MATLAB EXPO 2016

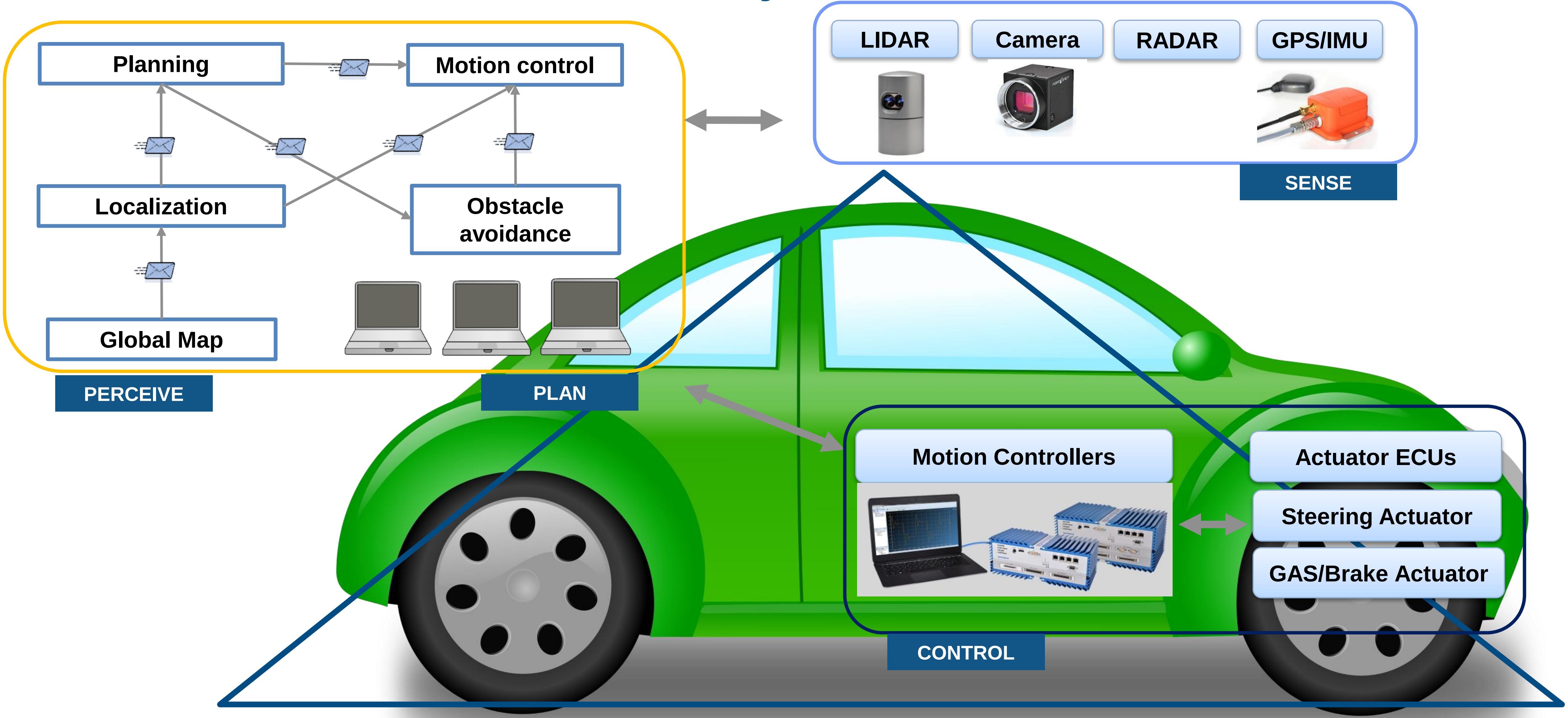
Robotics Development Workflow with MATLAB and Simulink

Carlos Santacruz-Rosero, Ph.D
Sr Application Engineer - Robotics

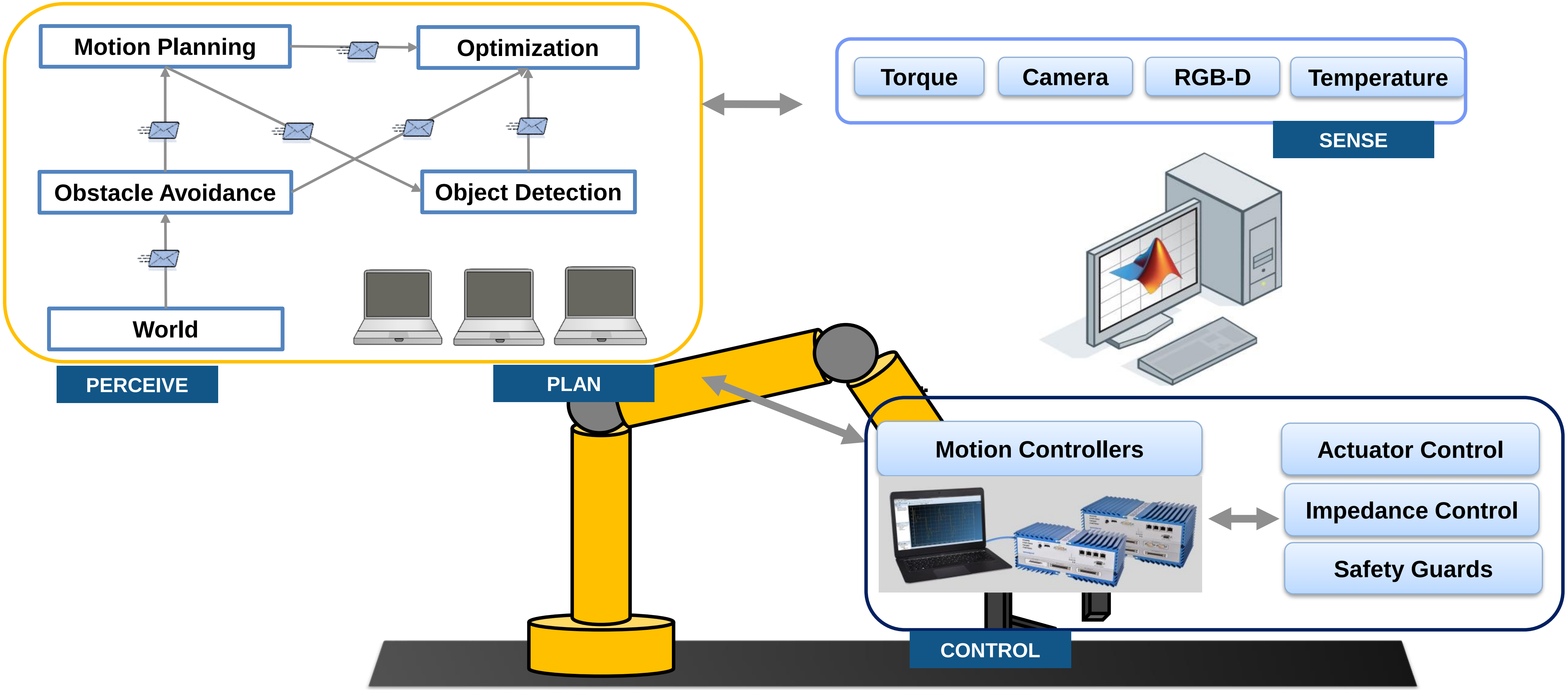
Agenda

- Introduction
- Advanced Robotics Systems
- Robotics Development Workflow with MATLAB and Simulink
- Takeaways

