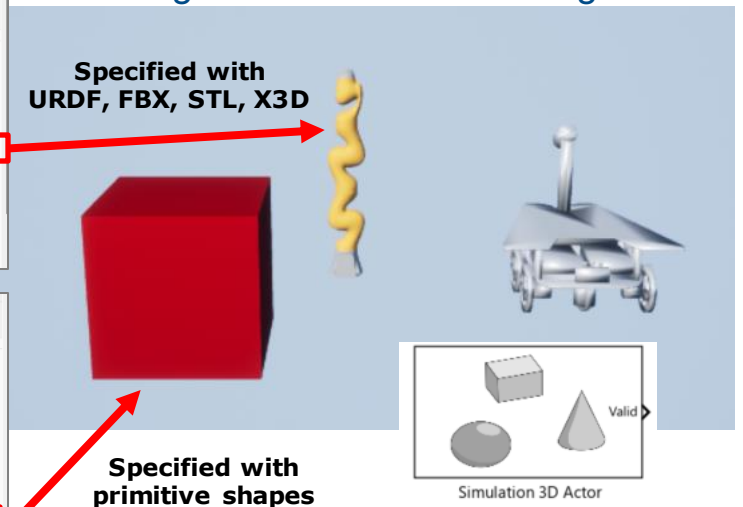

```
world = sim3d.World();

% Create actors in scene
actor = sim3d.Actor();
world.add(actor);
actor.load("f3dex_atlas.f3d");

% Add a view port
world.add(world.Viewports.Main);

%Run simulation
world.run(1/60, 60);
```

Simulink blocks
connecting to Unreal Engine co-simulation



CAD import, support for algorithmic scene authoring





Unreal Engine Co-Simulation Workflow

RoadRunner

1 RoadRunner Scene creation

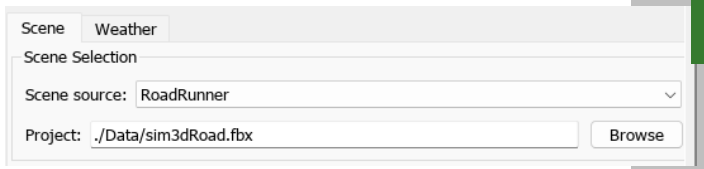
2 3D Scenes export (.FBX, .XML)

