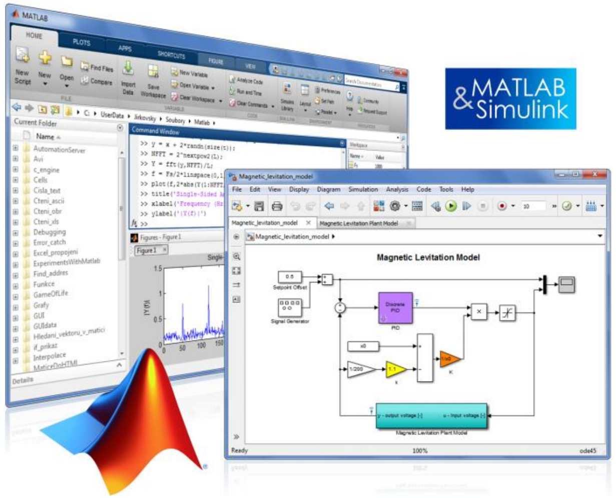
10.09.2020 Brno

TCC 2020

Nástroje pro vývoj robotických systémov



Michal Blaho

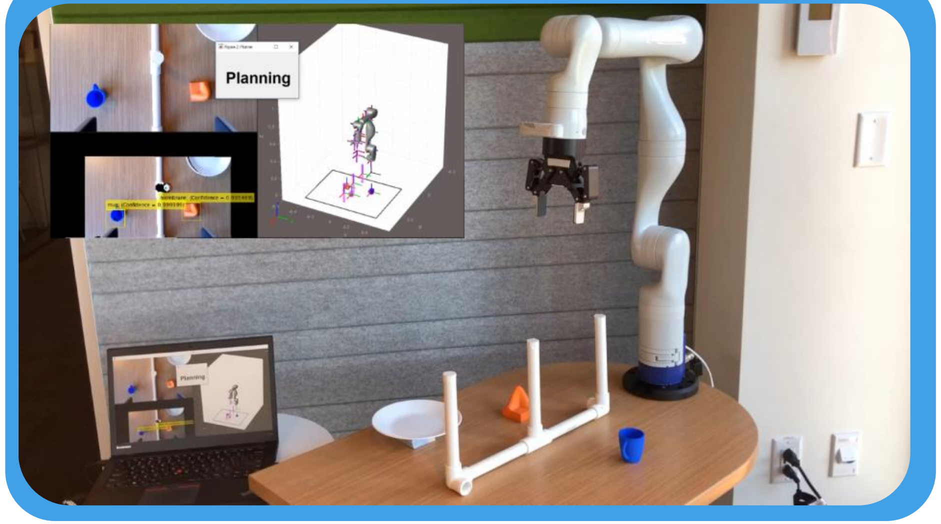
blaho@humusoft.cz

www.humusoft.cz
info@humusoft.cz

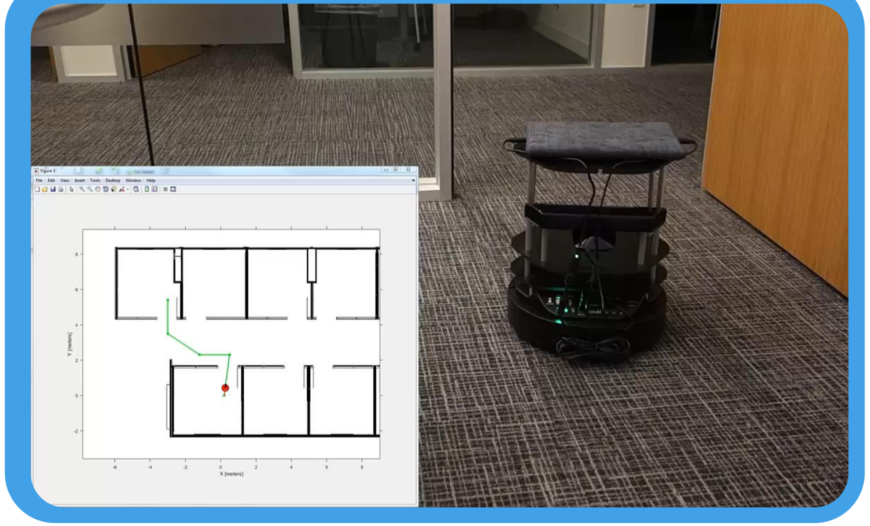
www.mathworks.com

Robot Applications

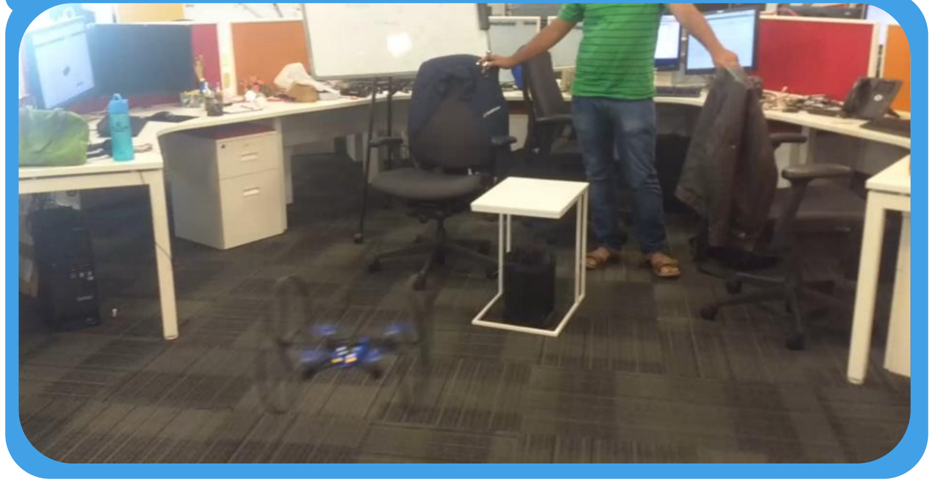
Manipulator Arms



Mobile Robots

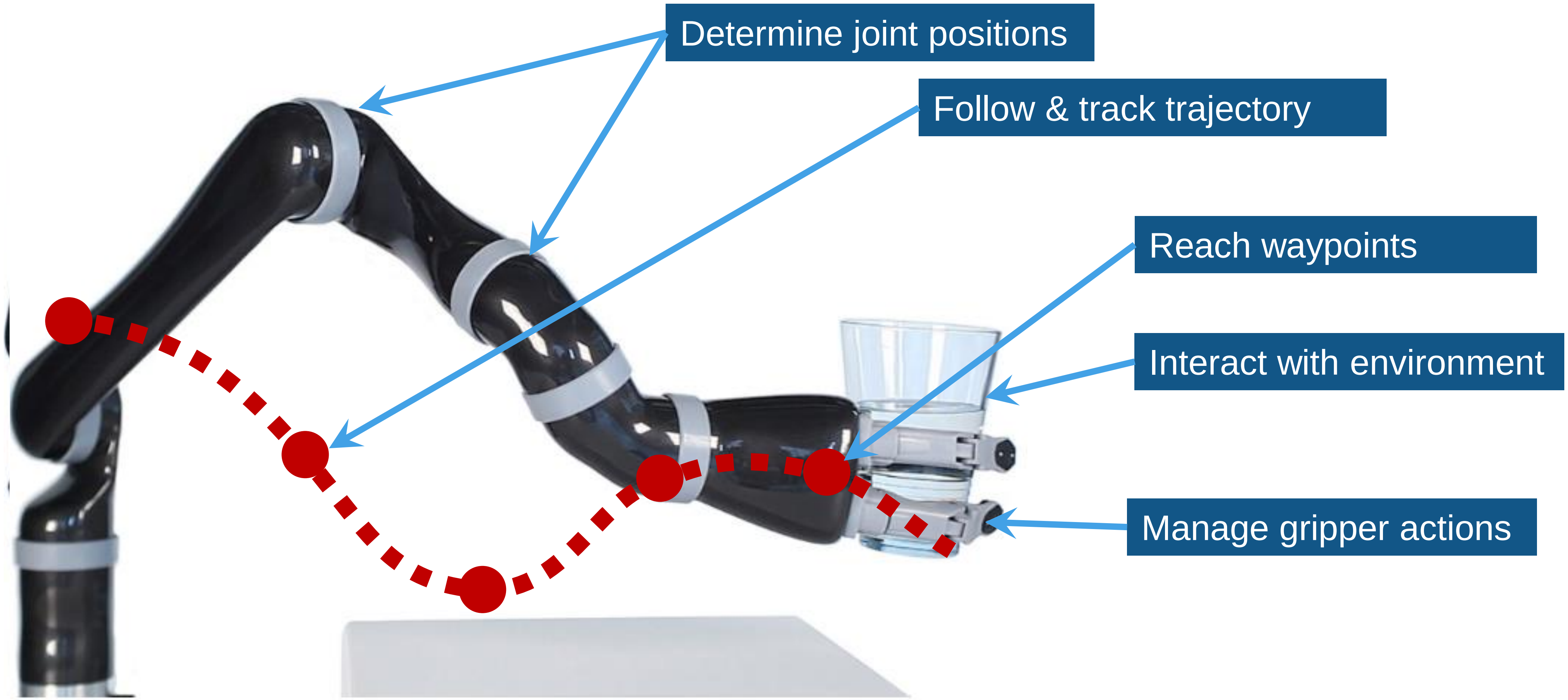


UAVs



Humanoids

Challenges in Designing Robotics System



Determine joint positions

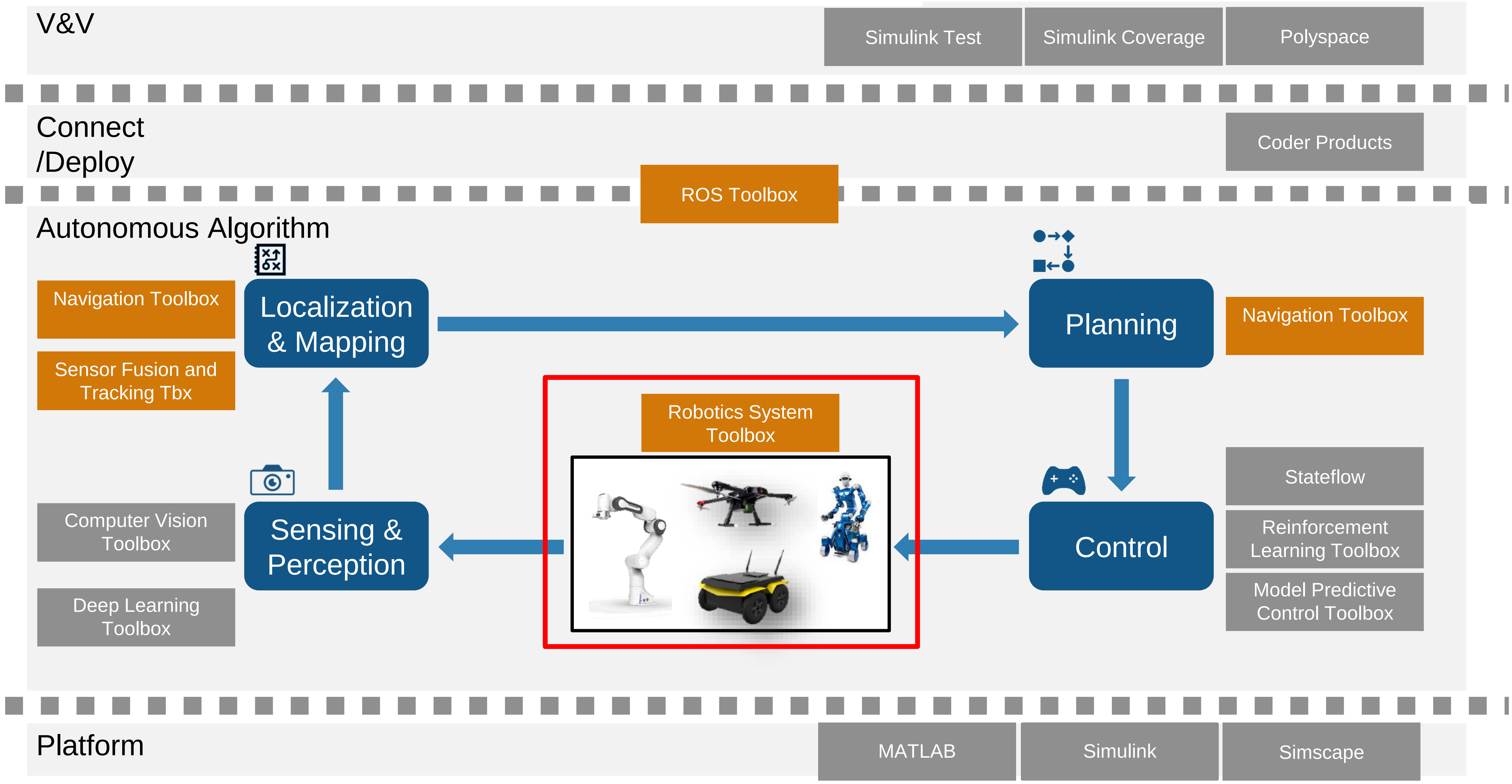
Follow & track trajectory

Reach waypoints

Interact with environment

Manage gripper actions

Components of Robotics System Development



V&V

Simulink Test

Simulink Coverage

Polyspace

Connect /Deploy

Coder Products

ROS Toolbox

Autonomous Algorithm

Navigation Toolbox

Sensor Fusion and Tracking Tbx

Localization & Mapping

Planning

Navigation Toolbox

Robotics System Toolbox

Computer Vision Toolbox

Deep Learning Toolbox

Sensing & Perception

Control

Stateflow

Reinforcement Learning Toolbox

Model Predictive Control Toolbox

Platform

MATLAB

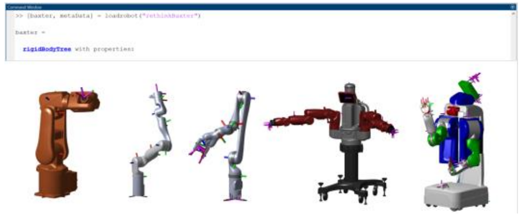
Simulink

Simscape

Robotics System Toolbox

Robot Model

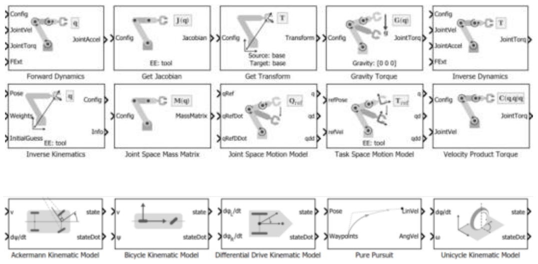
- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape multibody model



```
>> [baxter, metaData] = loadrobot("rethinkBaxter")

baxter =

  rigidBodyTree with properties:
```