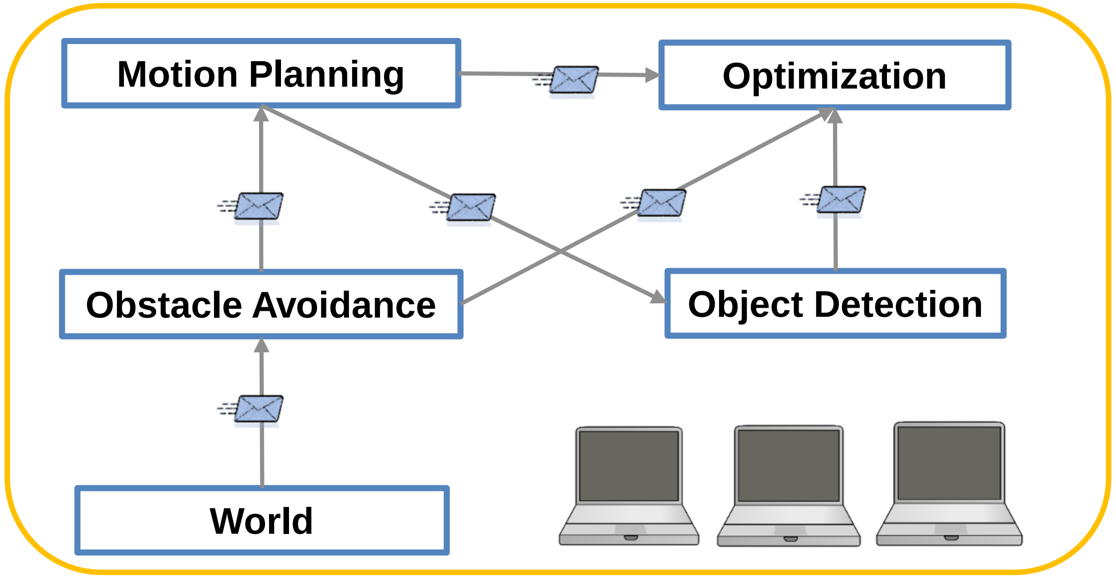
Car as an Advanced Robotics System



Collaborative Robot as an Advanced Robotics System



Architecture - Advanced Robotics System



Middleware

Data synchronization

CPU/GPU Cluster, Server, Cloud

Real-time requirements

Microcontroller, FPGA, Real-time PC

ol

Technology Solutions for Autonomous Systems

CONTROL

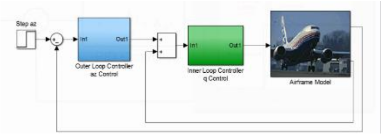
SENSE

PERCEIVE

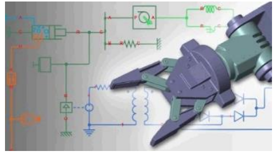
PLAN

CONNECT

Control System Tbx

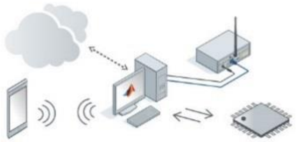


Simscape Toolboxes

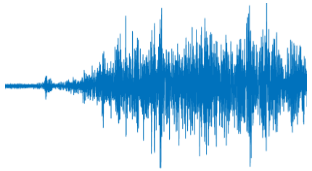


Simulink Real-Time

HW Support Packages

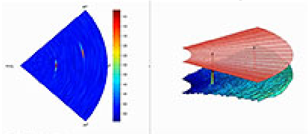


Data Acquisition Tbx

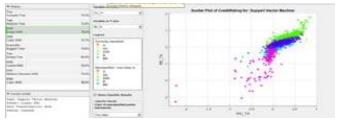


Computer Vision

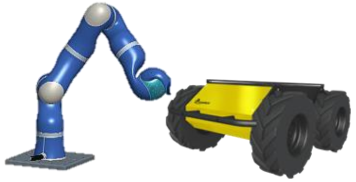
Phased Array



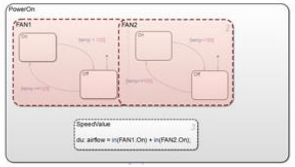
Statistics & Machine Learning



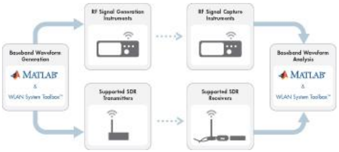
Robotics System Tbx



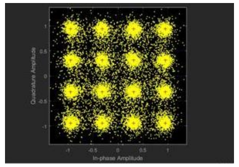
Stateflow



Communications Tbx


RF Signal Generation Instruments
RF Signal Capture Instruments
Baseband Waveform Generation
MATLAB
WLAN System Toolbox
Supported SDR Transmitters
Supported SDR Receivers
Baseband Waveform Analysis
MATLAB
WLAN System Toolbox