Unreal Engine

3 Import to UE Build a scene in UE

Need to install Unreal Editor and RoadRunner plugin

RoadRunner scene *direct import* in Simulation 3D Configuration Block



MATLAB & Simulink

4 Co-works with UE

Need to install Automated Driving Toolbox Interface for Unreal Engine Projects

Simulink model

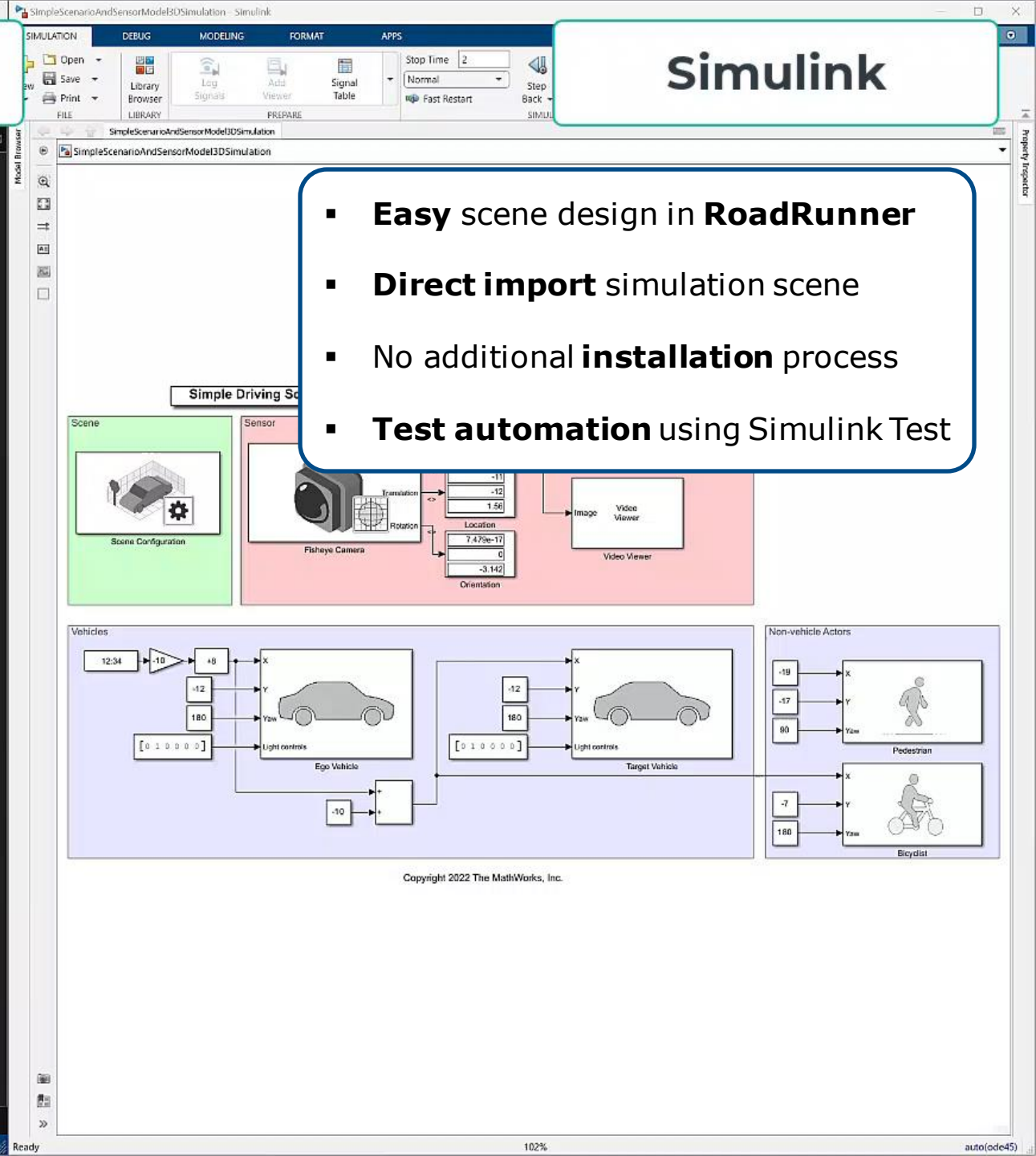
3 Co-works with UE

Without installation

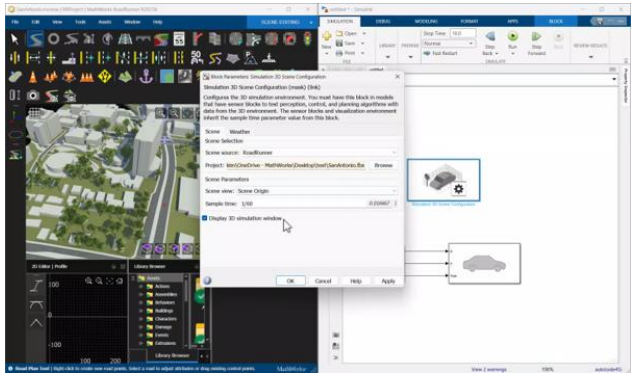
RoadRunner

Simulink

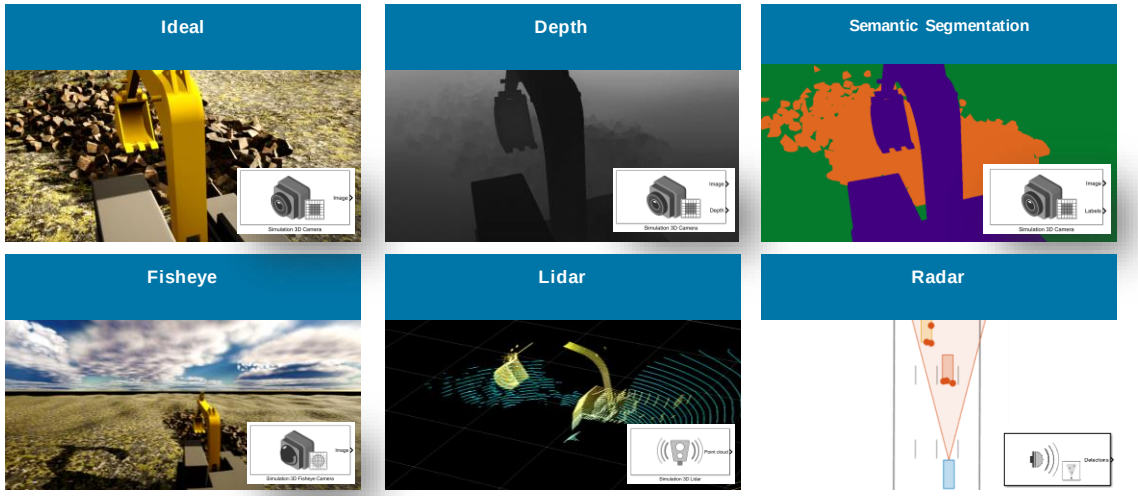
- **Easy** scene design in **RoadRunner**
- **Direct import** simulation scene
- No additional **installation** process
- **Test automation** using Simulink Test

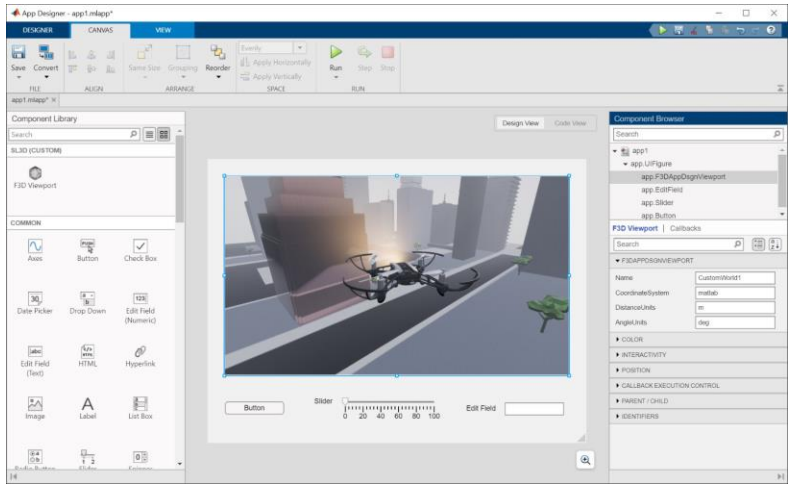


Benefits of Simulink 3D Animation

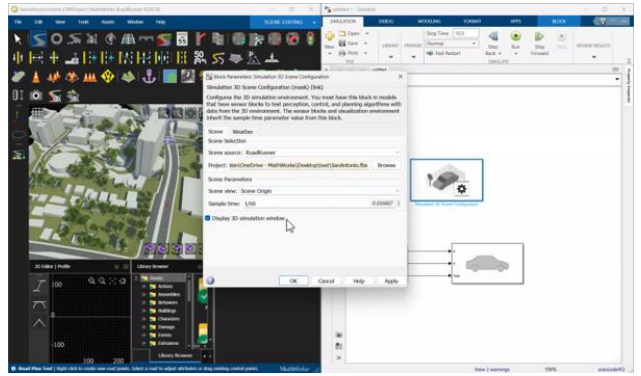


- Easy workflow to build a custom simulation scene and connect with MATLAB/Simulink
  - Unreal Editor
  - Programmatic creation using **MATLAB API**
  - **RoadRunner scene import** in Sim3d Configuration Block
- Available to integrate a Simulink/Simscape model with Unreal Engine for Visualization or Sensor simulation
  - Virtual cameras, LIDARs, depth sensors
  - App designer support

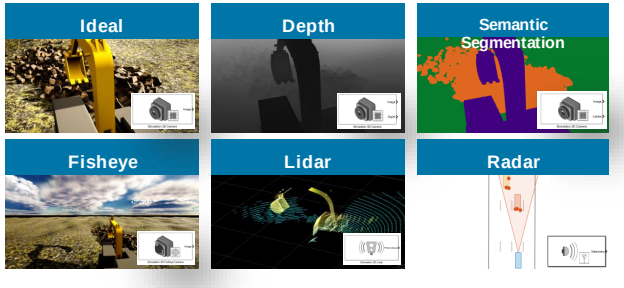


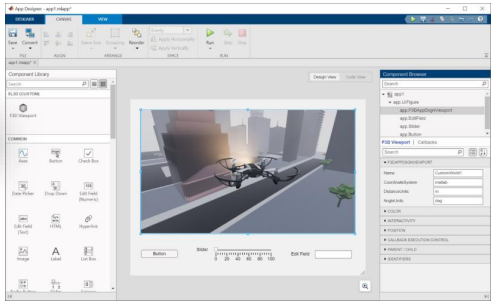


Benefits of Simulink 3D Animation

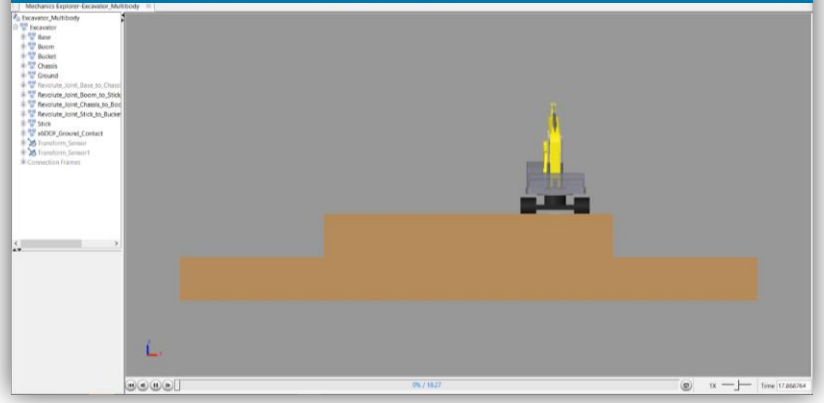


- Easy workflow to build a custom simulation scene and connect with MATLAB/Simulink
  - Unreal Editor
  - Programmatic creation using **MATLAB API**
  - **RoadRunner scene import** in Sim3d Configuration Block
- Available to integrate a Simulink/Simscape model with Unreal Engine for Visualization or Sensor simulation
  - Virtual cameras, LIDARs, depth sensors
  - App designer support