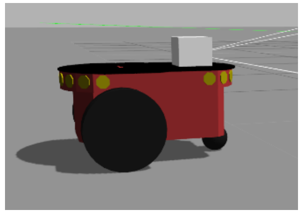
Algorithms

- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs



Simulation

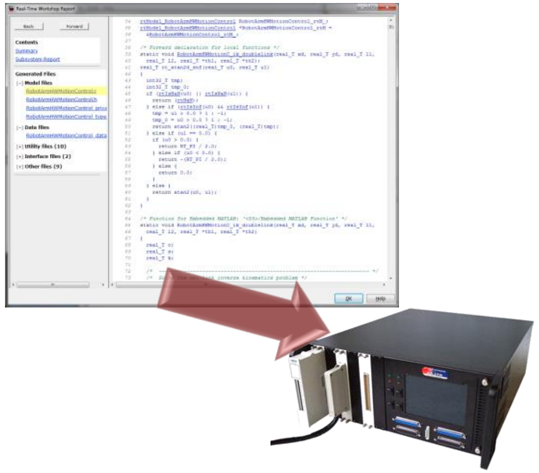
- Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.
- Synchronized Gazebo co-simulation from Simulink



GAZEBO

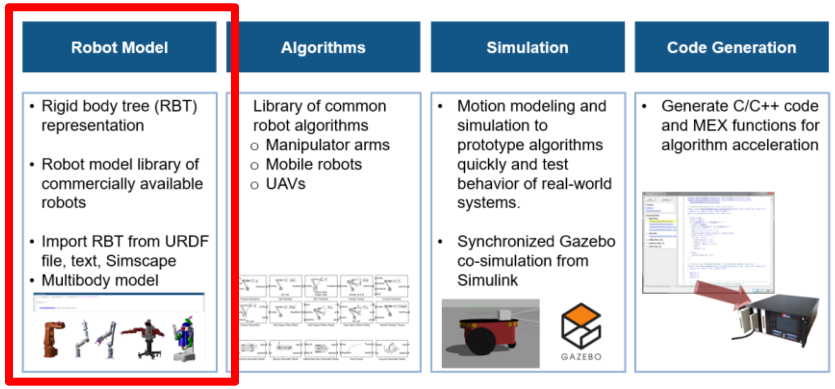
Deployment

- Generate C/C++ code and MEX functions for algorithm acceleration


Real-Time Workshop Report
Back
Forward
Contents
Summary
Subsystem Report
Generated Files
[-] Model files
RobotArmHWMotionControl.c
RobotArmHWMotionControl.h
RobotArmHWMotionControl_priva
RobotArmHWMotionControl_type
[-] Data files
RobotArmHWMotionControl_data
[+] Utility files (10)
[+] Interface files (2)
[+] Other files (9)
OK
Help

```
rtModel_RobotArmHWMotionControl RobotArmHWMotionControl_rtM_;
rtModel_RobotArmHWMotionControl *RobotArmHWMotionControl_rtM =
  &RobotArmHWMotionControl_rtM_;

/* Forward declaration for local functions */
static void RobotArmHWMotionC_ik_doublelink(real_T xd, real_T yd, real_T l1,
  real_T l2, real_T *th1, real_T *th2);
real_T rt_atan2d_snf(real_T u0, real_T u1)
{
  int32_T tmp;
  int32_T tmp_0;
  if (rtIsNaN(u0) || rtIsNaN(u1)) {
    return (rtNaN);
  } else if (rtIsInf(u0) && rtIsInf(u1)) {
    tmp = u1 > 0.0 ? 1 : -1;
    tmp_0 = u0 > 0.0 ? 1 : -1;
    return atan2((real_T)tmp_0, (real_T)tmp);
  } else if (u1 == 0.0) {
    if (u0 > 0.0) {
      return RT_PI / 2.0;
    } else if (u0 < 0.0) {
      return -(RT_PI / 2.0);
    } else {
      return 0.0;
    }
  } else {
    return atan2(u0, u1);
  }
}

/* Function for Embedded MATLAB: '<S3>/Embedded MATLAB Function' */
static void RobotArmHWMotionC_ik_doublelink(real_T xd, real_T yd, real_T l1,
  real_T l2, real_T *th1, real_T *th2)
{
  real_T c;
  real_T s;
  real_T k;

  /* ---------------------------------------------------------------- */
  /* Solve the two-link inverse kinematics problem */
```

Robot Model



Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape
- Multibody model

Algorithms

Library of common robot algorithms
- Manipulator arms
- Mobile robots
- UAVs

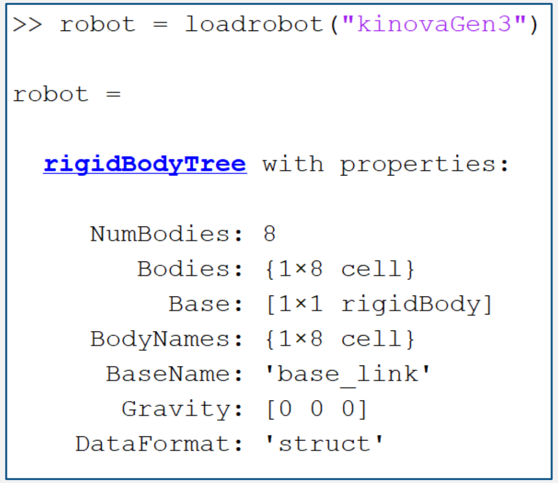
Simulation

- Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.
- Synchronized Gazebo co-simulation from Simulink

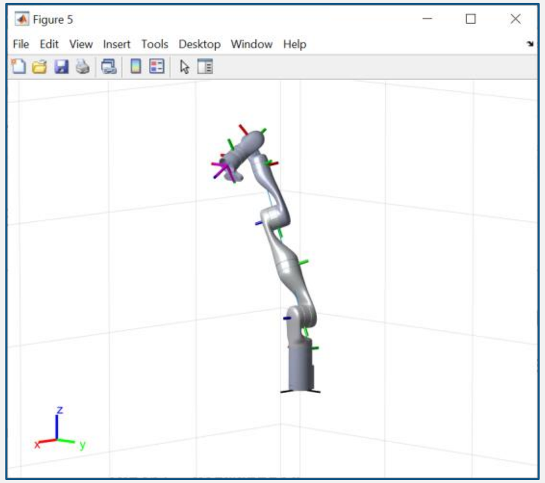
Code Generation

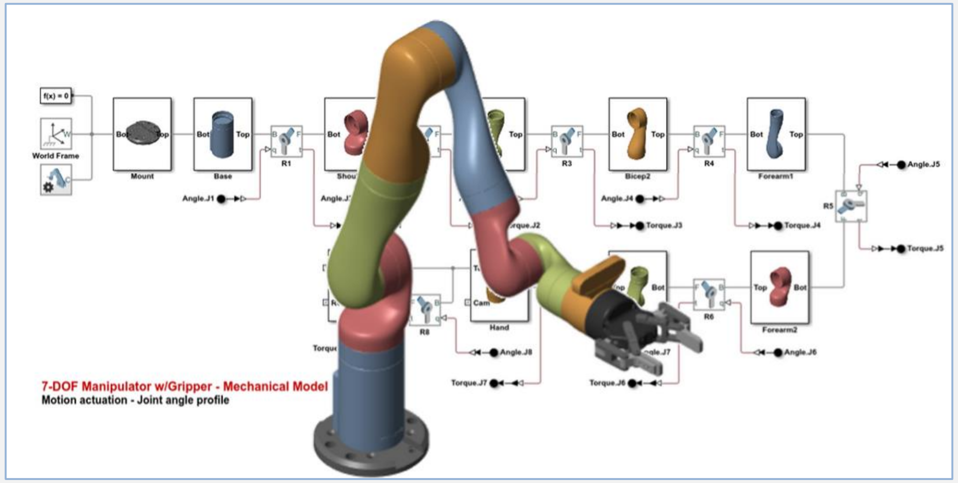
- Generate C/C++ code and MEX functions for algorithm acceleration

- Rigid body tree (RBT) representation
- Load a RBT robot model from a library of commonly used robots
- Import a RBT robot mode from URDF file



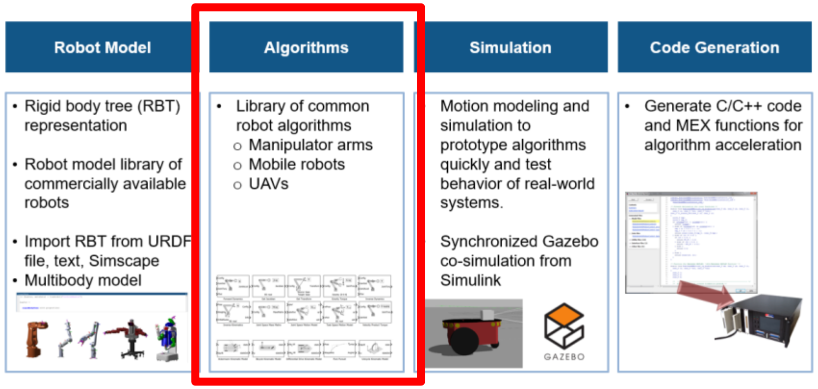
```
>> robot = loadrobot("kinovaGen3")

robot =

  rigidBodyTree with properties:

     NumBodies: 8
        Bodies: {1×8 cell}
          Base: [1×1 rigidBody]
     BodyNames: {1×8 cell}
      BaseName: 'base_link'
       Gravity: [0 0 0]
    DataFormat: 'struct'
```





7-DOF Manipulator w/Gripper - Mechanical Model
Motion actuation - Joint angle profile

Algorithms - Manipulation



Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape
- Multibody model

Algorithms

- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs

Simulation

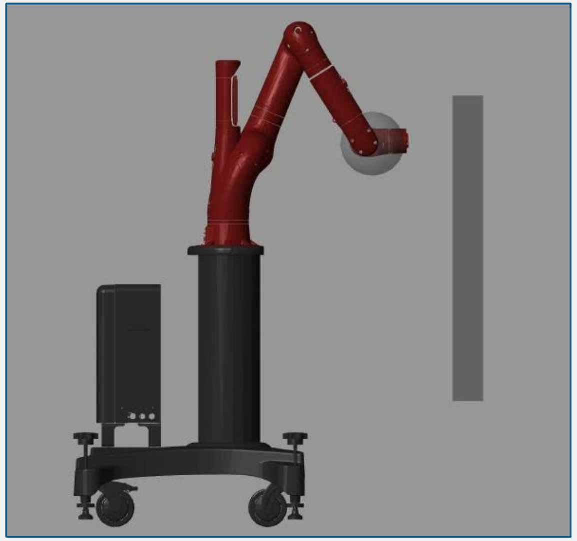
Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.

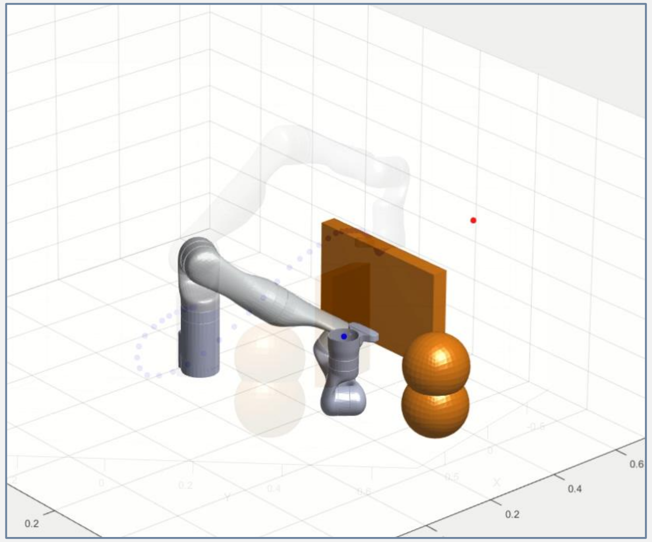
Synchronized Gazebo co-simulation from Simulink

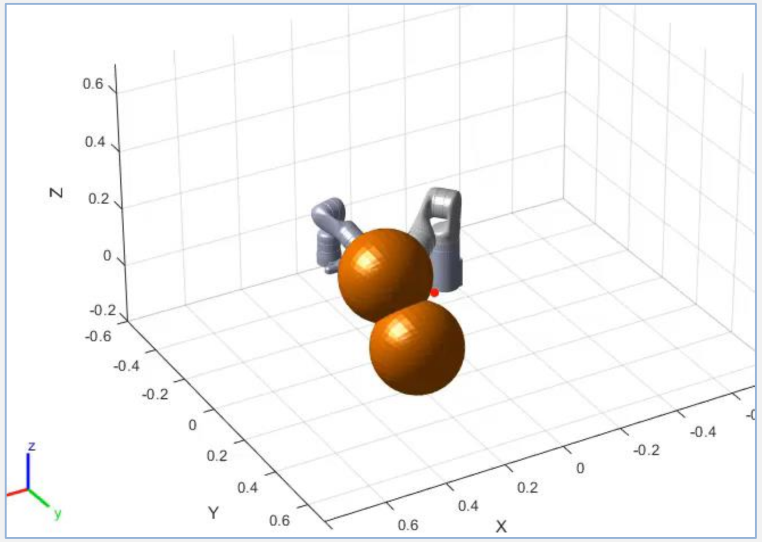
Code Generation

- Generate C/C++ code and MEX functions for algorithm acceleration

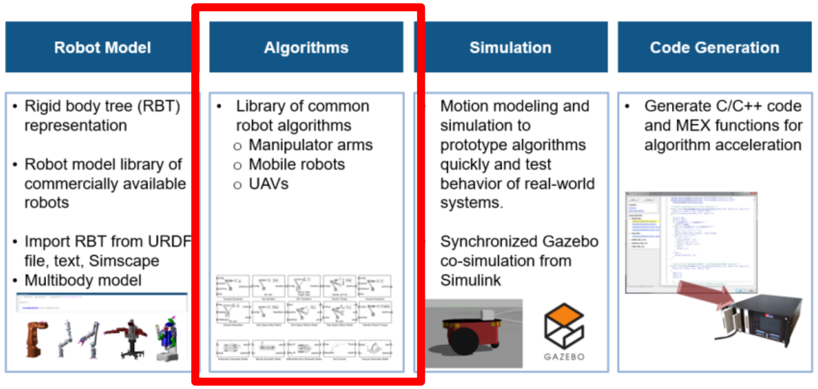
- Forward and inverse kinematics
- Generalized inverse kinematics & constraints
- Forward and inverse dynamics
- Trajectory generation
- Collision checking