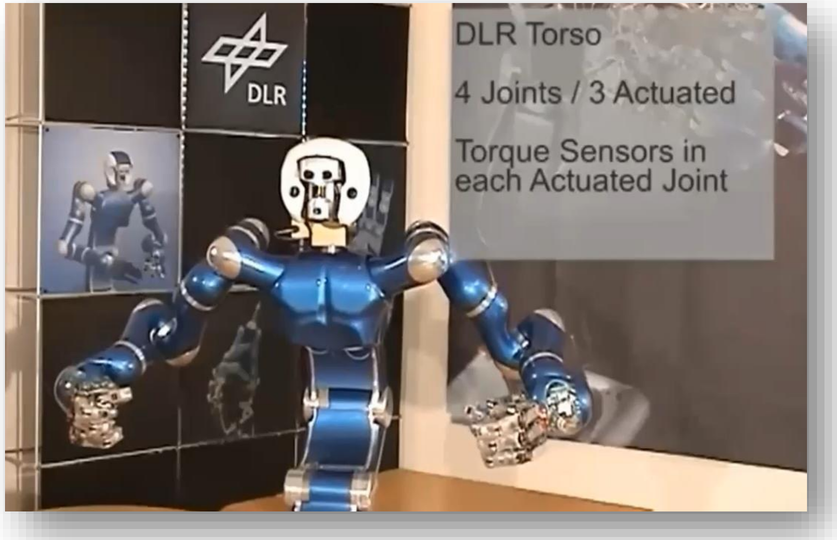
WLAN System Toolbox



Robotics System Tbx

ROS

Success Stories



Technology Solutions for Autonomous Systems

CONTROL

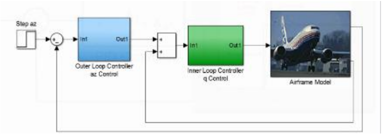
SENSE

PERCEIVE

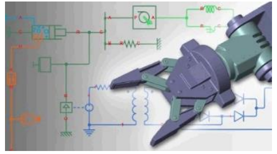
PLAN

CONNECT

Control System Tbx

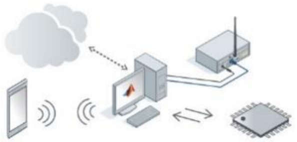


Simscape Toolboxes

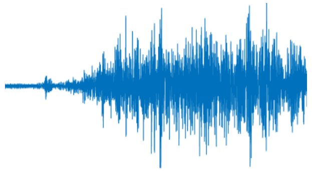


Simulink Real-Time

HW Support Packages

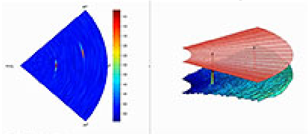


Data Acquisition Tbx

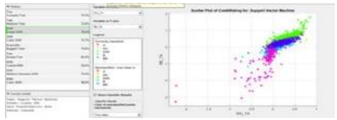


Computer Vision

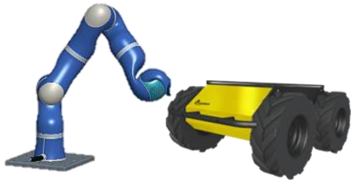
Phased Array



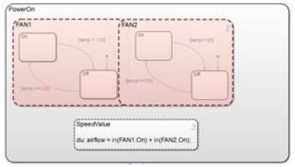
Statistics & Machine Learning



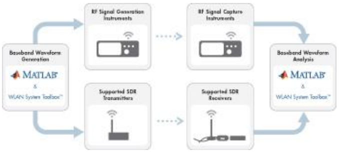
Robotics System Tbx



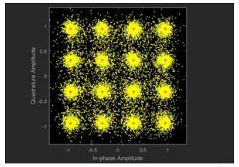
Stateflow



Communications Tbx


Baseband Waveform Generation
RF Signal Generation Instruments
RF Signal Capture Instruments
Baseband Waveform Analysis
MATLAB
WLAN System Toolbox
Supported SDR Transmitters
Supported SDR Receivers

WLAN System Toolbox



Robotics System Tbx

ROS

Robotics System Toolbox

Interface and Deployment to ROS

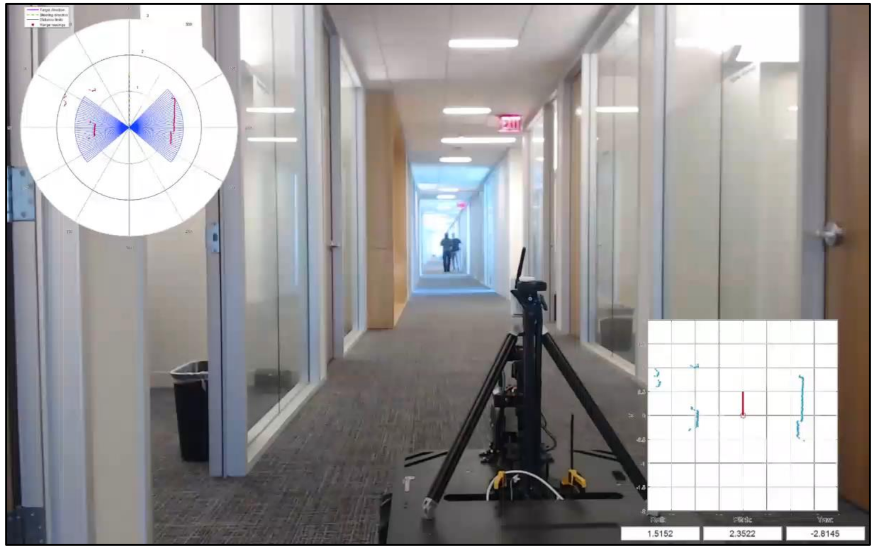
ROS

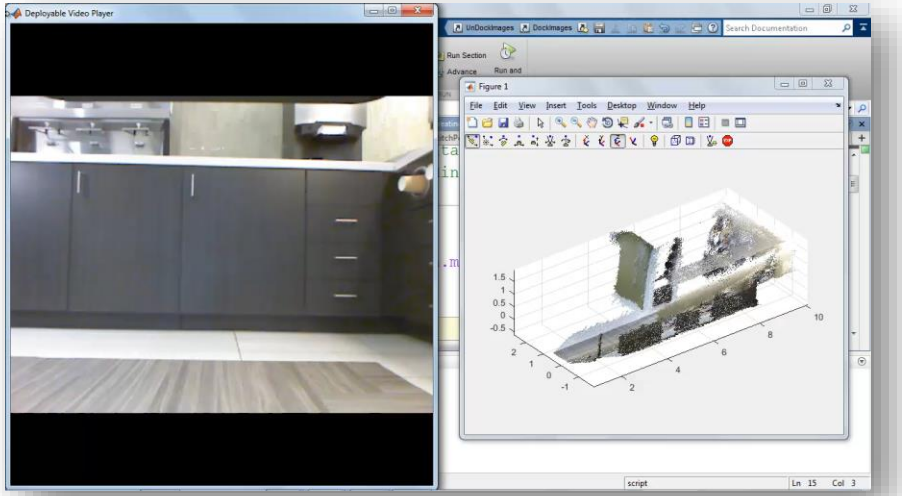
Algorithms for Mobile Robotics

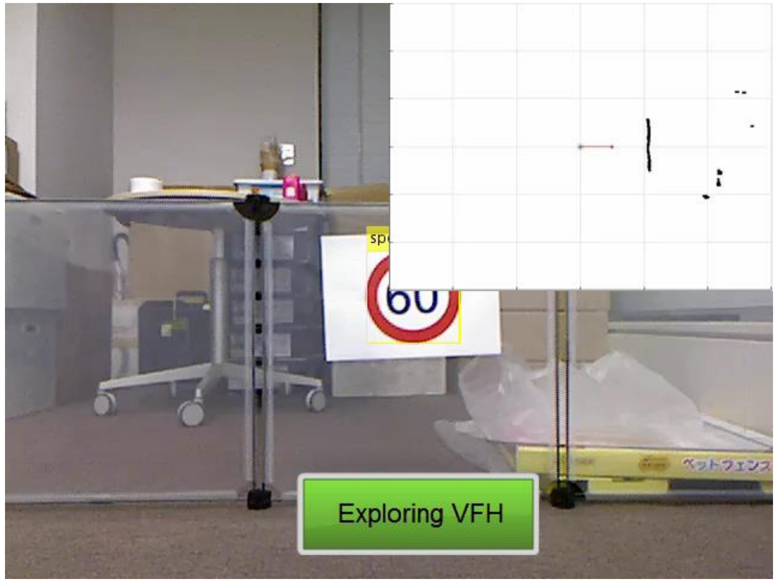
Algorithms for Manipulators and Humanoids