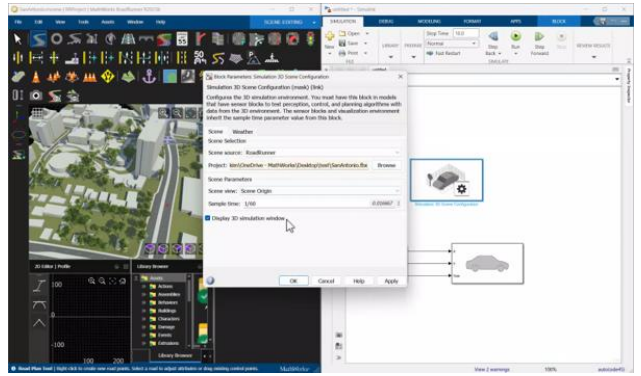


Simscape Multibody

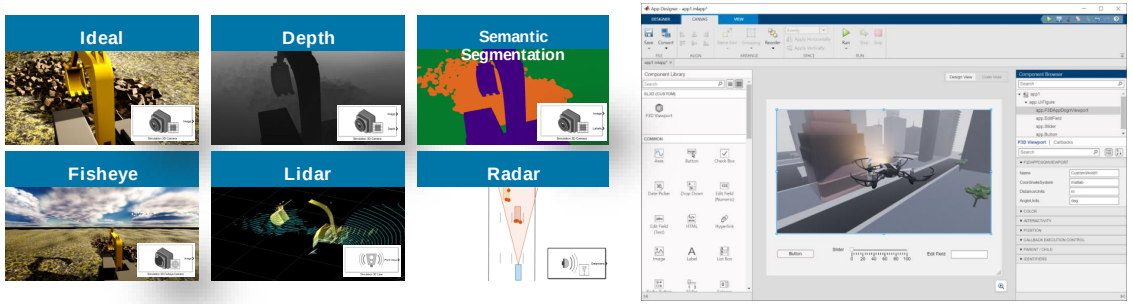


Unreal Engine

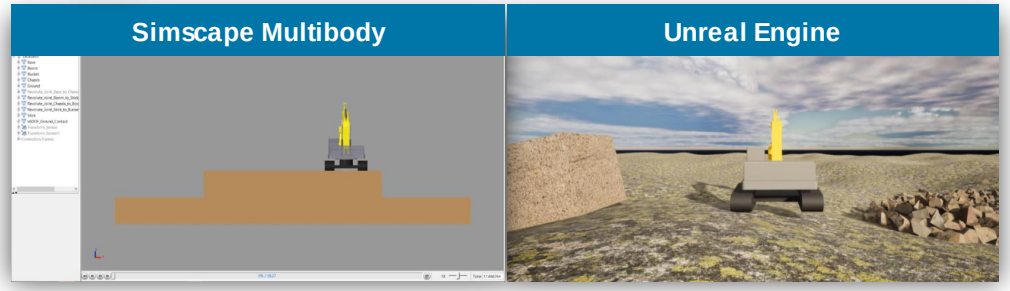
Benefits of Simulink 3D Animation



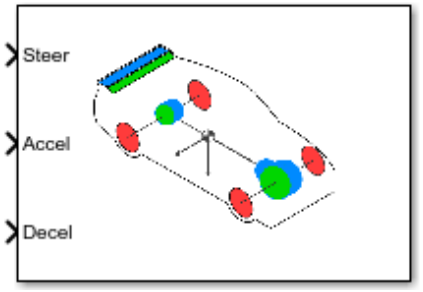
- Easy workflow to build a custom simulation scene and connect with MATLAB/Simulink
  - Unreal Editor
  - Programmatic creation using **MATLAB API**
  - **RoadRunner scene import** in Sim3d Configuration Block

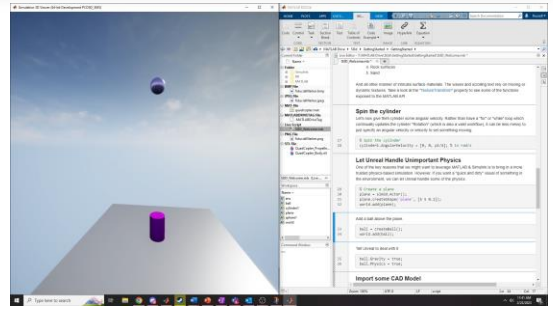


- Available to integrate a Simulink/Simscape model with Unreal Engine for Visualization or Sensor simulation
  - Virtual cameras, LIDARs, depth sensors
  - App designer support



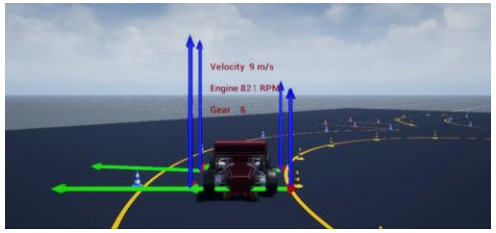
- Unreal Engine physics with a MATLAB/Simulink actor
  - **Physics Vehicle**:
    evaluate position, velocity, rotation, and force feedback of a vehicle actor using Chaos physics
  - Enabling **physics and gravity** of actor:
    position, velocity, etc determined by **Chaos physics**
    (collision, fluid dynamics, soft body dynamics)





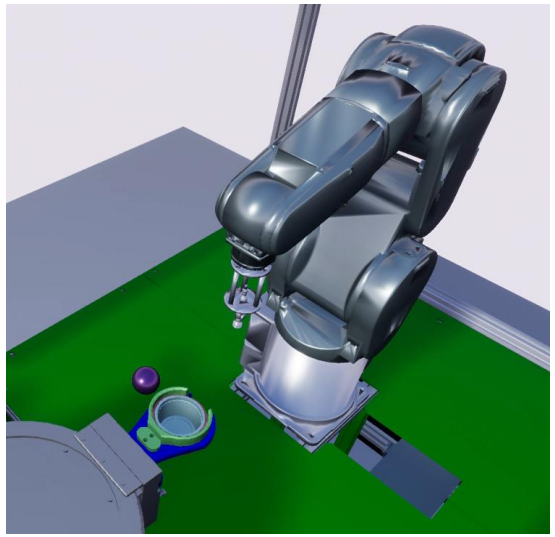
Improvements of Simulink 3D Animation in R2024a