Algorithms – Mobile Robots



Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape
- Multibody model

Algorithms

- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs

Simulation

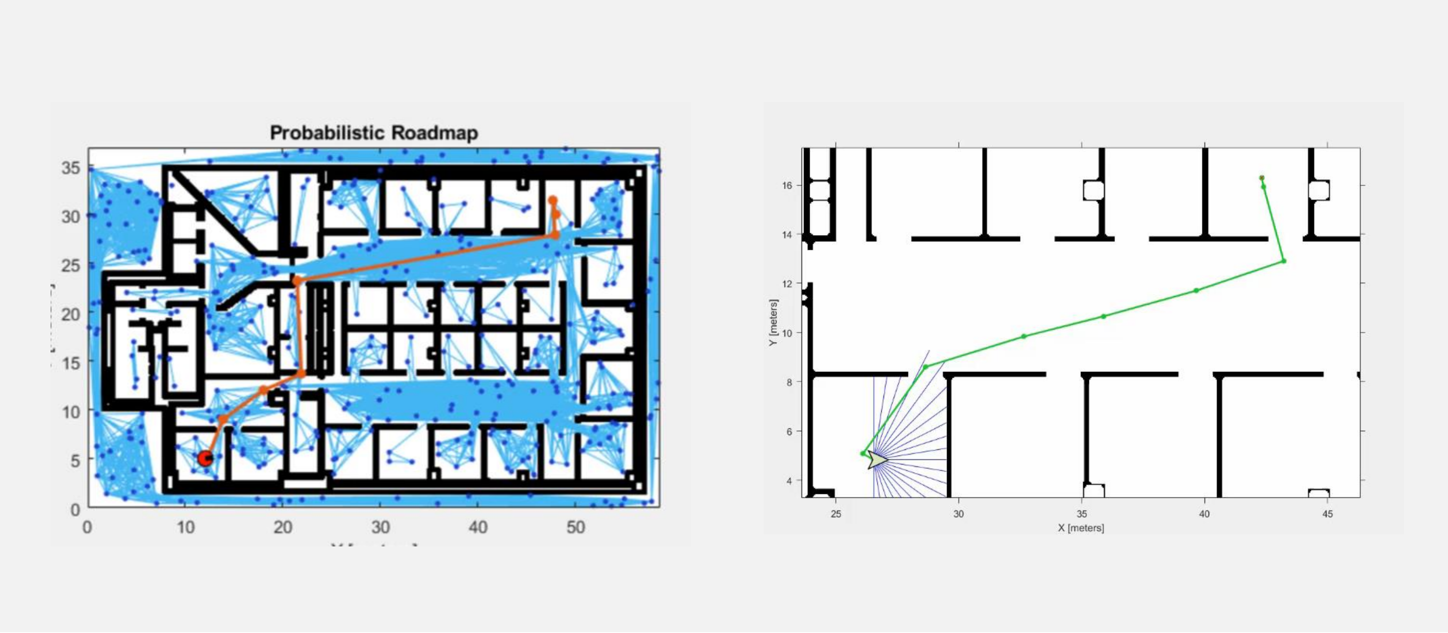
Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.

Synchronized Gazebo co-simulation from Simulink

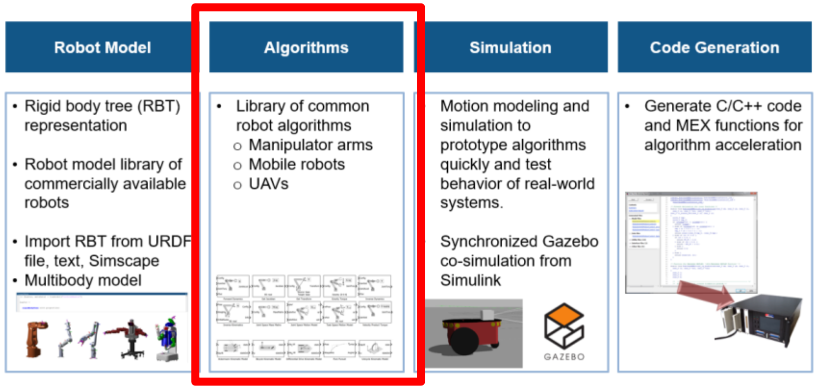
Code Generation

- Generate C/C++ code and MEX functions for algorithm acceleration

- Mapping and map representation
  - Binary occupancy grid
- Localization
  - Odometry
  - stateEstimatorPF
- Path planning
  - Probabilistic roadmap (PRM)
- Path following
  - Pure pursuit



Algorithms – UAVs (Add-On Library)



Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape
- Multibody model

Algorithms

- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs

Simulation

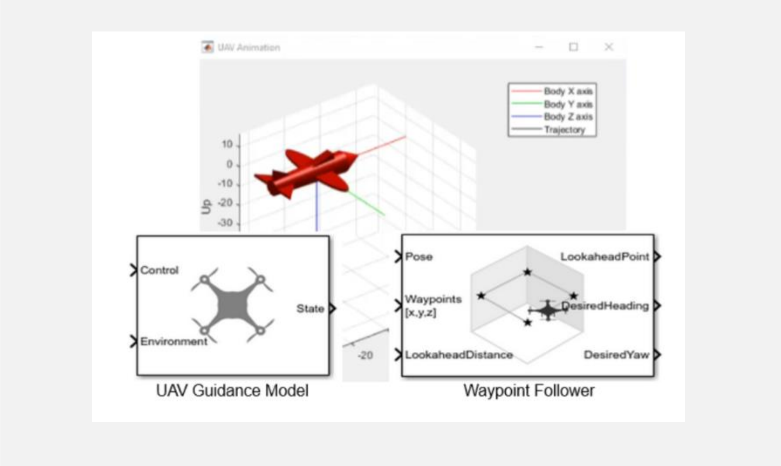
Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.

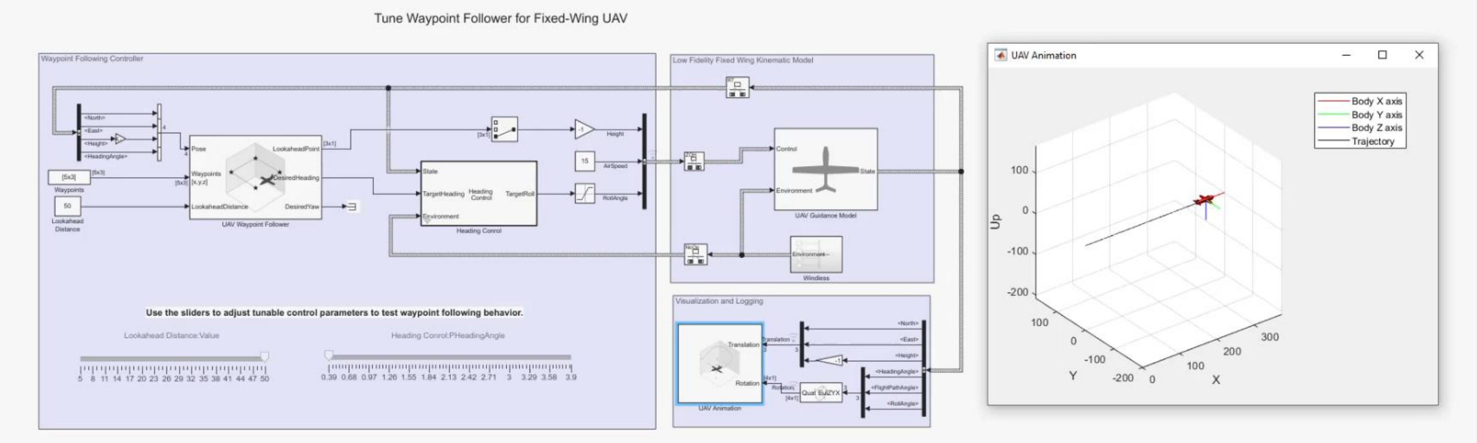
Synchronized Gazebo co-simulation from Simulink

Code Generation

- Generate C/C++ code and MEX functions for algorithm acceleration

- Guidance models
  - Reduced-order guidance model for fixed-wing and multi-rotor UAVs
- MAVLink communication
  - Communicate with simulated/physical UAV
  - Import and analyze UAV flight logs
- Waypoint following
  - Execute flight missions based on given waypoints

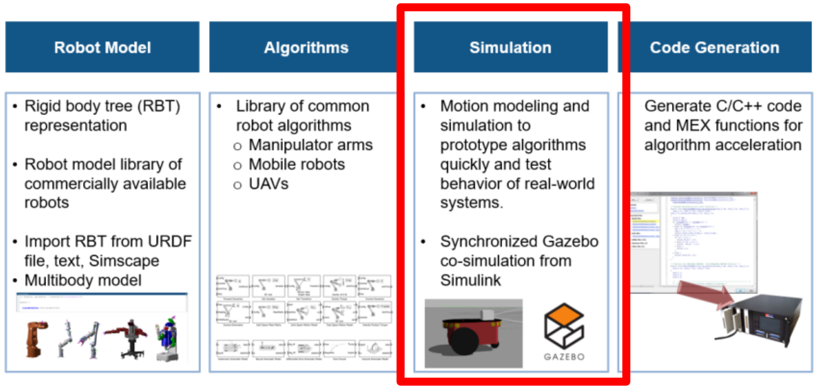




Tune Waypoint Follower for Fixed-Wing UAV

Use the sliders to adjust tunable control parameters to test waypoint following behavior.

Simulation



Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape Multibody model

Algorithms

- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs

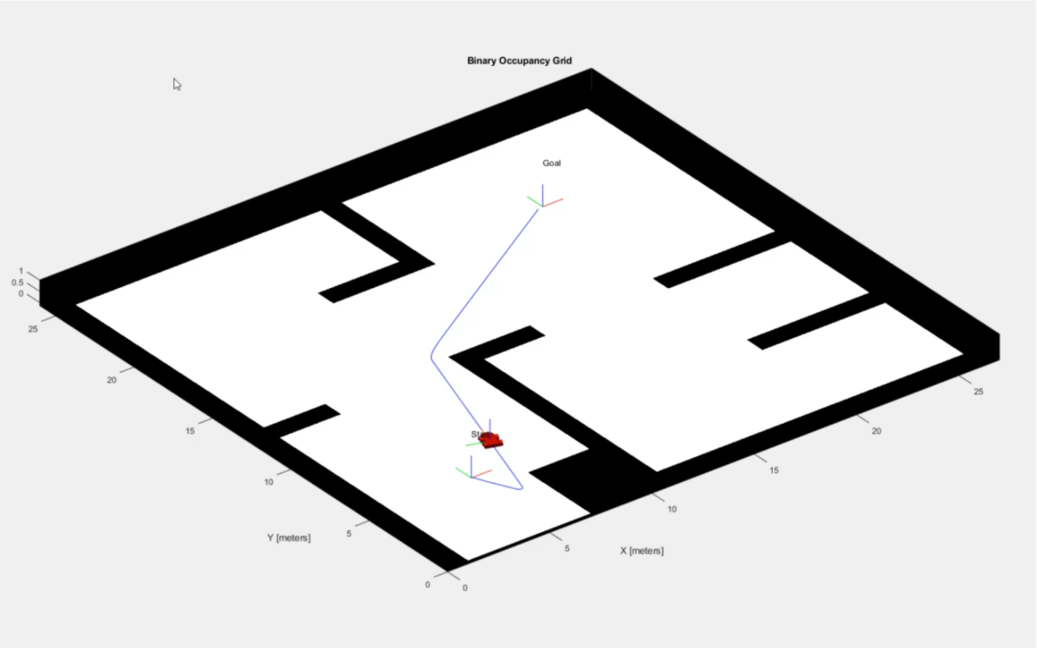
Simulation

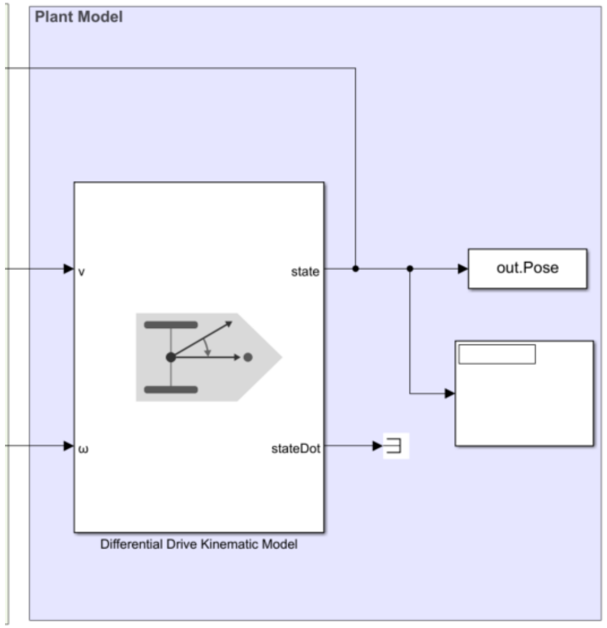
- Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.
- Synchronized Gazebo co-simulation from Simulink

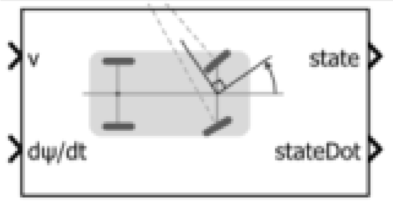
Code Generation

Generate C/C++ code and MEX functions for algorithm acceleration

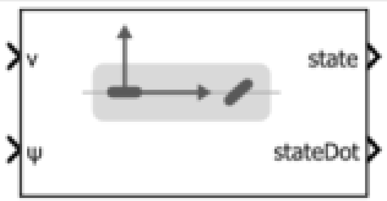
- Low-fidelity simulation



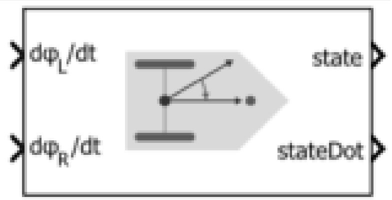




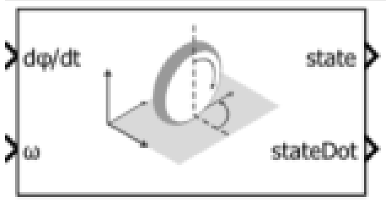
Ackermann Kinematic Model



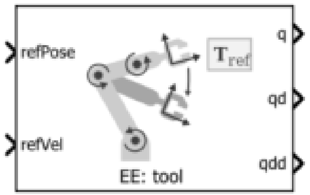
Bicycle Kinematic Model



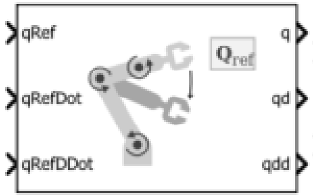
Differential Drive Kinematic Model



Unicycle Kinematic Model

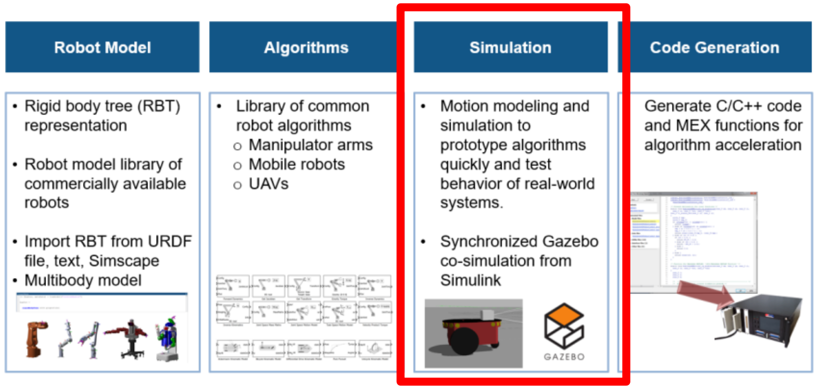


Task Space Motion Model



Joint Space Motion Model

Simulation



Robot Model
- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape Multibody model