*Environment for **prototyping**, **simulating**, and **deploying** robotics applications*

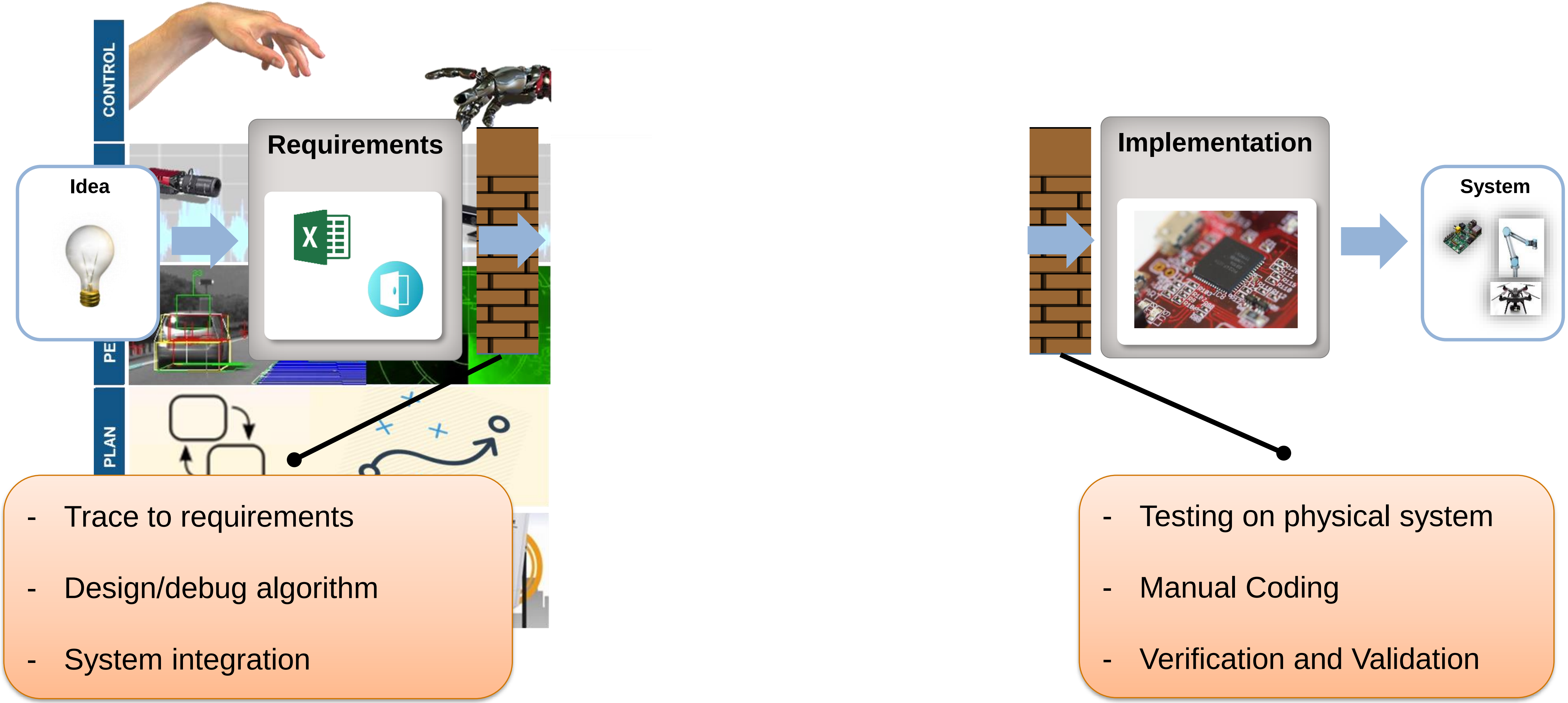
Robotics Applications with Robotics System Toolbox







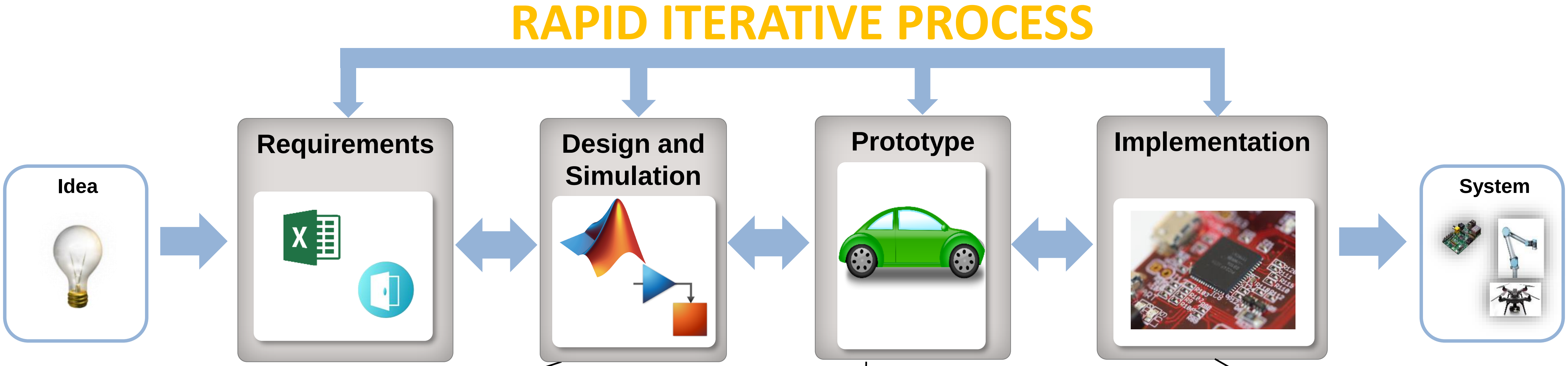
Workflow Convergence is Needed



- Trace to requirements
- Design/debug algorithm
- System integration

- Testing on physical system
- Manual Coding
- Verification and Validation

Robotics Development Workflow with MATLAB and Simulink



RAPID ITERATIVE PROCESS

Idea

Requirements

Design and Simulation

Prototype

Implementation

System

- Built-in algorithms and apps
- System-Level Simulation MBD
- Co-simulation

- C/C++ automatic code generation
- Processor-in-the-loop (PIL)
- Debug C/C++ with MATLAB Engine

Design independent of hardware implementation!