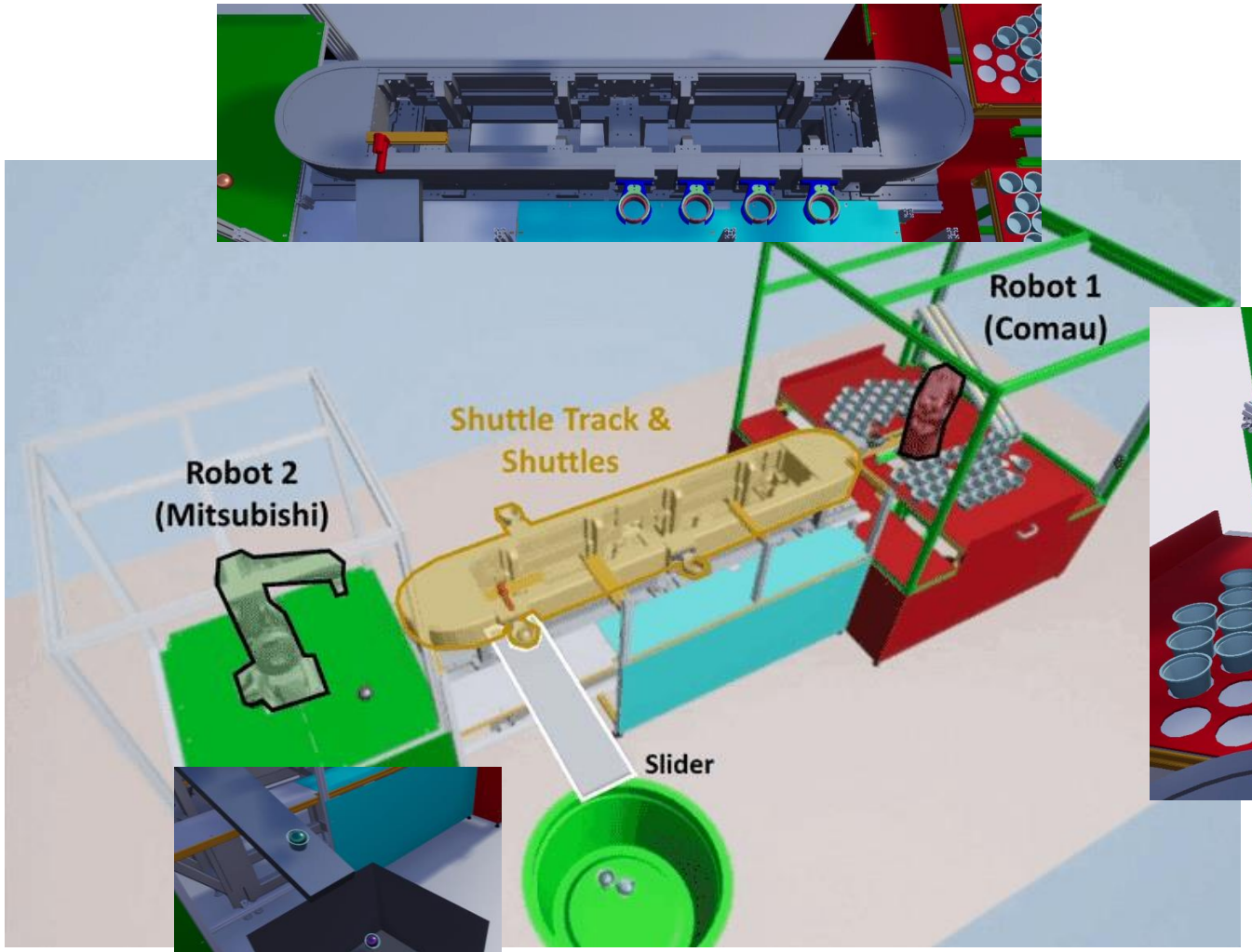
- Unreal Engine 5
  - Efficient rendering and memory usage
- Support for Linux platform
- Weather system
  - Rain particles now interact with camera lens
  - Snow
  - Volumetric clouds using 3D volume textures
  - Geo-location and time-of-day based sun configuration
- 3D Light Simulation
  - Create and add different types of lights
  - Create light array with single Light Actor
- 3D Annotation
  - Create multiple arrows and texts with vectorized inputs