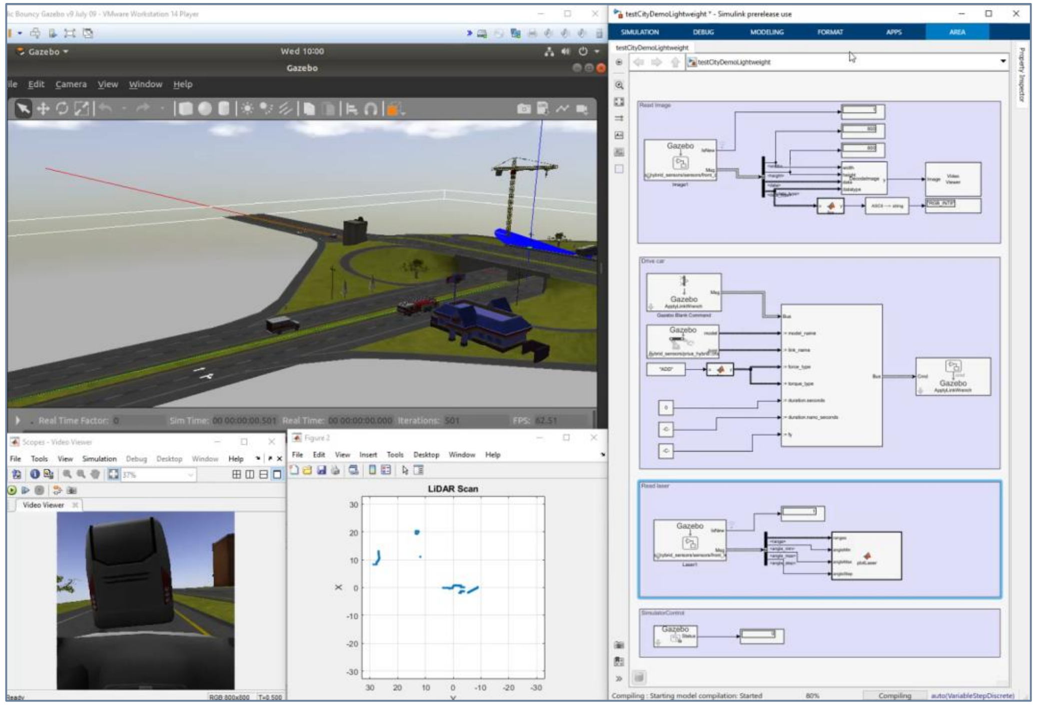
Algorithms
- Library of common robot algorithms
  - Manipulator arms
  - Mobile robots
  - UAVs

Simulation
- Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.
- Synchronized Gazebo co-simulation from Simulink

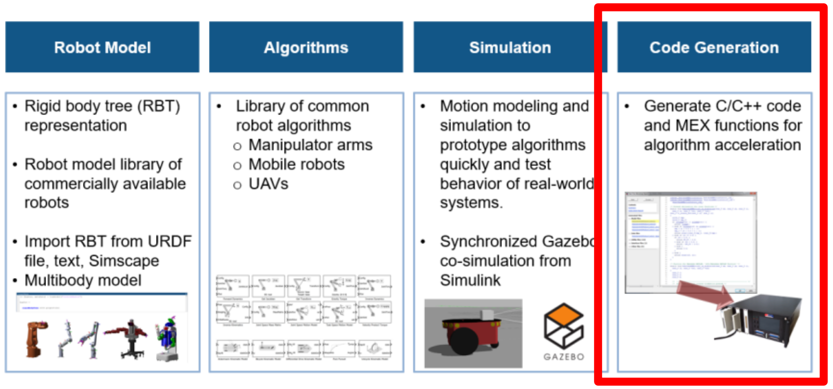
Code Generation

Generate C/C++ code and MEX functions for algorithm acceleration

- Gazebo Co-simulation
  - Provides synchronized stepping between Simulink and Gazebo simulator
  - Retrieve sensor data and ground truth pose for models from Gazebo simulator
  - Actuate model links and joints in Gazebo simulator

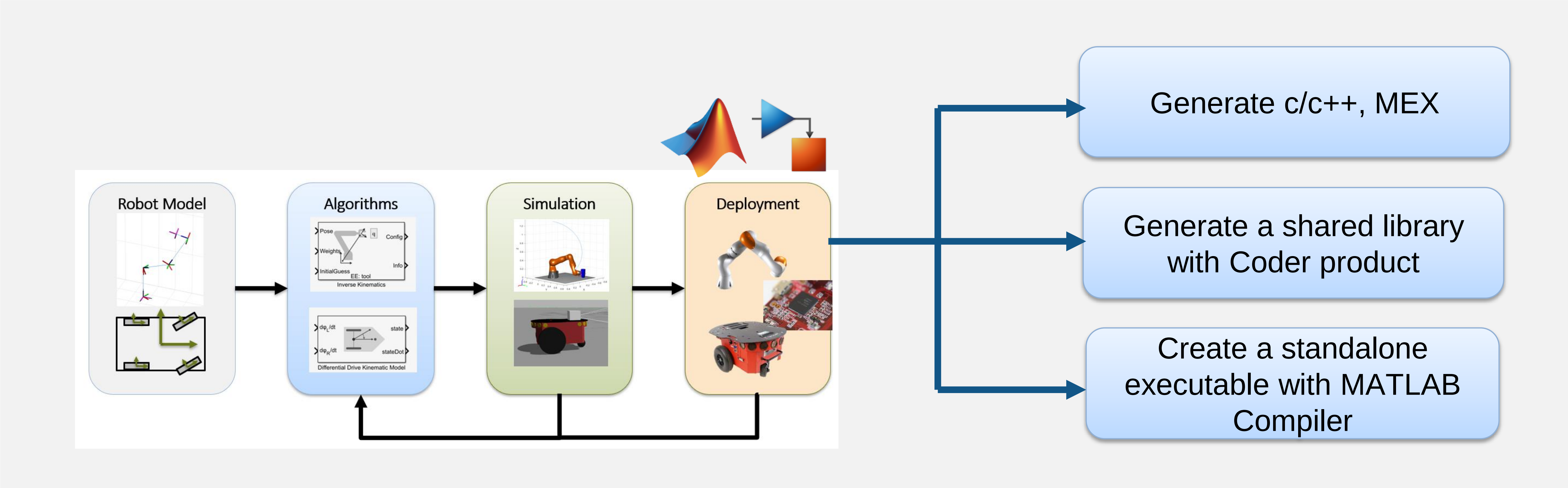

Gazebo
Wed 10:00
File Edit Camera View Window Help
Video Viewer
Figure 2
LiDAR Scan
SIMULATION DEBUG MODELING FORMAT APPS AREA
testCityDemoLightweight

Deployment



Robot Model	Algorithms	Simulation	Code Generation
• Rigid body tree (RBT) representation • Robot model library of commercially available robots • Import RBT from URDF file, text, Simscape • Multibody model	• Library of common robot algorithms ○ Manipulator arms ○ Mobile robots ○ UAVs	• Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems. • Synchronized Gazebo co-simulation from Simulink	• Generate C/C++ code and MEX functions for algorithm acceleration

- Accelerate robotics algorithms with code generation

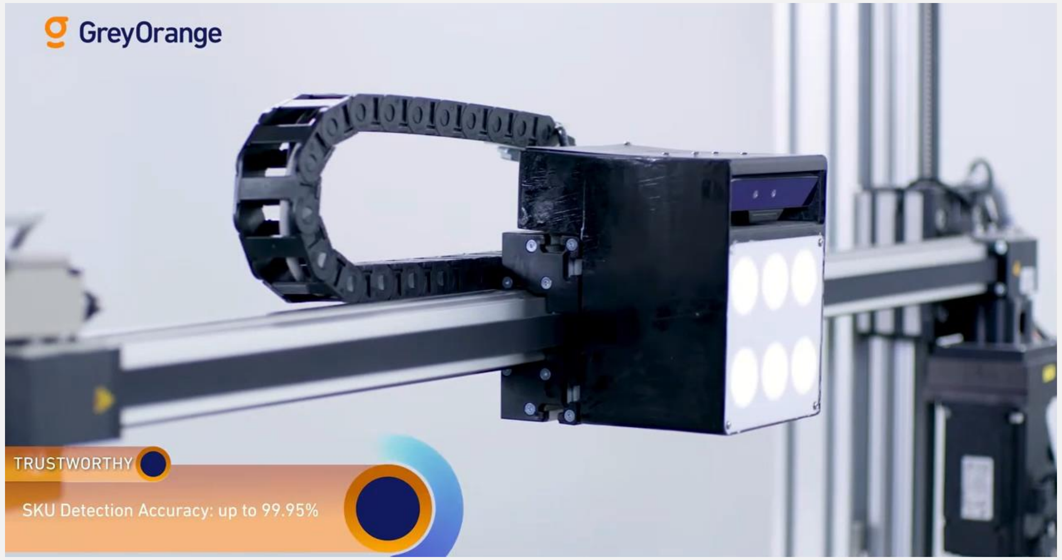


Generate c/c++, MEX

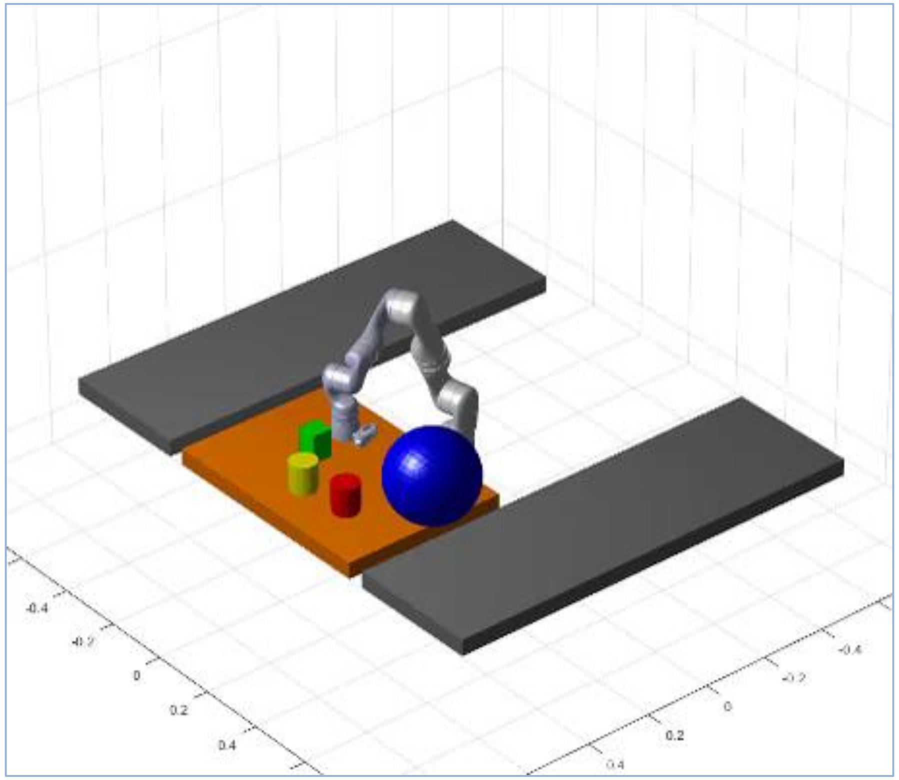
Generate a shared library with Coder product

Create a standalone executable with MATLAB Compiler

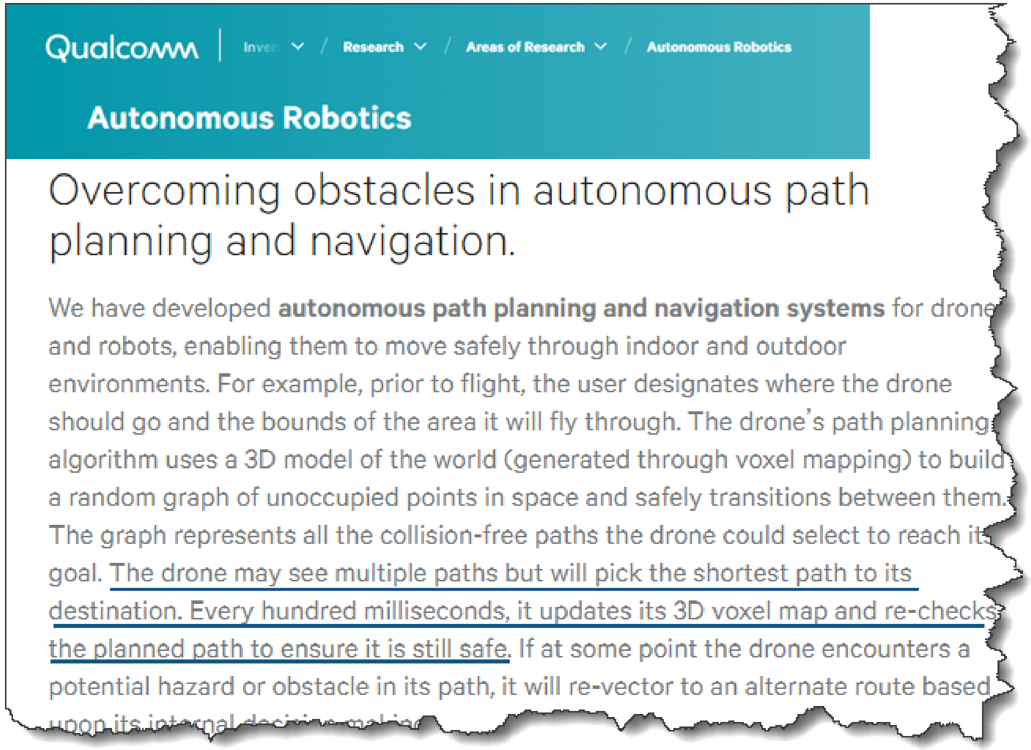
Example – Pick-and-Place Robot Arm



- Applications in **warehouses**, **manufacturing**, and **medical industries**
- RST: robot model, plan, control, and simulate robot
- MPC: trajectory optimization
- Stateflow: task-level planning and execution