Let’s solve a real problem: Sign Detection System

Clearpath Husky Robot

- ROS Enabled

- Kinect (RGB and Point cloud)

- Hokuyo 2D Lidar

Simulation model in Gazebo

Track and Park

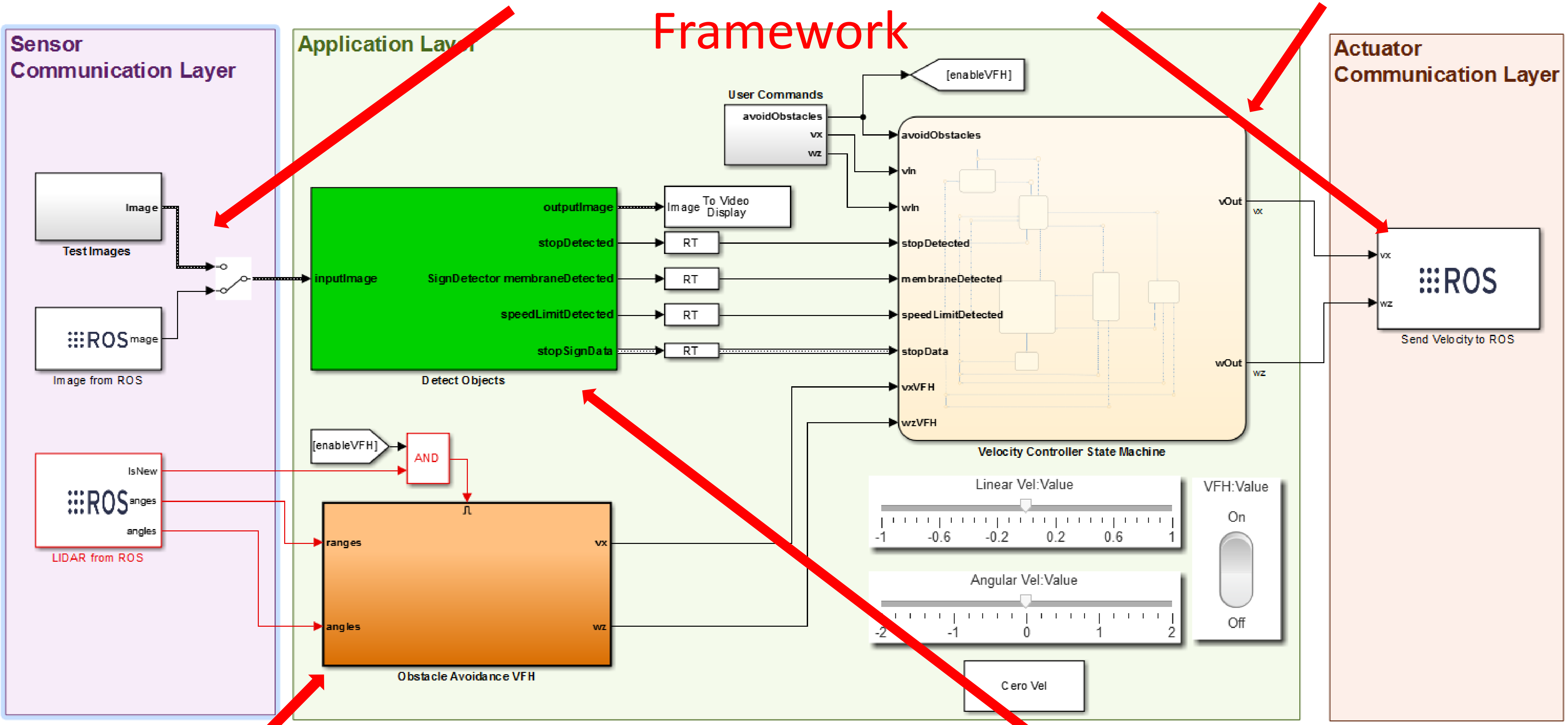
Reduce Speed

Collision Avoidance

System Level Design

ROS as Communication Framework

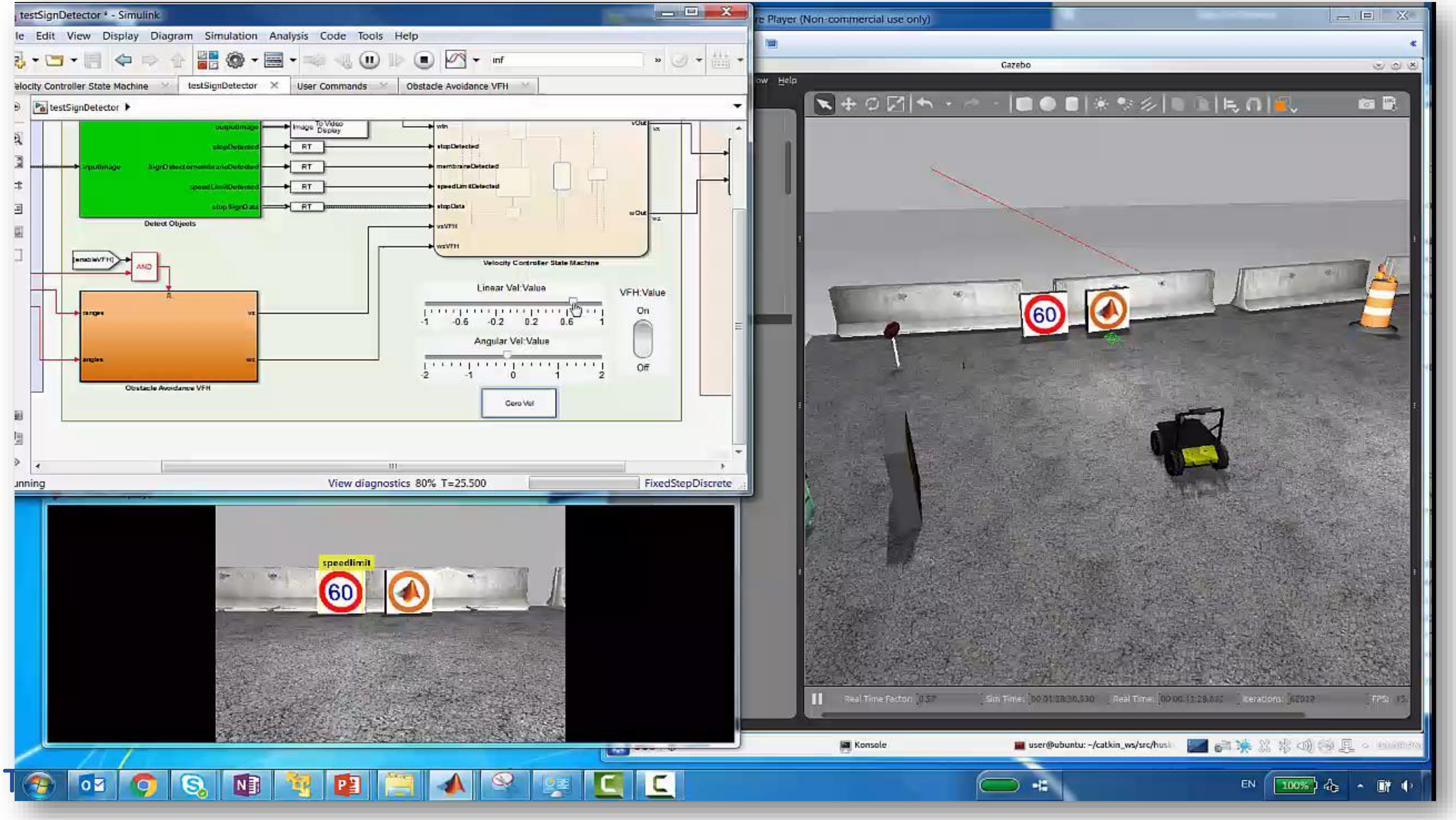
State Machine



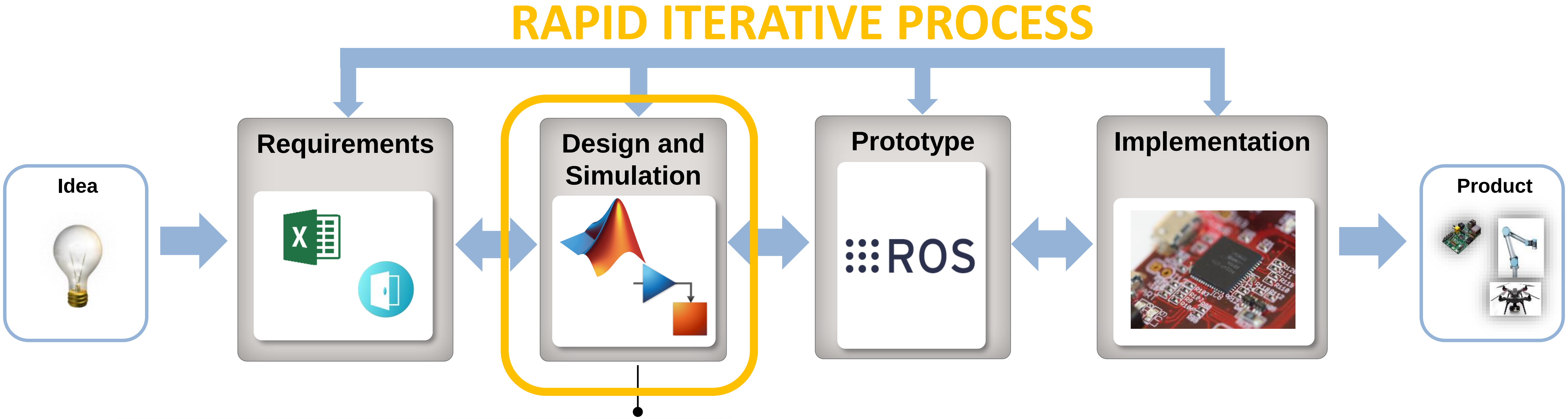