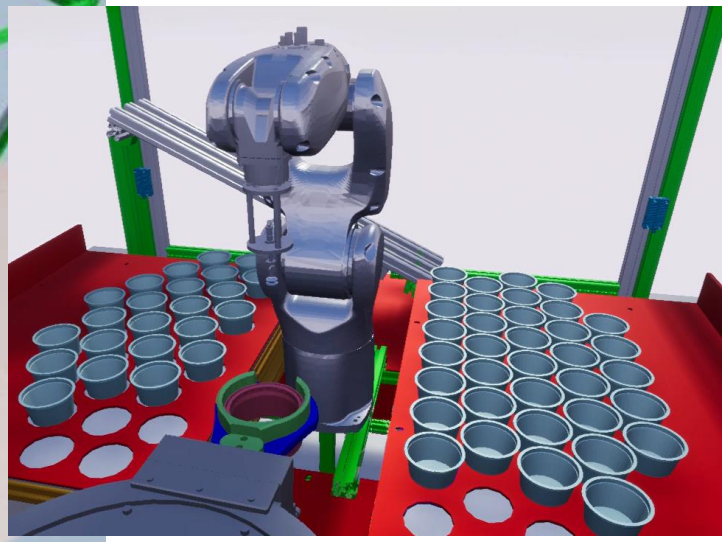


Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

System Configuration

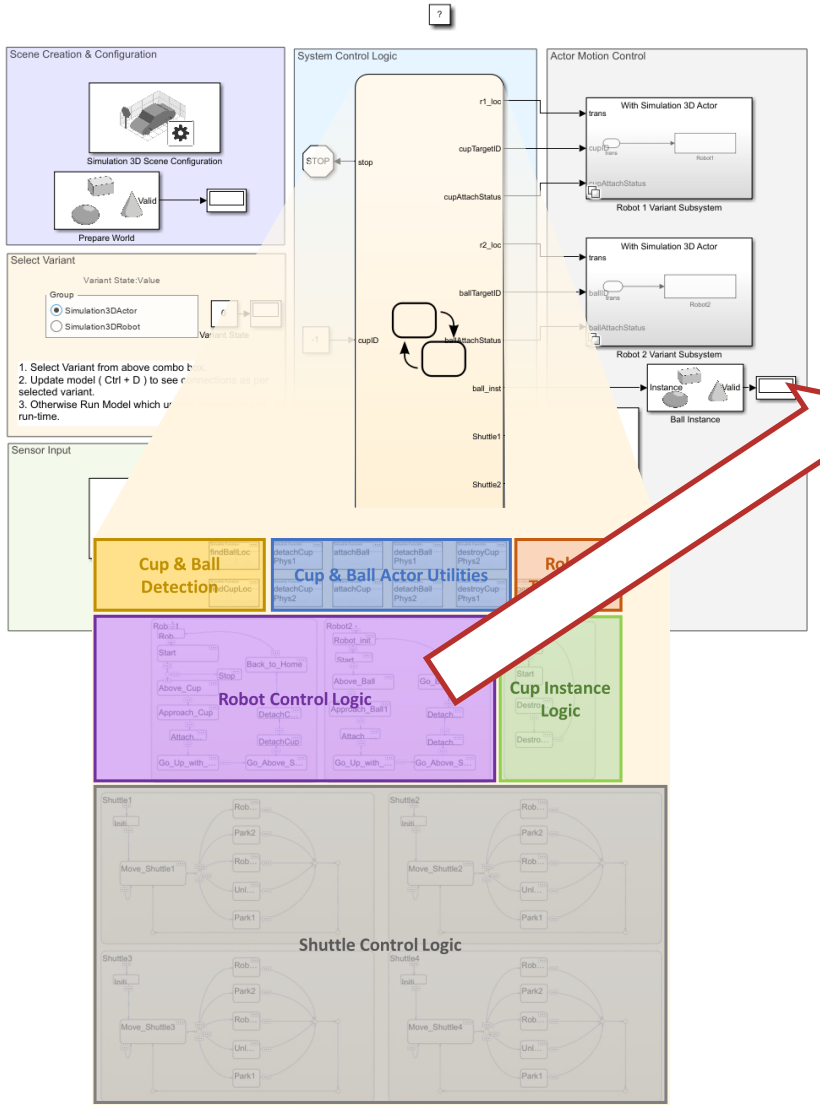


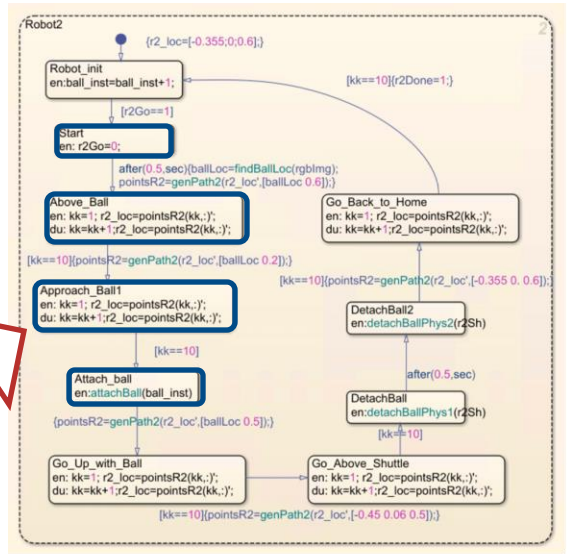


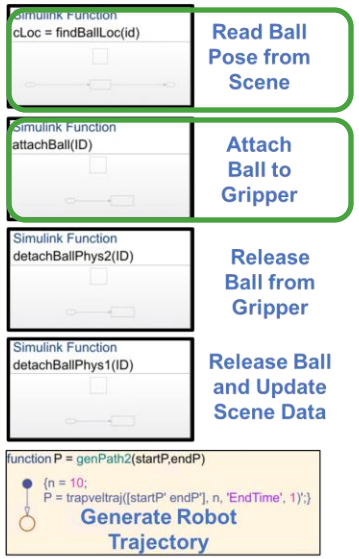


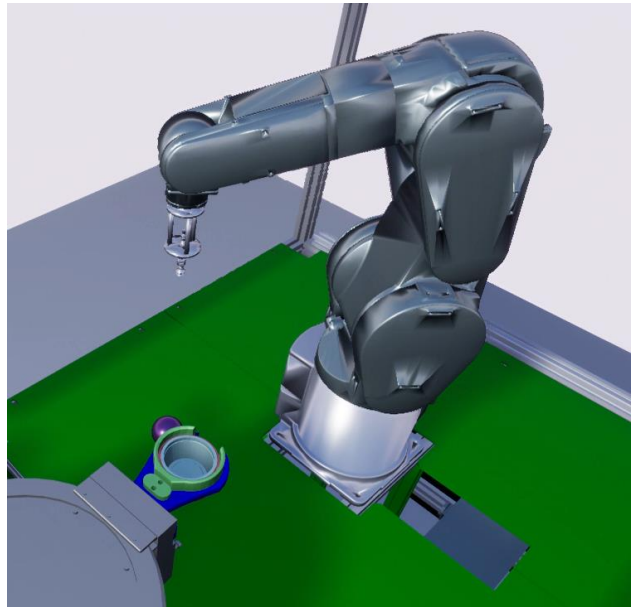
Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

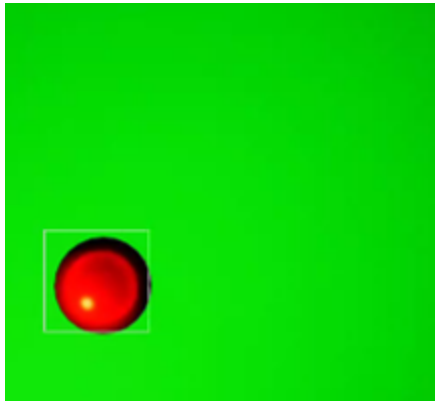
Robot Control Logic Design with Stateflow