Navigation is critical for autonomous systems



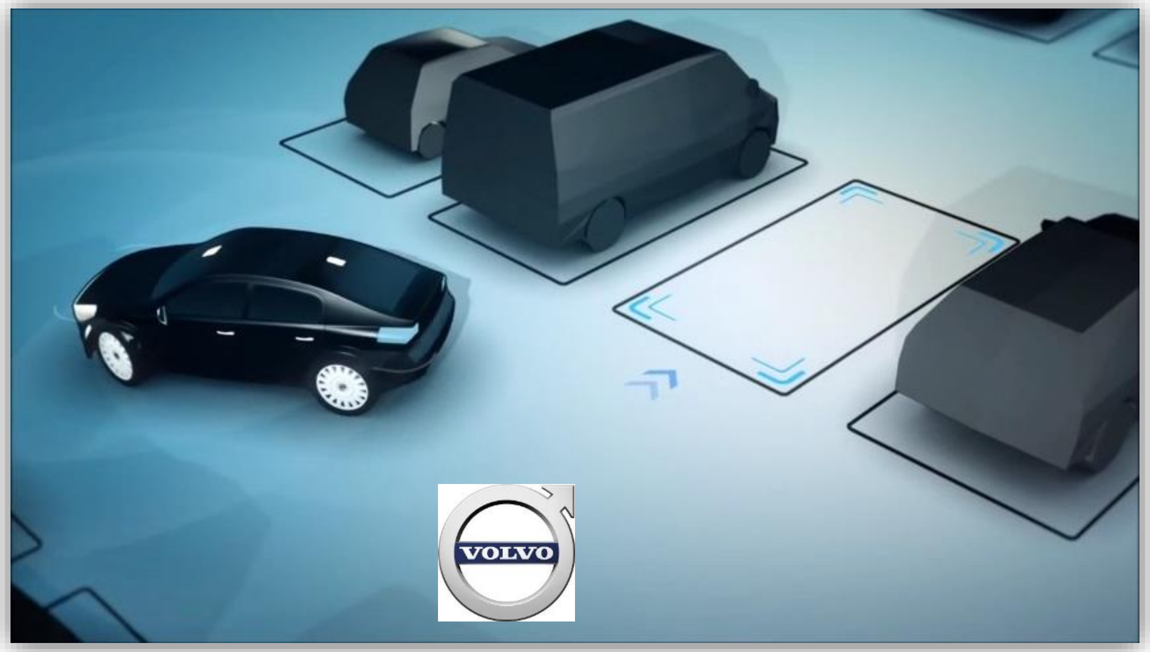
Autonomous Robotics

Overcoming obstacles in autonomous path planning and navigation.

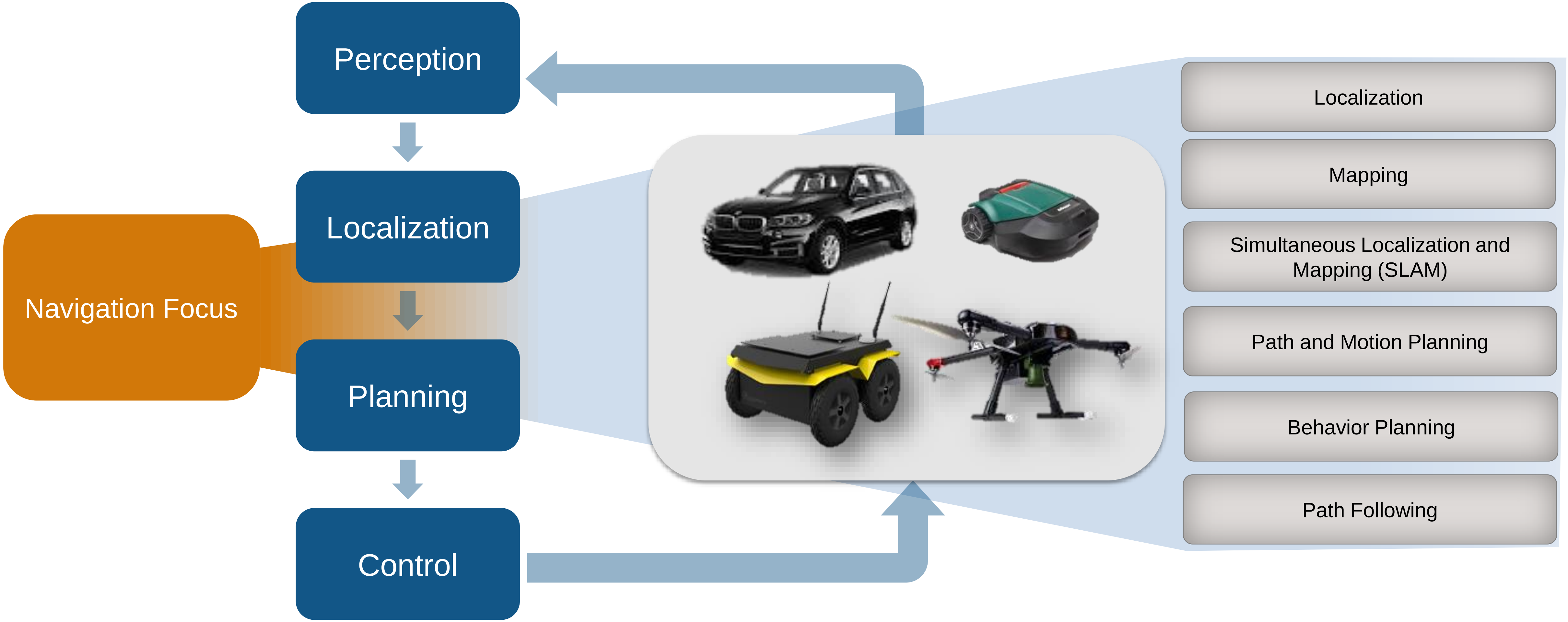
We have developed **autonomous path planning and navigation systems** for drone and robots, enabling them to move safely through indoor and outdoor environments. For example, prior to flight, the user designates where the drone should go and the bounds of the area it will fly through. The drone's path planning algorithm uses a 3D model of the world (generated through voxel mapping) to build a random graph of unoccupied points in space and safely transitions between them. The graph represents all the collision-free paths the drone could select to reach it goal. The drone may see multiple paths but will pick the shortest path to its destination. Every hundred milliseconds, it updates its 3D voxel map and re-checks the planned path to ensure it is still safe. If at some point the drone encounters a potential hazard or obstacle in its path, it will re-vector to an alternate route based upon its internal decision-making

Navigation Tools

▪ *Where am I going?*	**Behavior Planning**
▪ *What’s the best way there?*	**Path / Motion Planning**
▪ *Where have I been?*	**Mapping**
▪ *Where am I on map?*	**Localization**
▪ *What if you don’t have a map?*	**SLAM**



Autonomous Navigation Workflow



Perception

Localization

Planning

Control

Navigation Focus

Localization

Mapping

Simultaneous Localization and Mapping (SLAM)

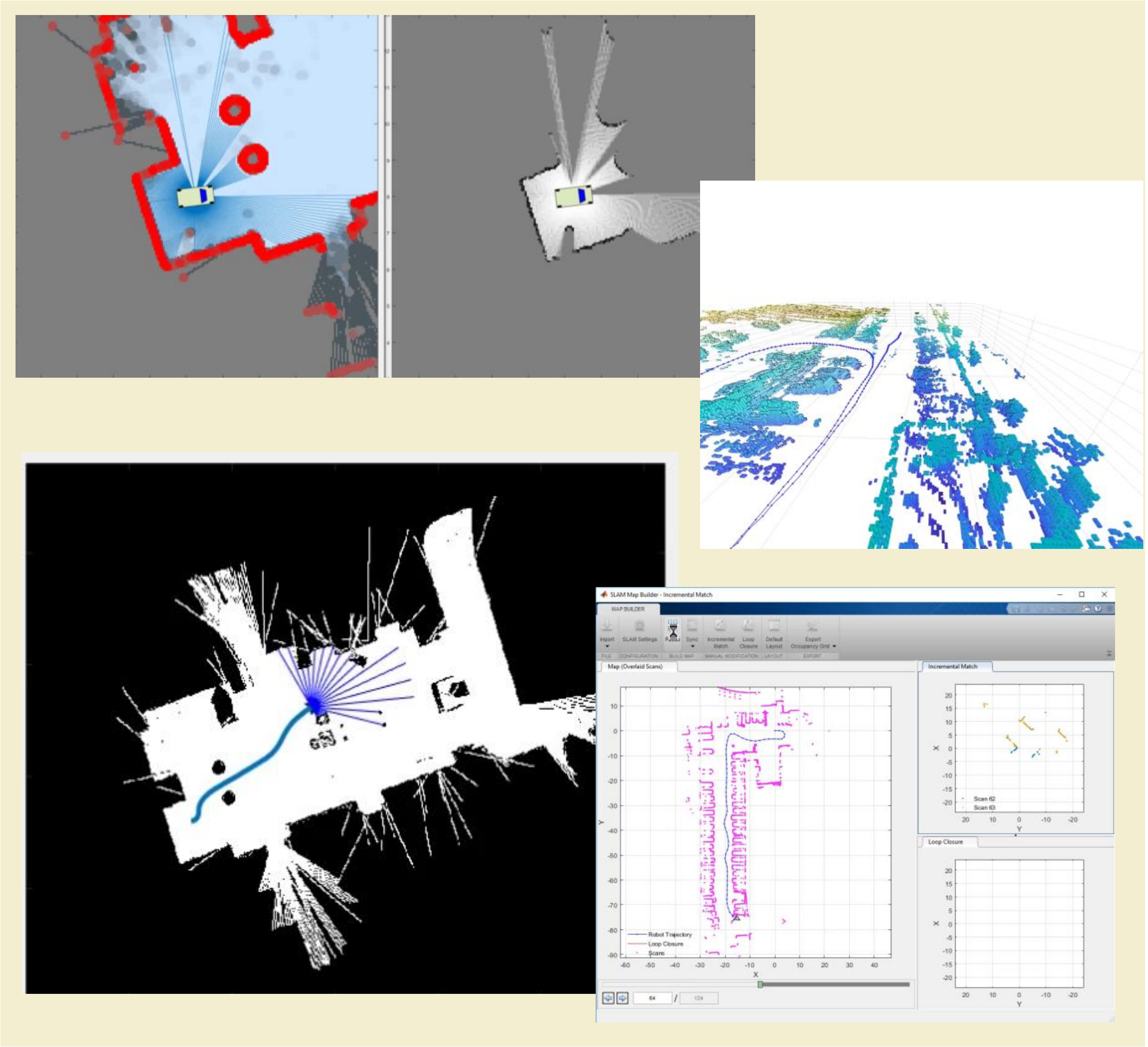
Path and Motion Planning

Behavior Planning

Path Following

Navigation Toolbox

- **Mapping and localization**
  - **2D and 3D SLAM**
  - **Egocentric maps**
  - **SLAM map builder App**



Navigation Toolbox

- Mapping and localization
  - 2D and 3D SLAM
  - Egocentric maps
  - SLAM map builder App
- **Path planning and Following**
  - **Algorithms for path planning**
  - **Planner interface**
  - **Path metrics**
  - **Path following and controls**