Obstacle Avoidance

Object Classifier

Sign Recognition with Collision Avoidance



Robotics Development Workflow with MATLAB and ROS



RAPID ITERATIVE PROCESS

Idea

Requirements

Design and Simulation

Prototype

Implementation

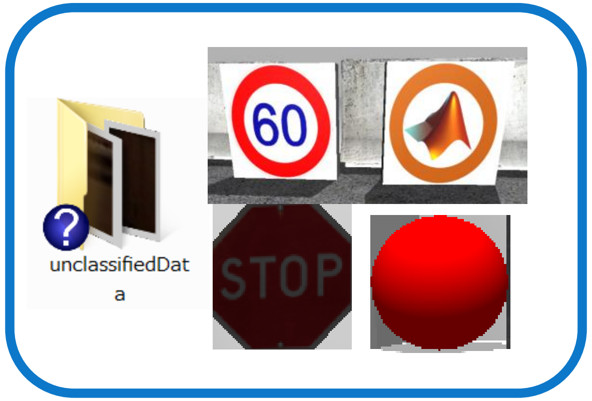
Product

- Import logged data
- Train a classifier
- Test component

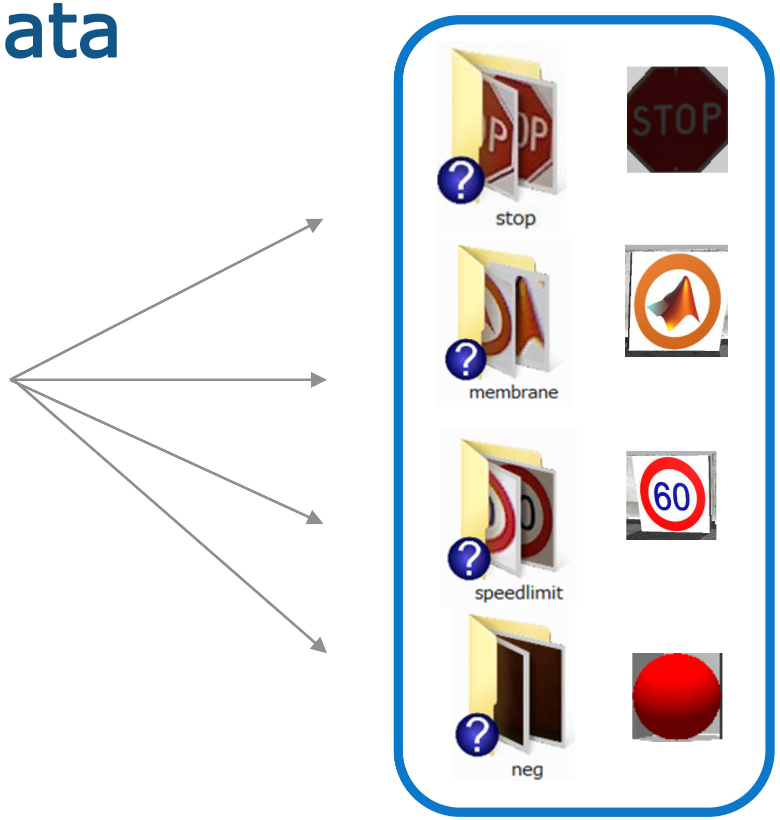
Importing Simulation and Experimental Data