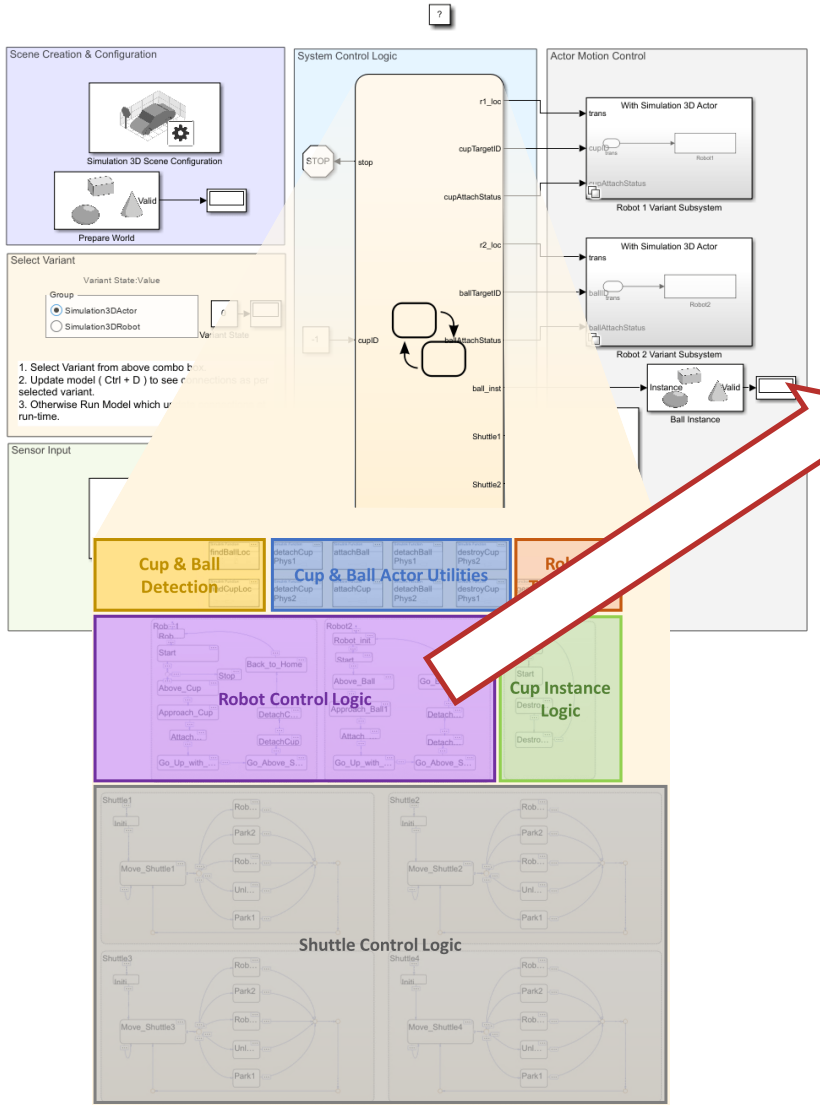


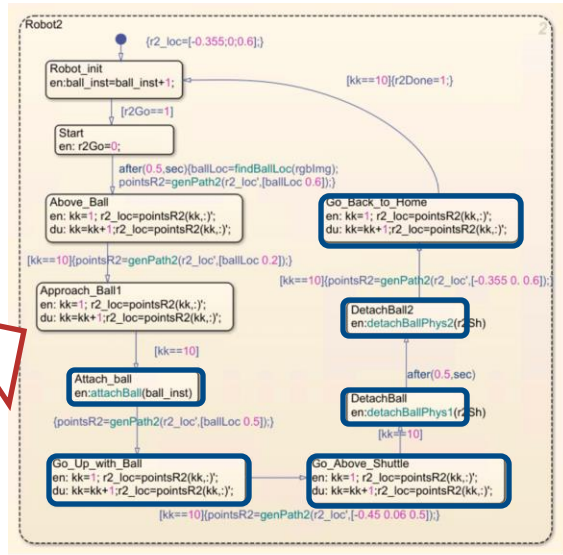


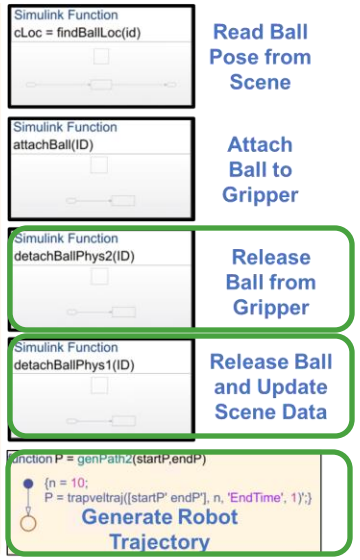


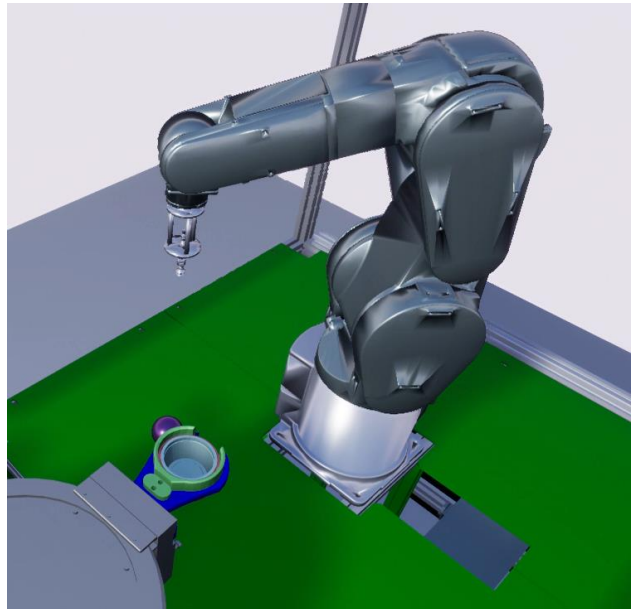
Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

Robot Control Logic Design with Stateflow





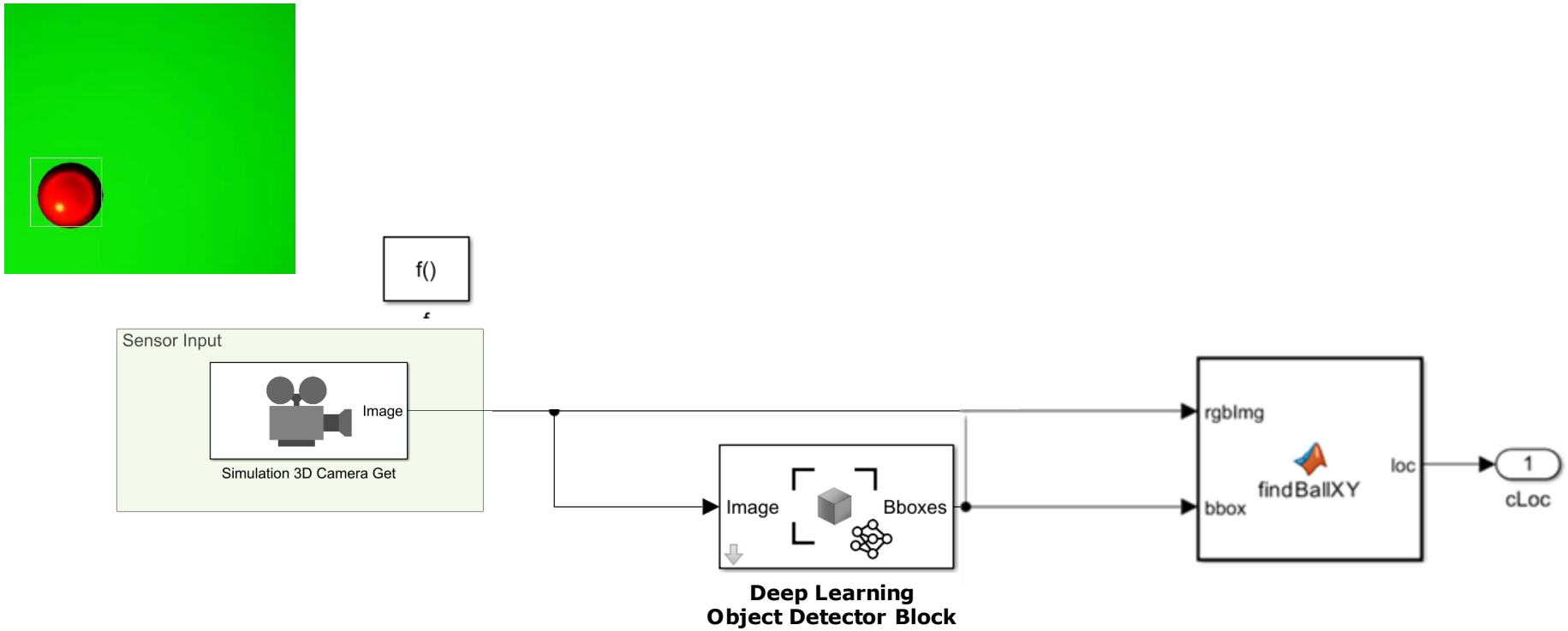




Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

Sensor Simulation

- The ball actor is created with a ***small random offset*** in position to reflect the real-life variability.
- **3D Camera Get**: provide an image to detect the ball
- **Deep Learning Object Detector:** takes the image data and outputs the bounding boxes (yolov2ObjectDetector).
- **findBallXY** MATLAB function: converts the bounding boxes to the XY-positions of the ball.
- **Video Viewer**: Overlay a bounding box of a detected ball on a captured image and display it.