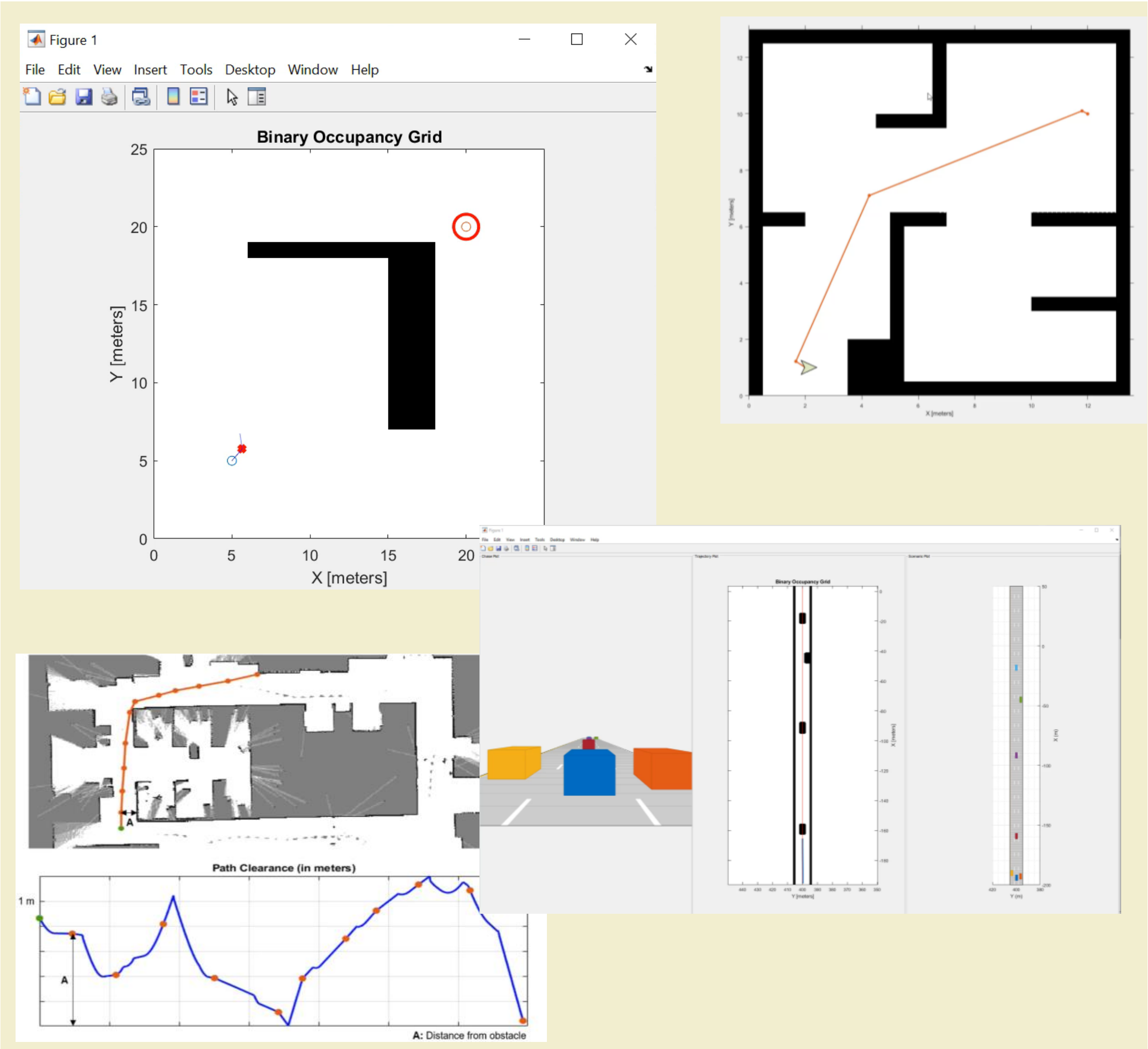
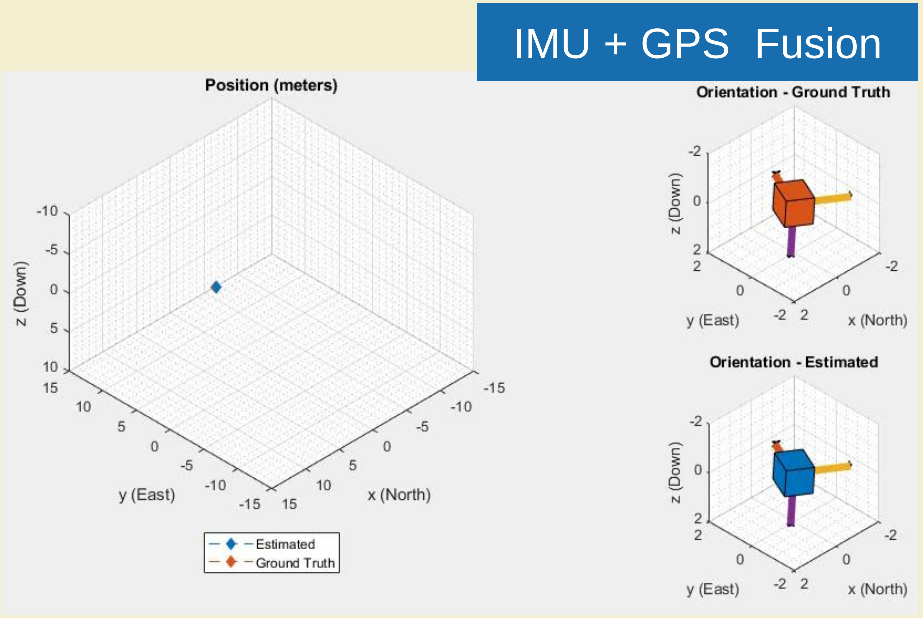
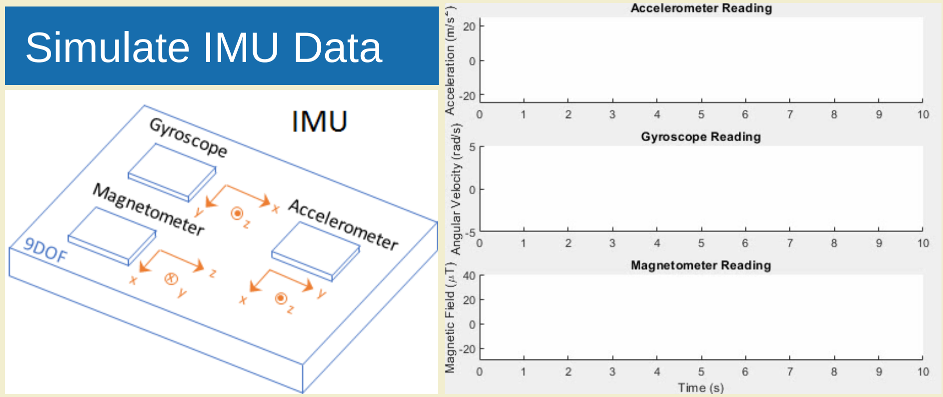


Figure 1
File Edit View Insert Tools Desktop Window Help
Binary Occupancy Grid
Y [meters]
X [meters]
0
5
10
15
20
25
Path Clearance (in meters)
1 m
A
A: Distance from obstacle
Binary Occupancy Grid

Navigation Toolbox

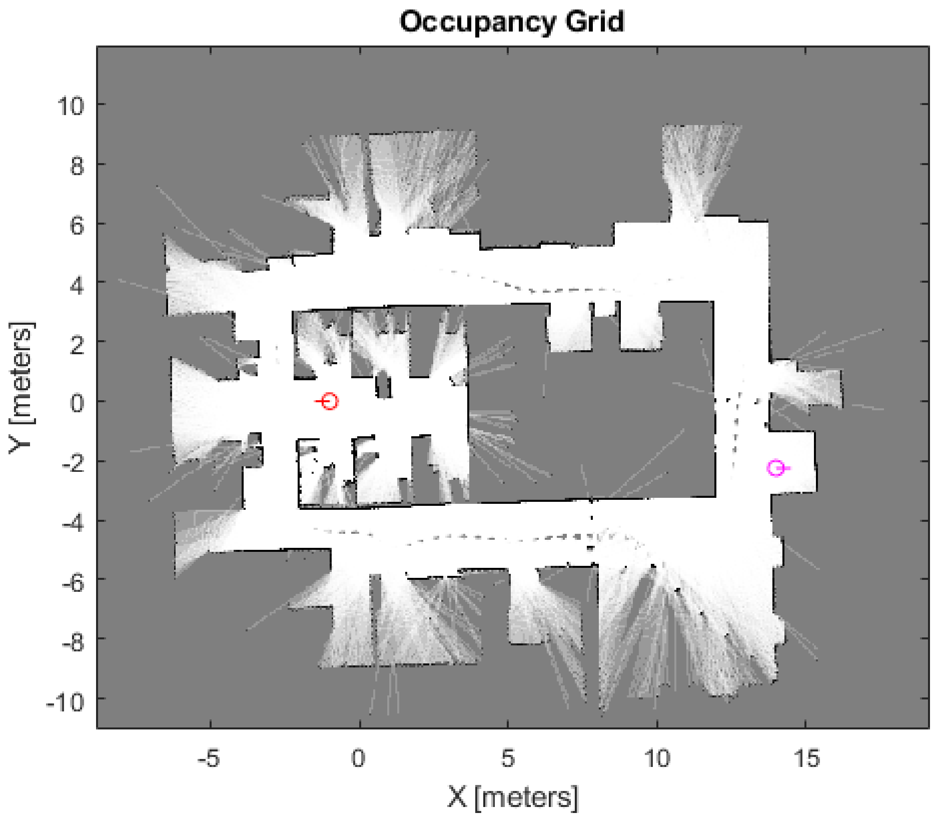
- Mapping and localization
  - 2D and 3D SLAM
  - Egocentric maps
  - SLAM map builder App
- Path planning and Following
  - Algorithms for path planning
  - Planner interface
  - Path metrics
  - Path following and controls
- **Sensor modeling and simulation**
  - **IMU, GPS, INS sensors**

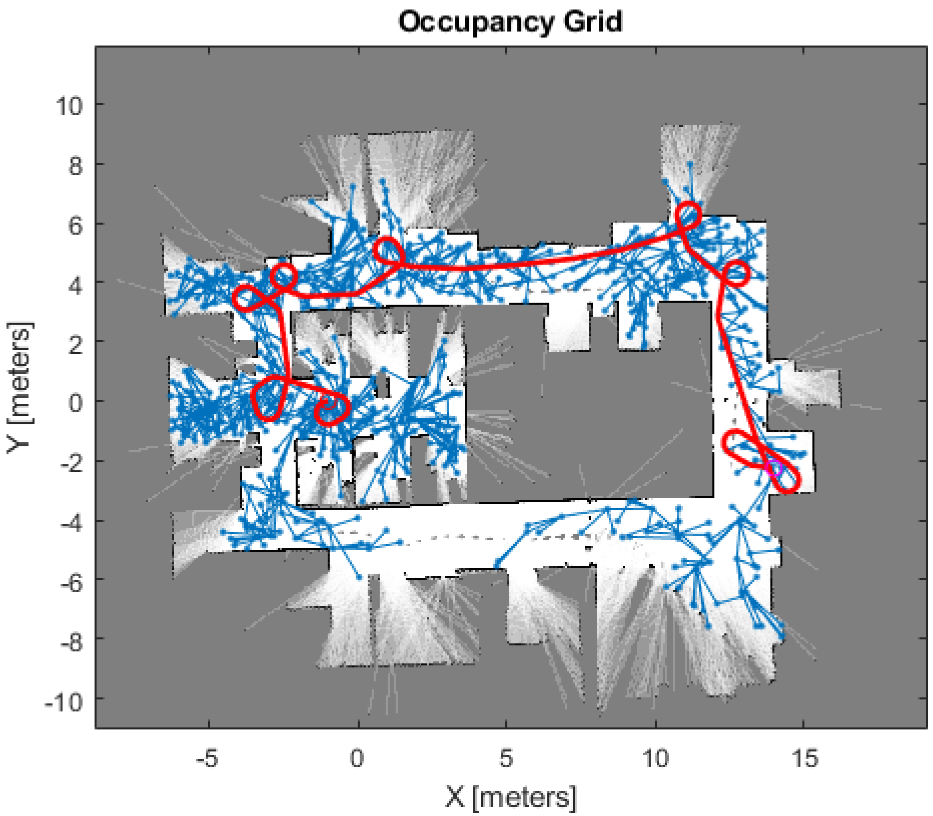

IMU + GPS Fusion


Simulate IMU Data

Example – Plan Mobile Robot Paths using RRT

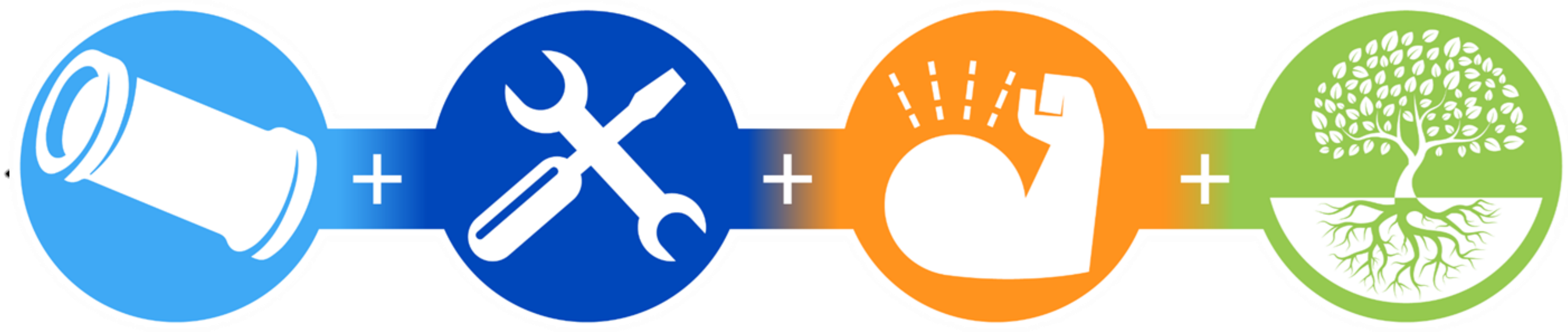
- Load an existing occupancy map of a small office space
- Specify the state space of the vehicle
- Plan a path for a vehicle





ROS – A Distribution in Software for Automation

- Open Source
- Established to prevent re-inventing the wheel
- Maintained by Open Robotics
- Reusable Software Components
- >1,000,000 user downloads/mo



Plumbing Tools Capabilities Ecosystem

Why ROS? Growth and Adoption of ROS

ROS

10 years huge growth

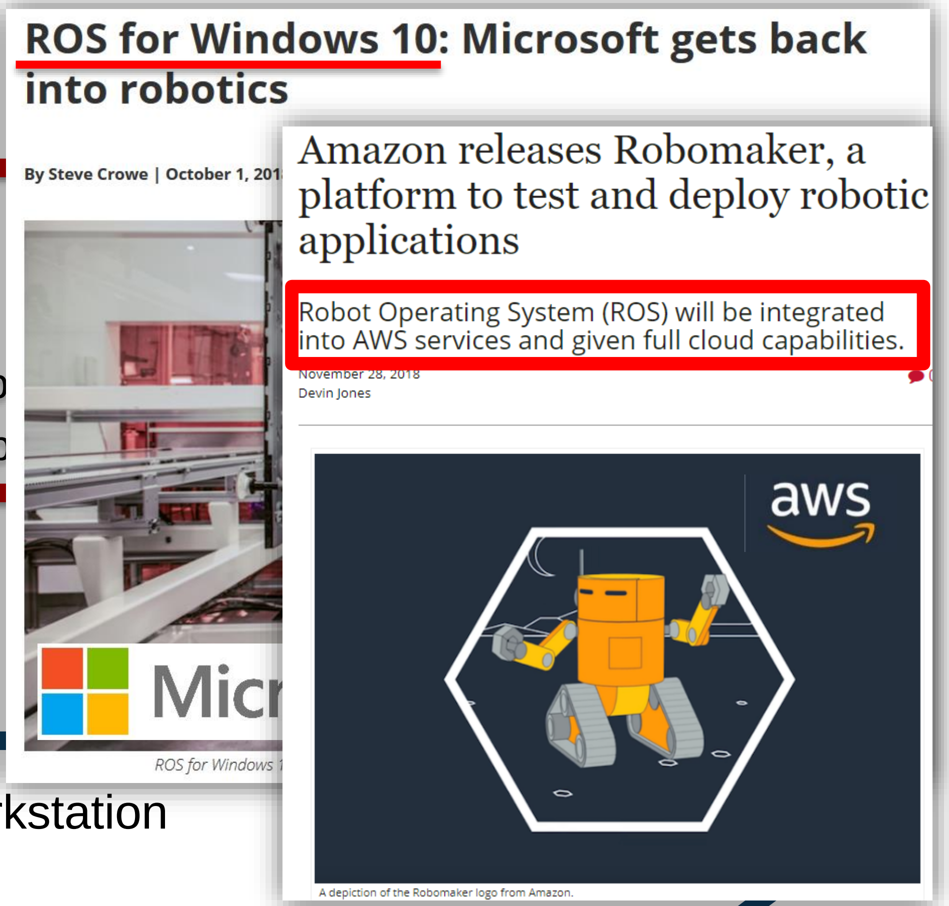
- Plenty of development tools
- Active community (ROS wiki p
- 5,000 packages, 18,000 wiki p

ROS1_Bridge

Completely Re-Written

2

- Run on range of systems: embedded to workstation
- For use in real-time systems
- For safety- and mission-critical applications and productions



ROS for Windows 10: Microsoft gets back into robotics

By Steve Crowe | October 1, 201

ROS for Windows 1

Amazon releases Robomaker, a platform to test and deploy robotic applications

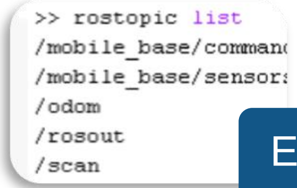
Robot Operating System (ROS) will be integrated into AWS services and given full cloud capabilities.