Experimental data or simulation Data

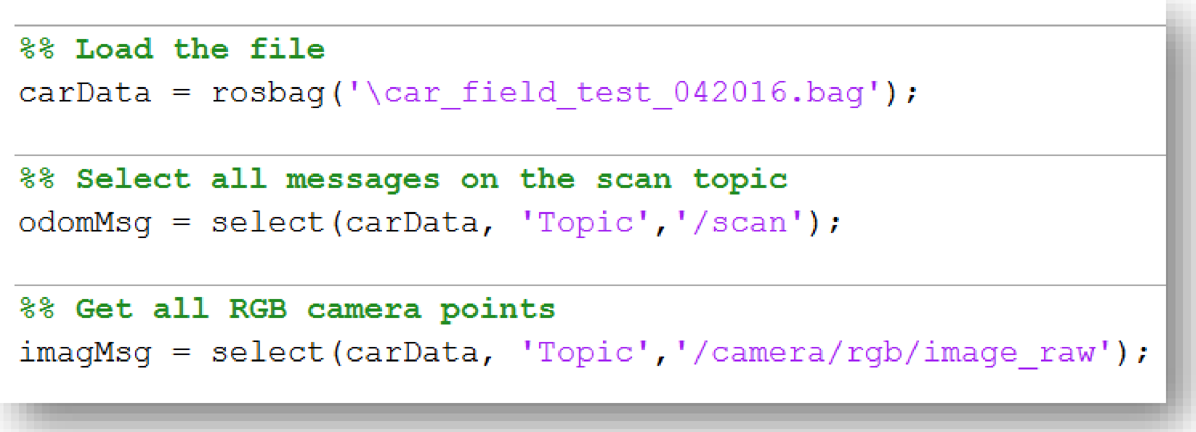
Import ROS Data



Filter your logged field data by topic or time interval



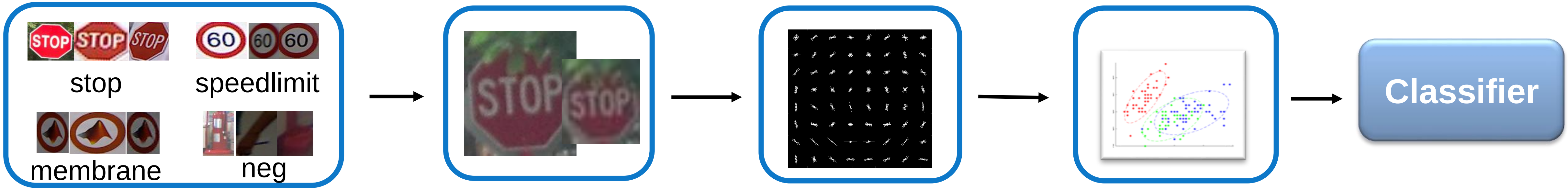
Data ready to design algorithms



```
%% Load the file
carData = rosbag('\car_field_test_042016.bag');

%% Select all messages on the scan topic
odomMsg = select(carData, 'Topic','/scan');

%% Get all RGB camera points
imagMsg = select(carData, 'Topic','/camera/rgb/image_raw');
```

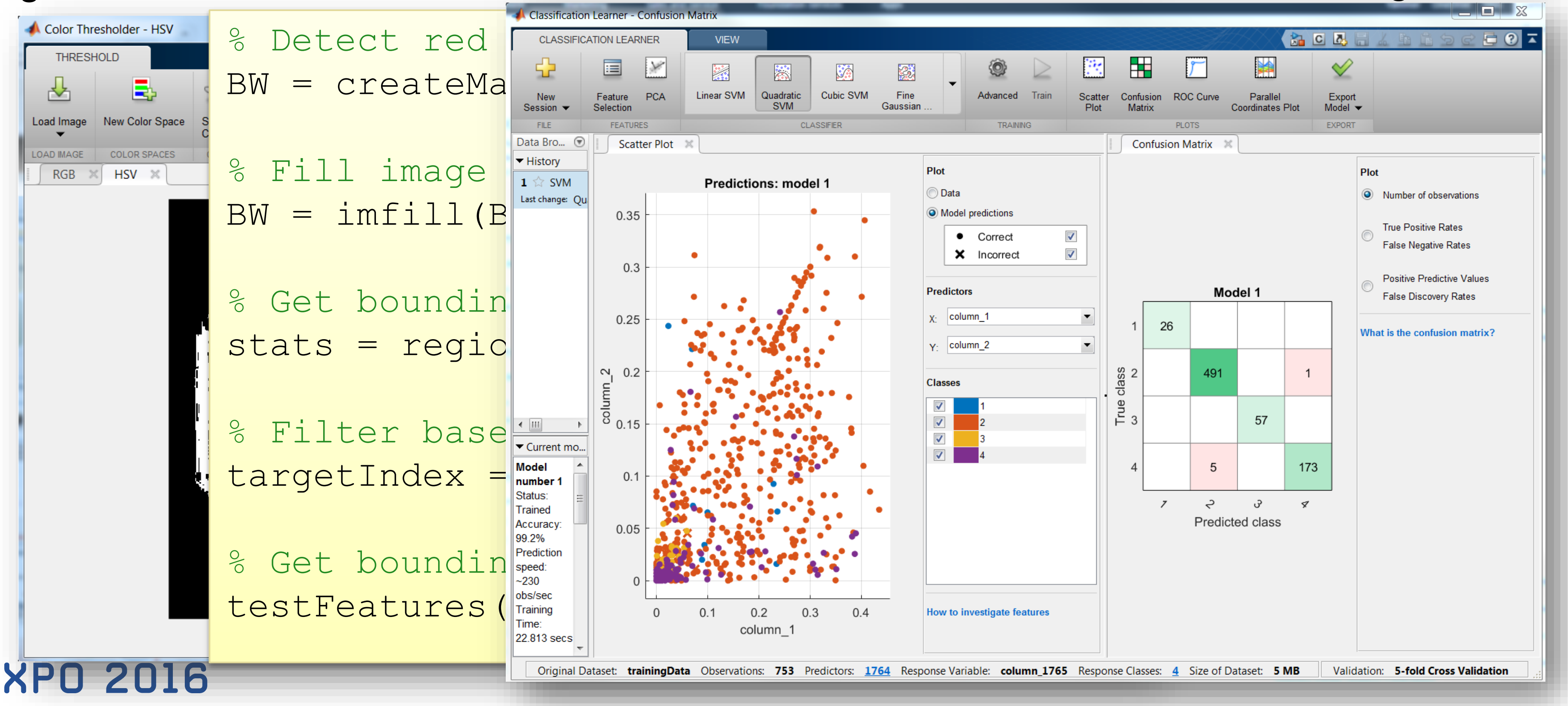
Robotics System Toolbox™

Visualize, Analyze, and Process Data: Classifier



Training data → Preprocessing → Feature Extraction → Training → Classifier

Input



```
% Detect red
BW = createMa

% Fill image
BW = imfill(B

% Get boundin
stats = regio

% Filter base
targetIndex =

% Get boundin
testFeatures(
```

1 ☆ SVM
Last change: Qu

Model number 1
Status: Trained
Accuracy: 99.2%
Prediction speed: ~230 obs/sec
Training Time: 22.813 secs