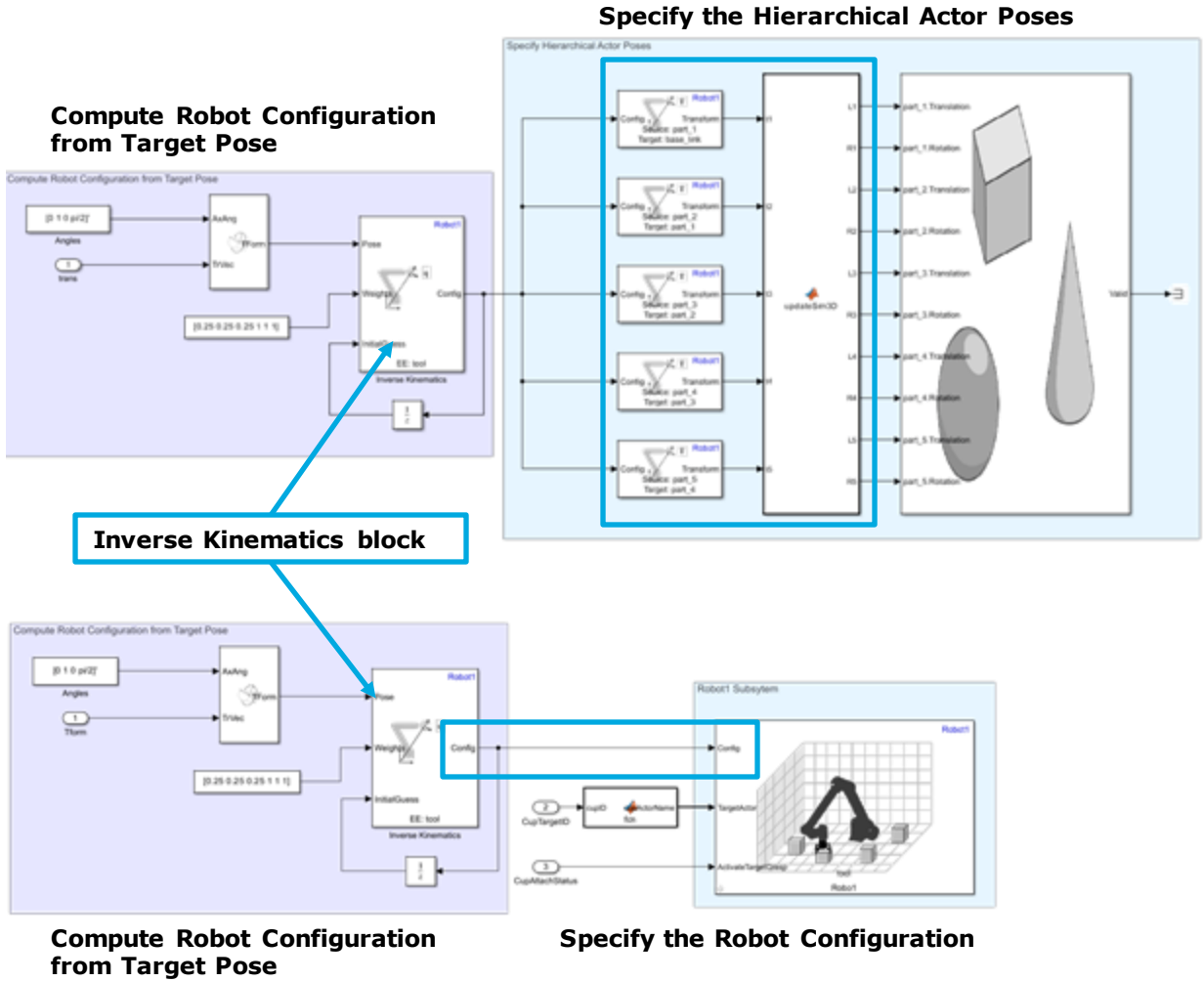
Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

Visualization of Robot Motion

- During scene creation, the system imports the rigid body trees for Robots from their original URDF files to the base MATLAB workspace.

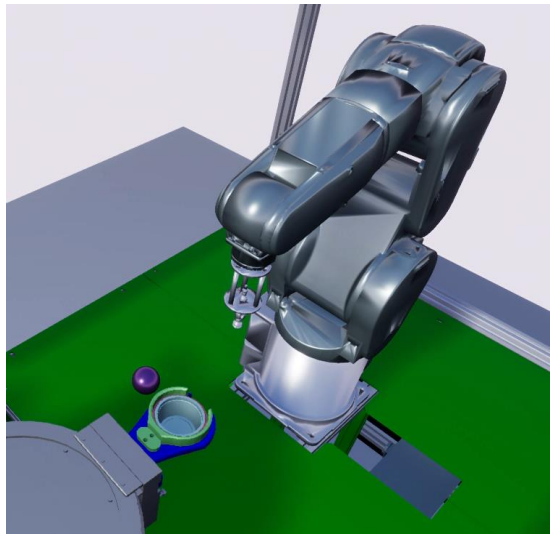
- ***Simulink 3D Actor*** block together with Get Transform blocks
  - Defines actors using STL, FBX, X3D
  - Enables to ***freely position*** any of the robot bodies regardless of kinematic constraints.
  - Requires the set of joint angles into six relative poses that relate each body to their corresponding parent body.

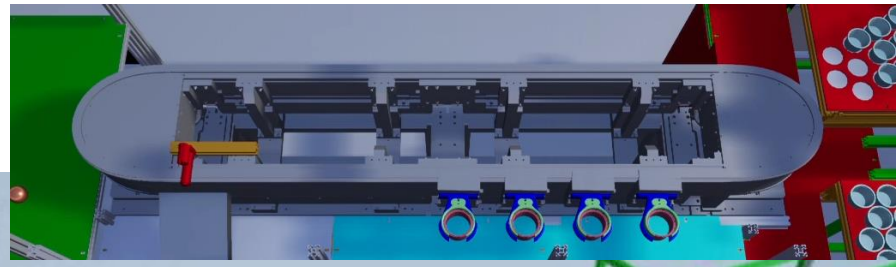
- ***Simulation 3D Robot*** block
  - Defines actors using ***URDF***
  - Enable to position the robot bodies ***considering kinematics constraints*** of the robot defined in URDF file.
  - Requires robot configuration info.