November 28, 2018

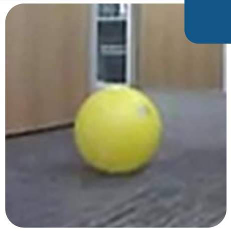
Devin Jones

A depiction of the Robomaker logo from Amazon.

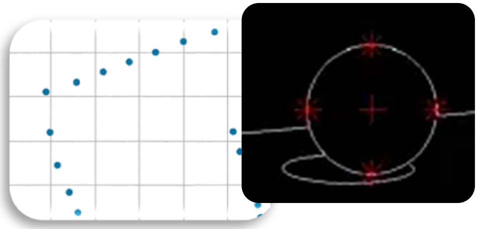
Development Workflow for ROS-based Applications



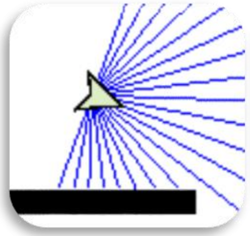
```
>> rostopic list
/mobile_base/comman
/mobile_base/sensors
/odom
/rosout
/scan
```



Explore ROS Network Interface



Develop Algorithm

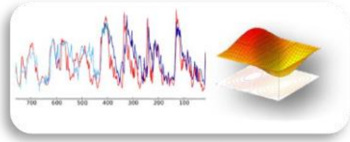


Test and Refine in Simulation

Deploy a ROS node to the Robot

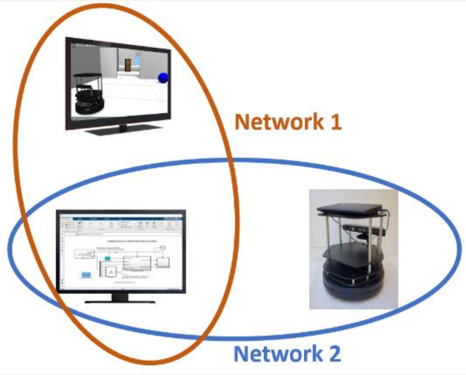
Test and Refine on Real Robot

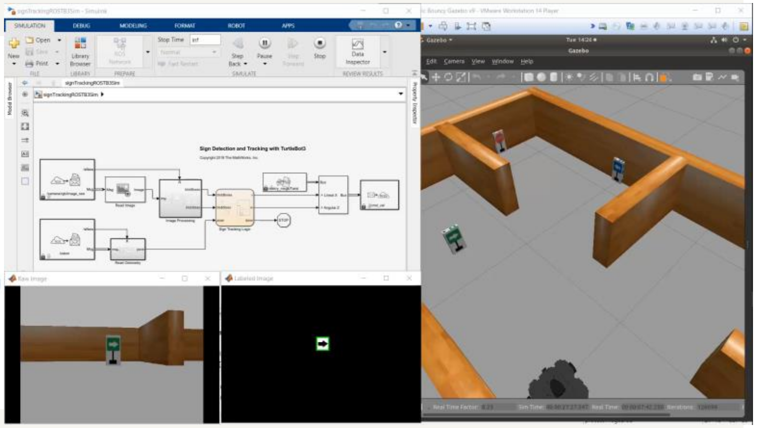
Performance Analysis

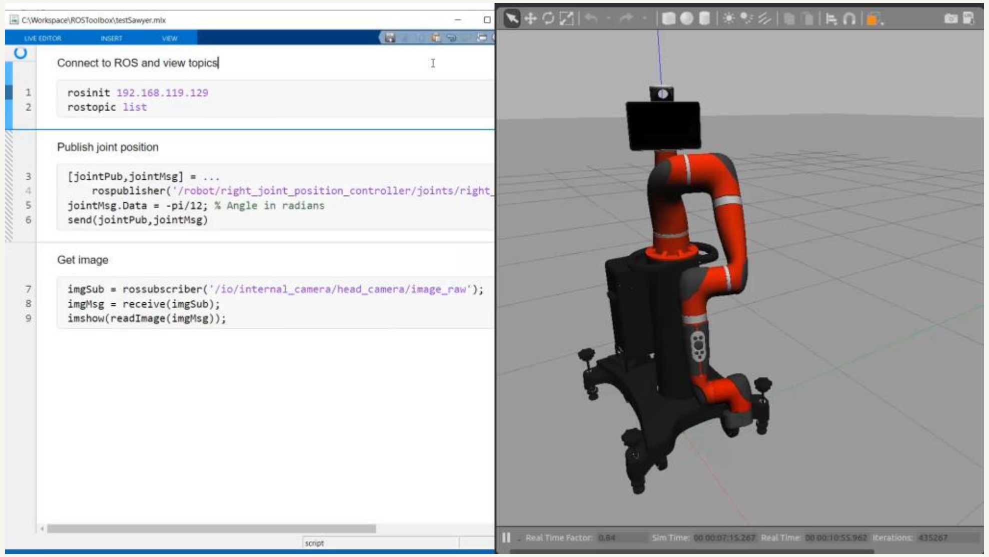


ROS Toolbox

- **ROS network and communication**
  - **Live connectivity from MATLAB and Simulink to ROS and ROS2**







C:\Workspace\ROSToolbox\testSawyer.mlx

Connect to ROS and view topics

```
rosinit 192.168.119.129
rostopic list
```

Publish joint position

```
[jointPub,jointMsg] = ...
    rospublisher('/robot/right_joint_position_controller/joints/right_
jointMsg.Data = -pi/12; % Angle in radians
send(jointPub,jointMsg)
```

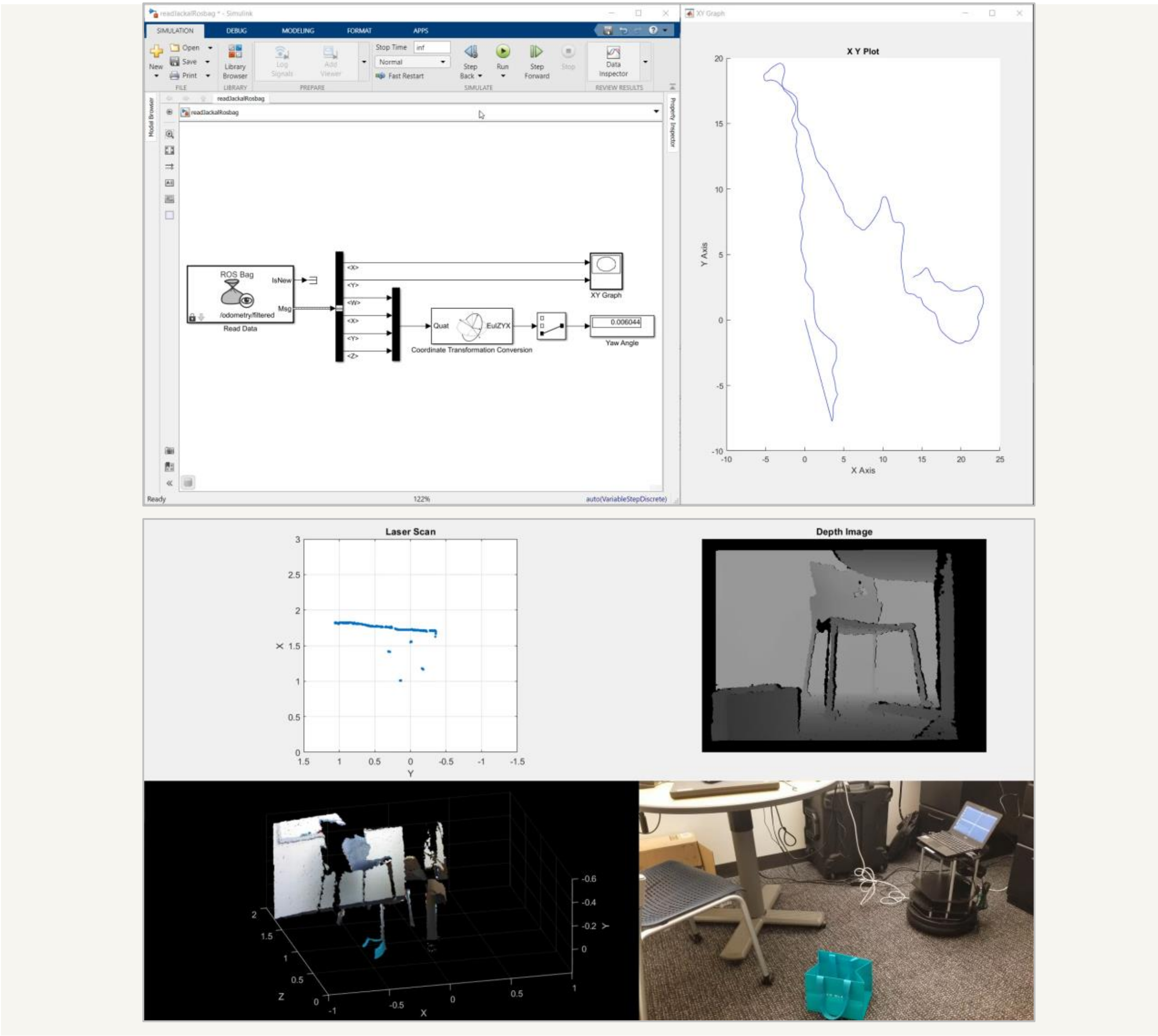
Get image

```
imgSub = rossubscriber('/io/internal_camera/head_camera/image_raw');
imgMsg = receive(imgSub);
imshow(readImage(imgMsg));
```

ROS Toolbox

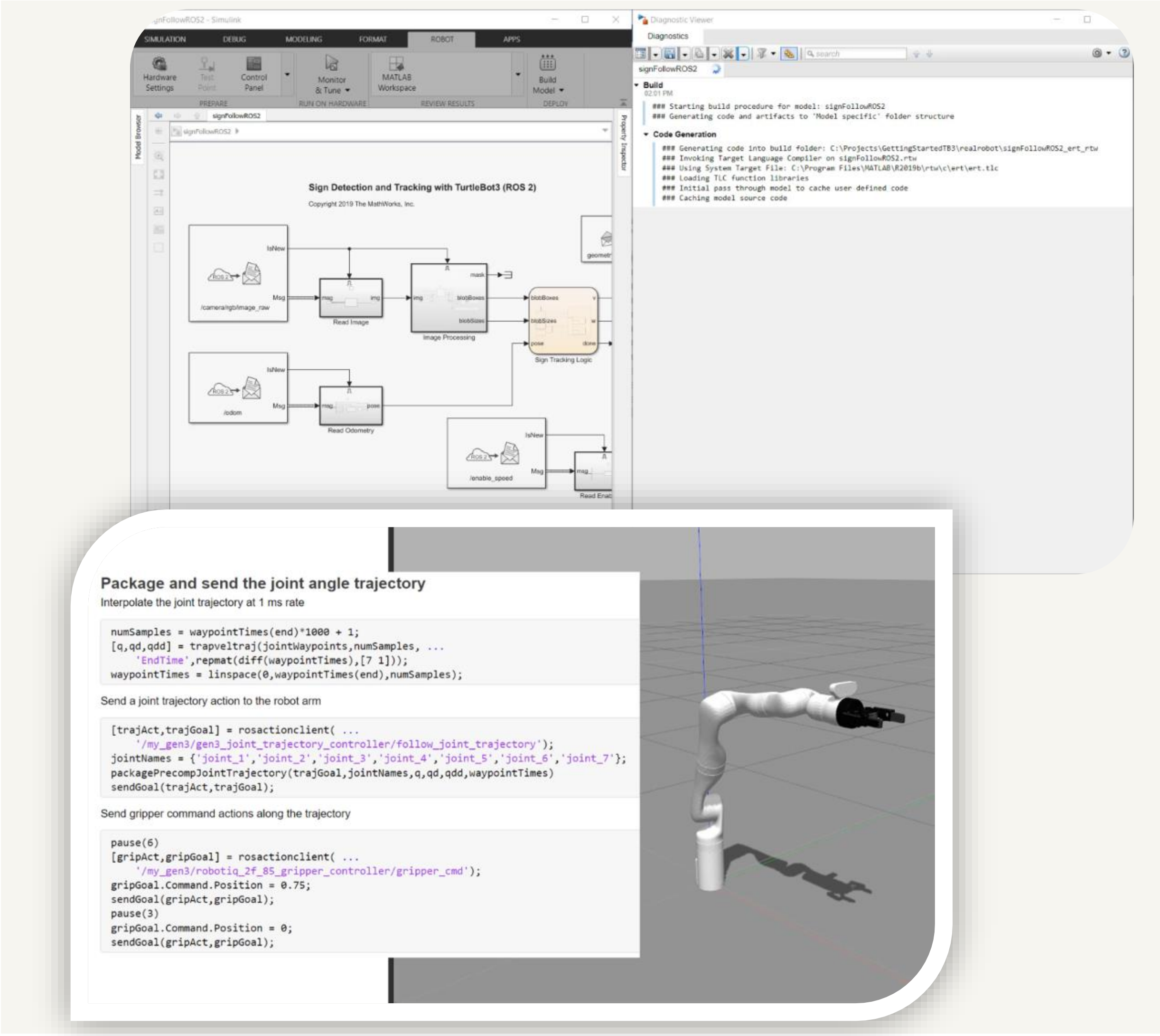
- ROS network and communication
  - Live connectivity from MATLAB and Simulink to ROS and ROS2

- **ROS Message**
  - **rosbag data import and playback**
  - **Specialized ROS message**



ROS Toolbox

- ROS network and communication
  - Live connectivity from MATLAB and Simulink to ROS and ROS2
- ROS Message
  - rosbag data import and playback
  - Specialized ROS message
- **ROS node generation**
  - **Node generation from Simulink for prototyping and deploying autonomous systems**



Sign Detection and Tracking with TurtleBot3 (ROS 2)

Build
02:01 PM

```
### Starting build procedure for model: signFollowROS2
### Generating code and artifacts to 'Model specific' folder structure
```

Code Generation

```
### Generating code into build folder: C:\Projects\GettingStartedTB3\realrobot\signFollowROS2_ert_rtw
### Invoking Target Language Compiler on signFollowROS2.rtw
### Using System Target File: C:\Program Files\MATLAB\R2019b\rtw\c\ert\ert.tlc
### Loading TLC function libraries
### Initial pass through model to cache user defined code
### Caching model source code
```

Package and send the joint angle trajectory

Interpolate the joint trajectory at 1 ms rate

```
numSamples = waypointTimes(end)*1000 + 1;
[q,qd,qdd] = trapveltraj(jointWaypoints,numSamples, ...
    'EndTime',repmat(diff(waypointTimes),[7 1]));
waypointTimes = linspace(0,waypointTimes(end),numSamples);
```

Send a joint trajectory action to the robot arm

```
[trajAct,trajGoal] = rosactionclient( ...
    '/my_gen3/gen3_joint_trajectory_controller/follow_joint_trajectory');
jointNames = {'joint_1','joint_2','joint_3','joint_4','joint_5','joint_6','joint_7'};
packagePrecompJointTrajectory(trajGoal,jointNames,q,qd,qdd,waypointTimes)
sendGoal(trajAct,trajGoal);
```

Send gripper command actions along the trajectory

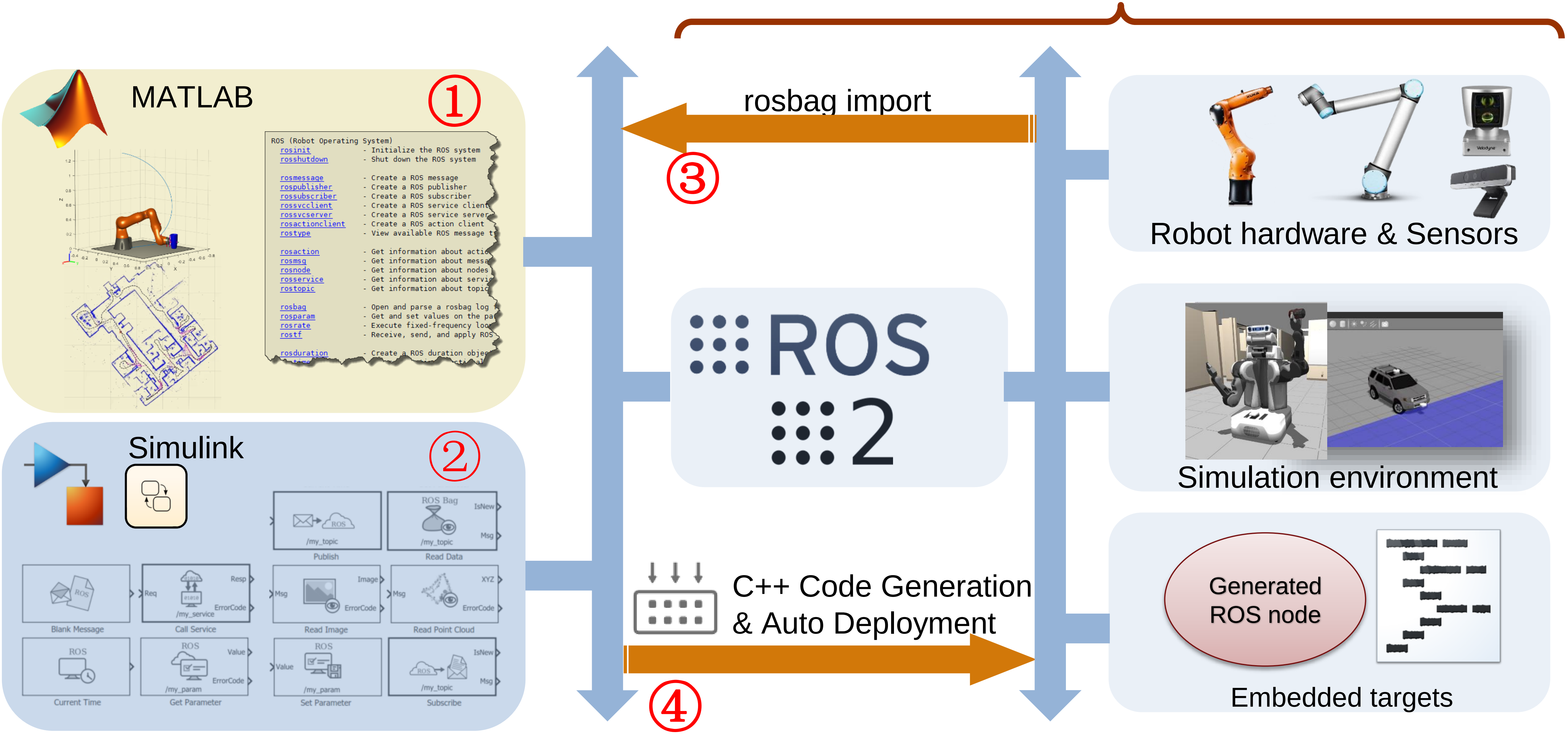
```
pause(6)
[gripAct,gripGoal] = rosactionclient( ...
    '/my_gen3/robotiq_2f_85_gripper_controller/gripper_cmd');
gripGoal.Command.Position = 0.75;
sendGoal(gripAct,gripGoal);
pause(3)
gripGoal.Command.Position = 0;
sendGoal(gripAct,gripGoal);
```

MATLAB/Simulink ROS Functionality

	ROS	2	ROS ROS 2 ROS Common
	• Topic – Pub / Sub • Service – Server / Client • Action – Client • Parameter Server – Get/Set • Custom Message • rosbag read	• Topic – Pub / Sub • Custom Message	• Read Data • Read / Write Image • Read Point Cloud • Read Occupancy Map
	• Topic – Pub / Sub • Service – Call • Parameter – Get / Set • ROS Time • rosbag playback • Code Generation	• Topic – Pub / Sub • Code Generation	• Read Data • Read Image • Read Point Cloud

ROS Toolbox enables you to communicate with a ROS

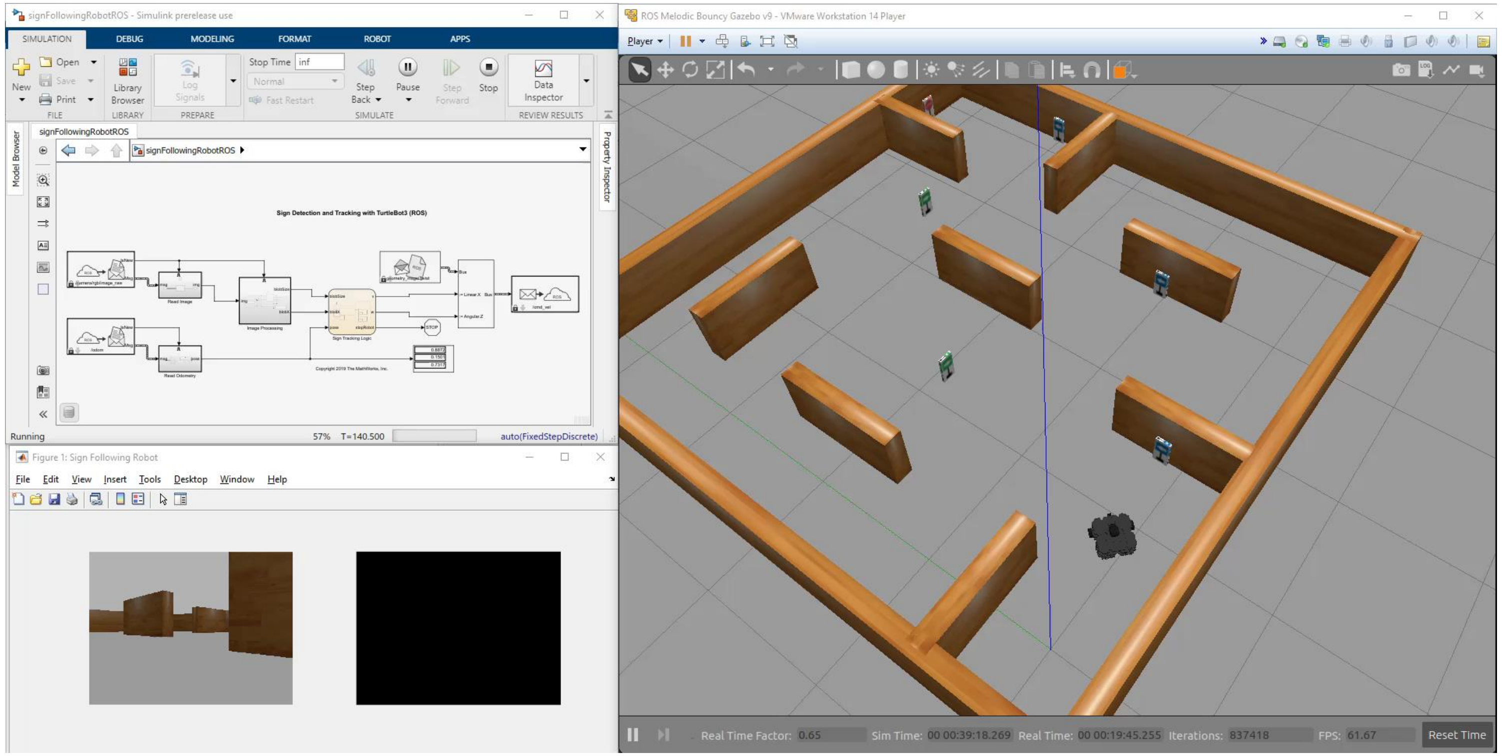
Traditional ROS users



```
ROS (Robot Operating System)
  rosinit             - Initialize the ROS system
  rosshutdown         - Shut down the ROS system

  rosmessage          - Create a ROS message
  rospublisher        - Create a ROS publisher
  rossubscriber       - Create a ROS subscriber
  rossvcclient        - Create a ROS service client
  rossvcserver        - Create a ROS service server
  rosactionclient     - Create a ROS action client
  rostype             - View available ROS message t

  rosaction           - Get information about actio
  rosmsg              - Get information about messa
  rosnode             - Get information about nodes
  rosservice          - Get information about servic
  rostopic            - Get information about topic

  rosbag              - Open and parse a rosbag log f
  rosparam            - Get and set values on the pa
  rosrate             - Execute fixed-frequency loo
  rostf               - Receive, send, and apply ROS

  rosduration         - Create a ROS duration objec
```

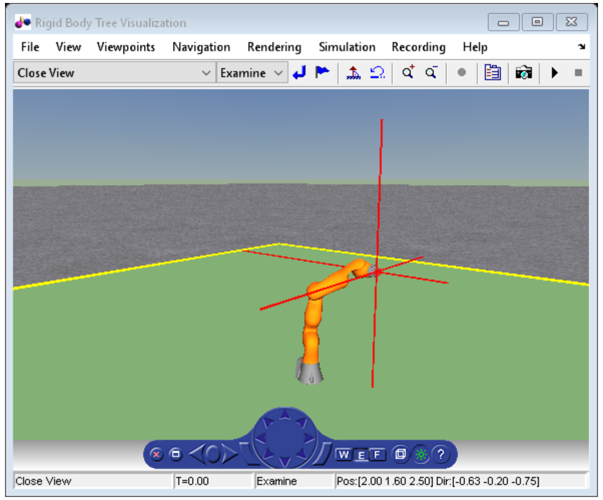
Example – Sign-following Robot

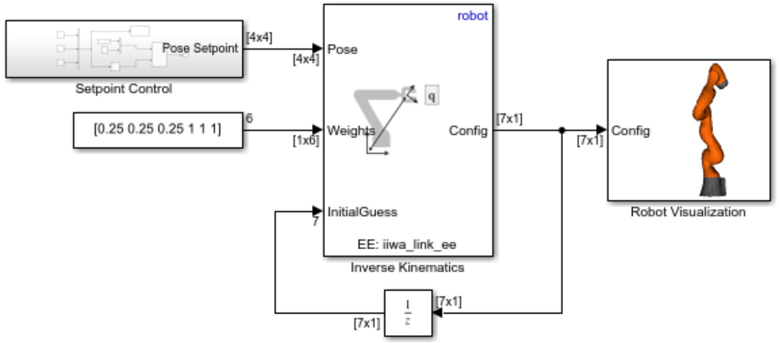
- Detect the color of the sign and send the velocity commands to turn the robot
- Connect with ROS-enabled simulator, i.e., Gazebo
- And connect with hardware

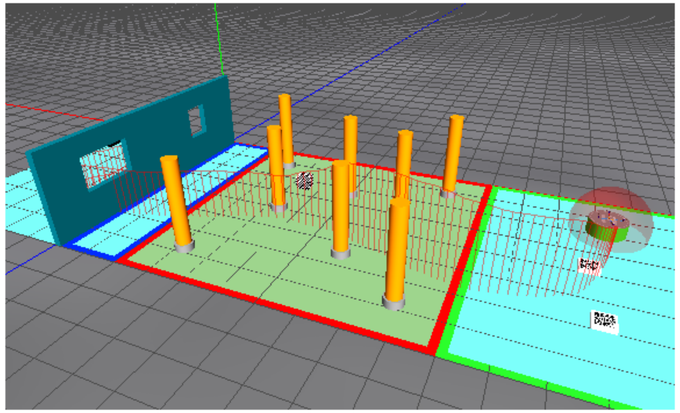


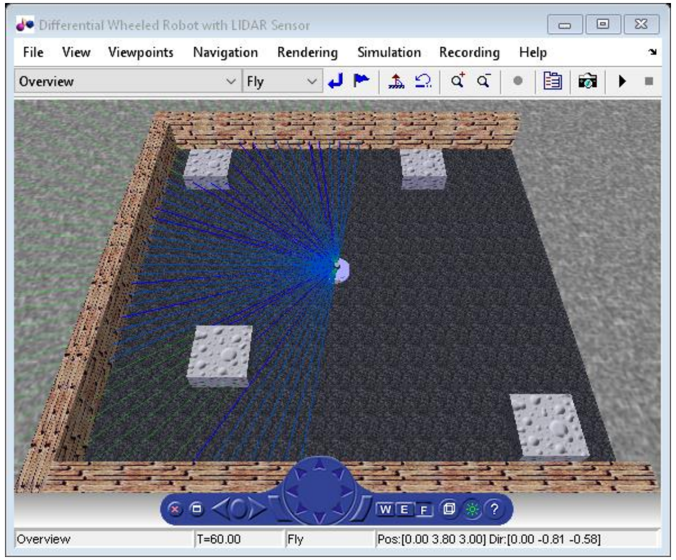
Robotics with Simulink 3D Animation

- Import Robot model
  - URDF – vrimport
  - axis_ prefix – simple manipulation
- Detect Object Collisions
  - LinePickSensor – LIDAR
    - Mapping, SLAM
  - PrimitivePickSensor
    - Bounding box
- VR RigidBodyTree block
  - RST RBT object
  - only joints angle needed


Rigid Body Tree Visualization
File View Viewpoints Navigation Rendering Simulation Recording Help
Close View
Examine
W E F
Close View
T=0.00
Examine
Pos:[2.00 1.60 2.50] Dir:[-0.63 -0.20 -0.75]






Differential Wheeled Robot with LIDAR Sensor
File View Viewpoints Navigation Rendering Simulation Recording Help
Overview
Fly
W E F
Overview
T=60.00
Fly
Pos:[0.00 3.80 3.00] Dir:[0.00 -0.81 -0.58]

Ďakujem za pozornosť