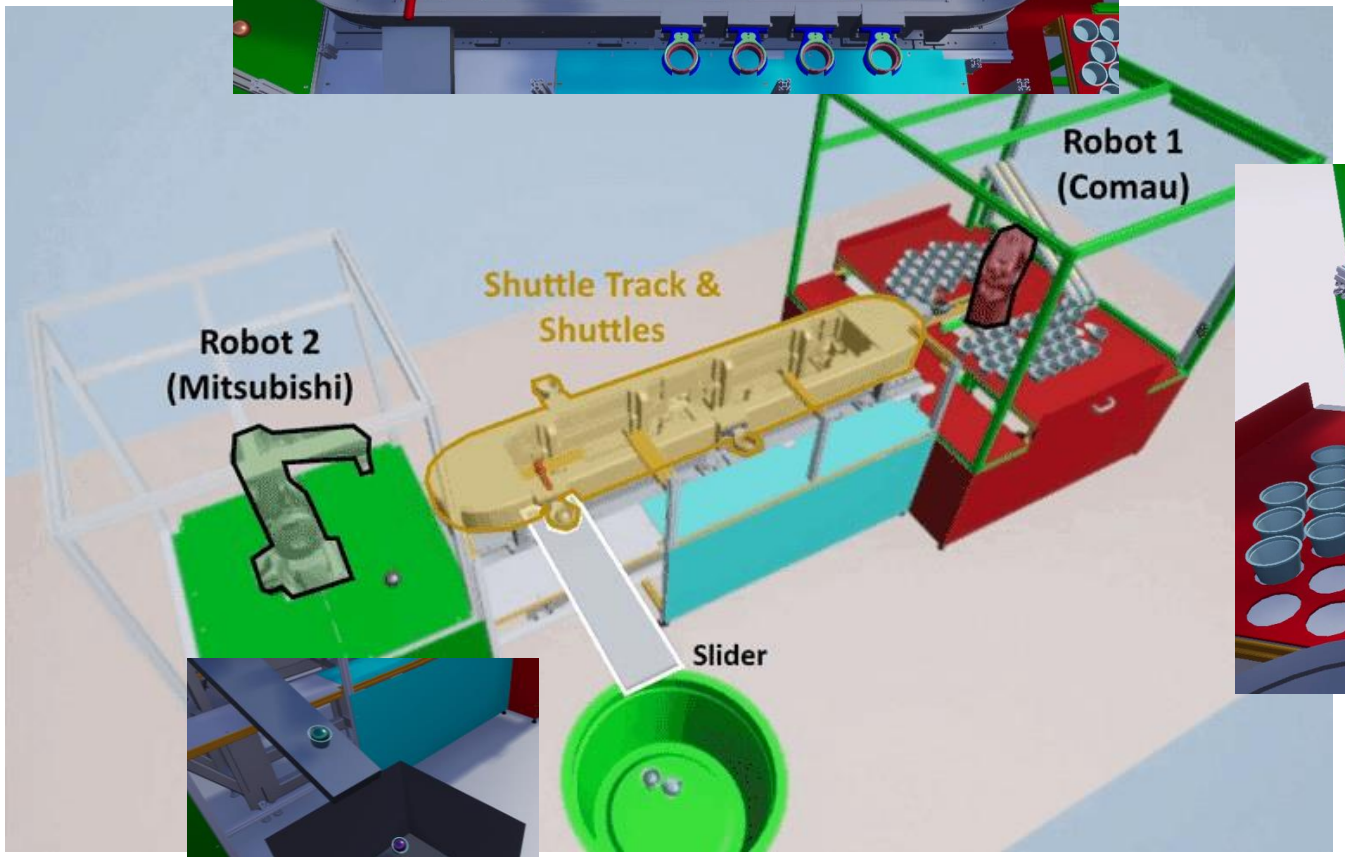


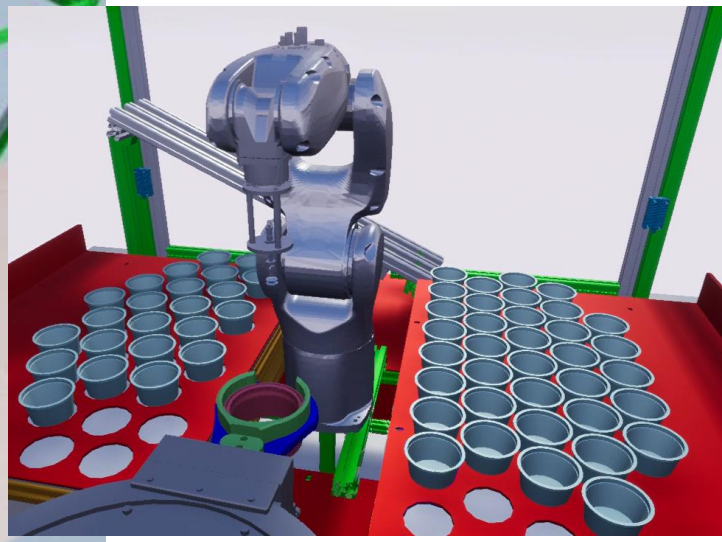
Case Study: Automate Virtual Assembly Line with Two Robotic Workcells

Simulation Result

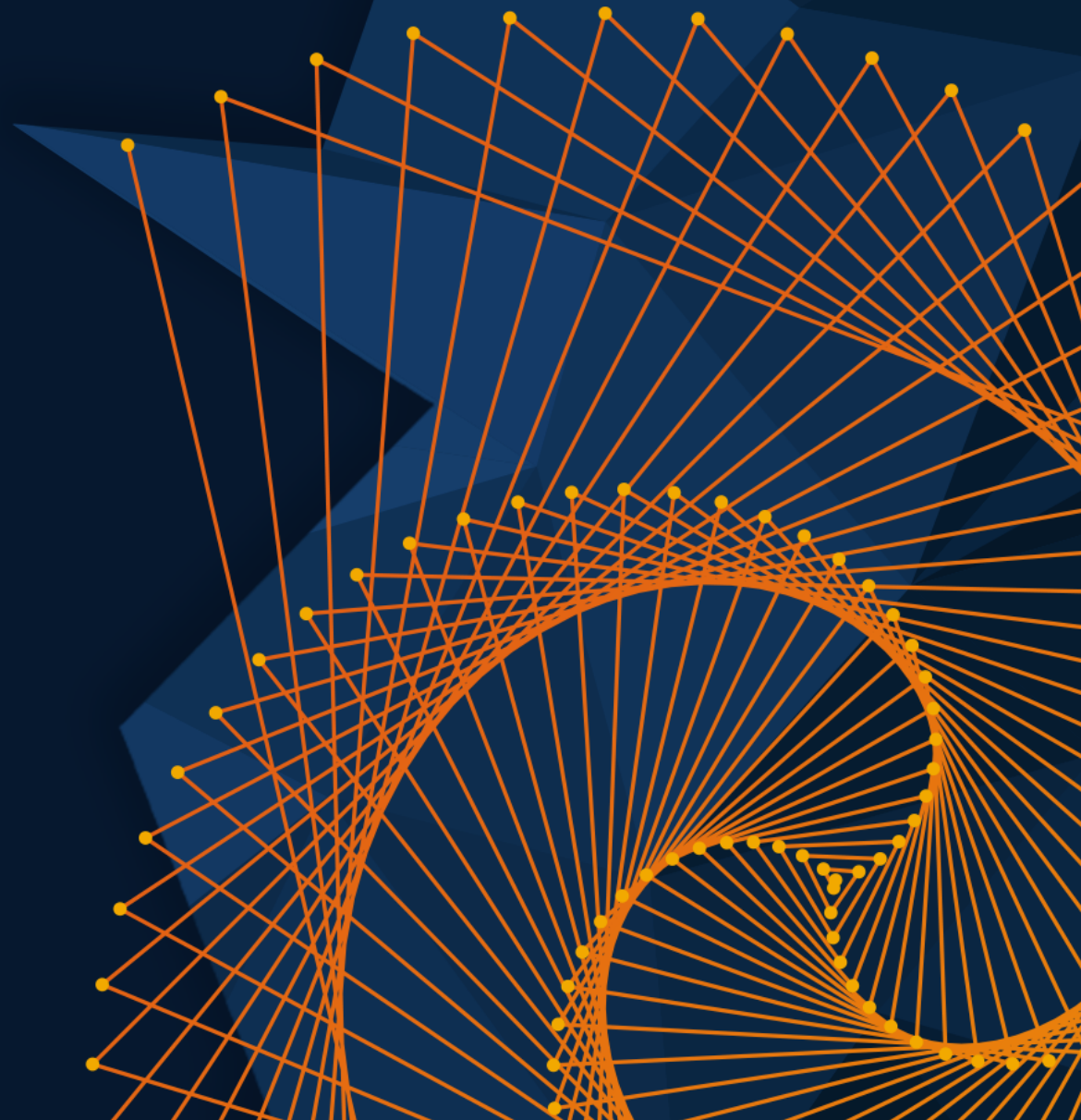








MATLAB EXPO



MathWorks®