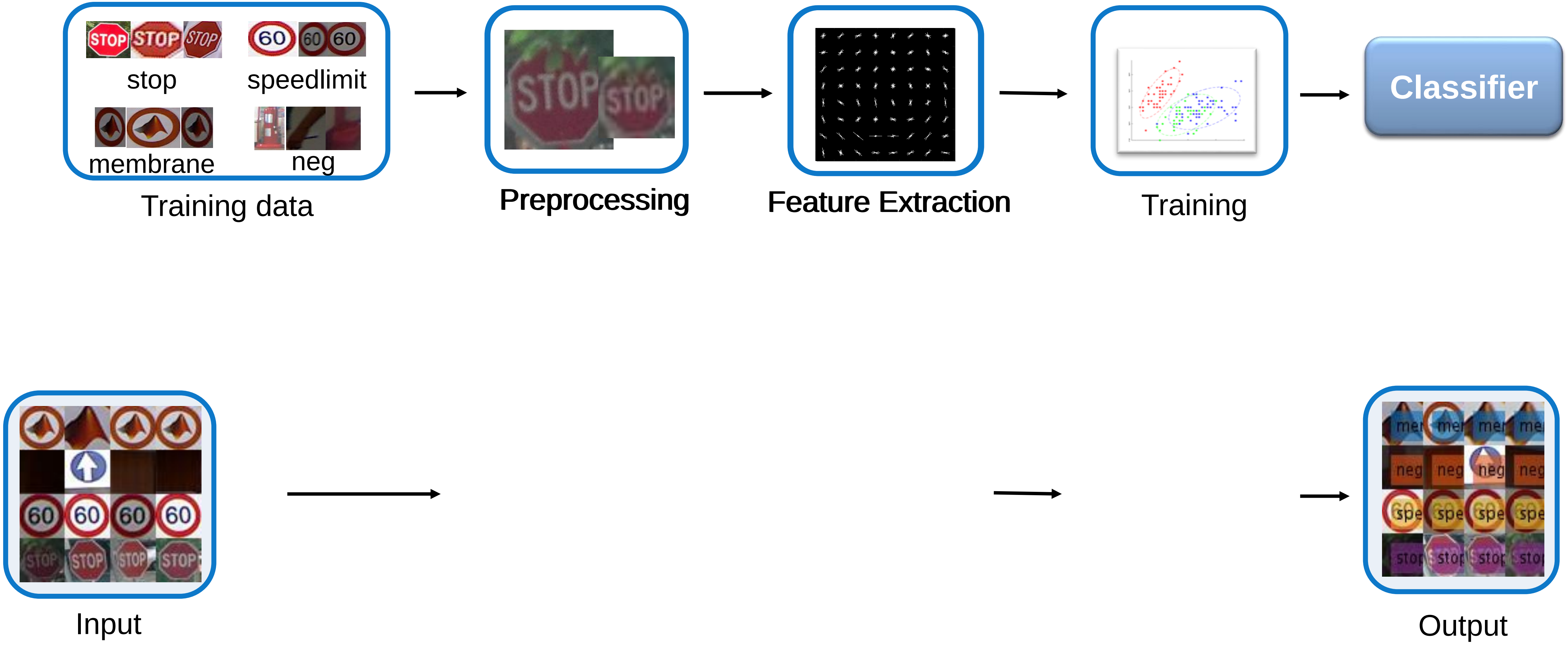
Model 1

True class \ Predicted class	1	2	3	4
1	26			
2		491		1
3			57	
4		5		173

Original Dataset: **trainingData** Observations: **753** Predictors: 1764 Response Variable: **column_1765** Response Classes: 4 Size of Dataset: **5 MB** Validation: **5-fold Cross Validation**

Output

Visualize, Analyze, and Process Data: Classifier



stop

speedlimit

membrane

neg

Training data

Preprocessing

Feature Extraction

Training

Classifier

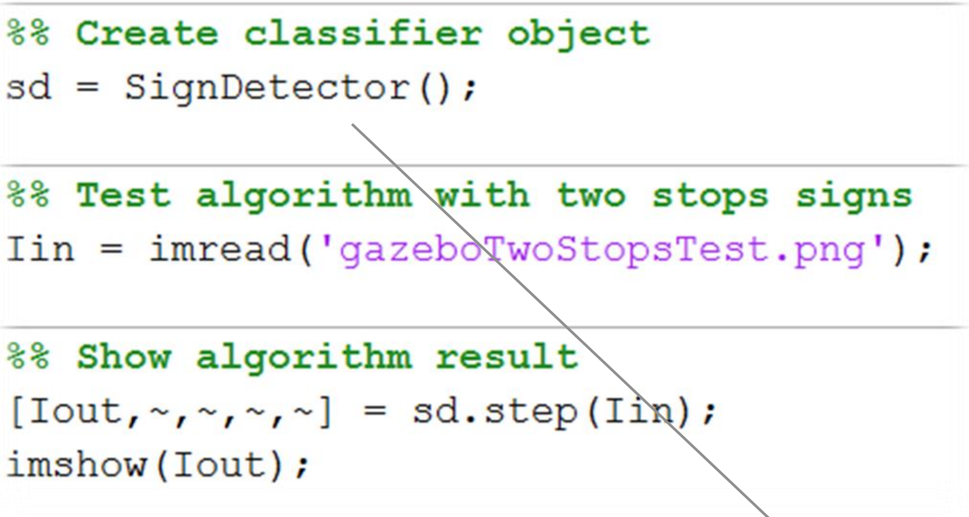
Input

Output

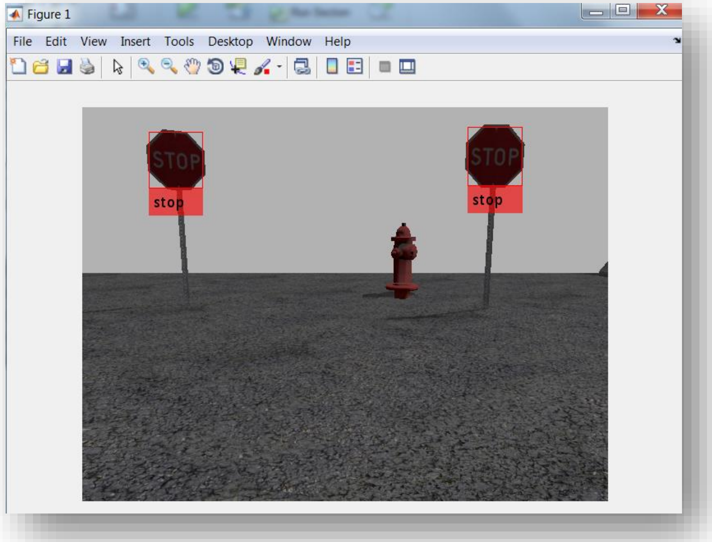
Computer Vision System Toolbox™

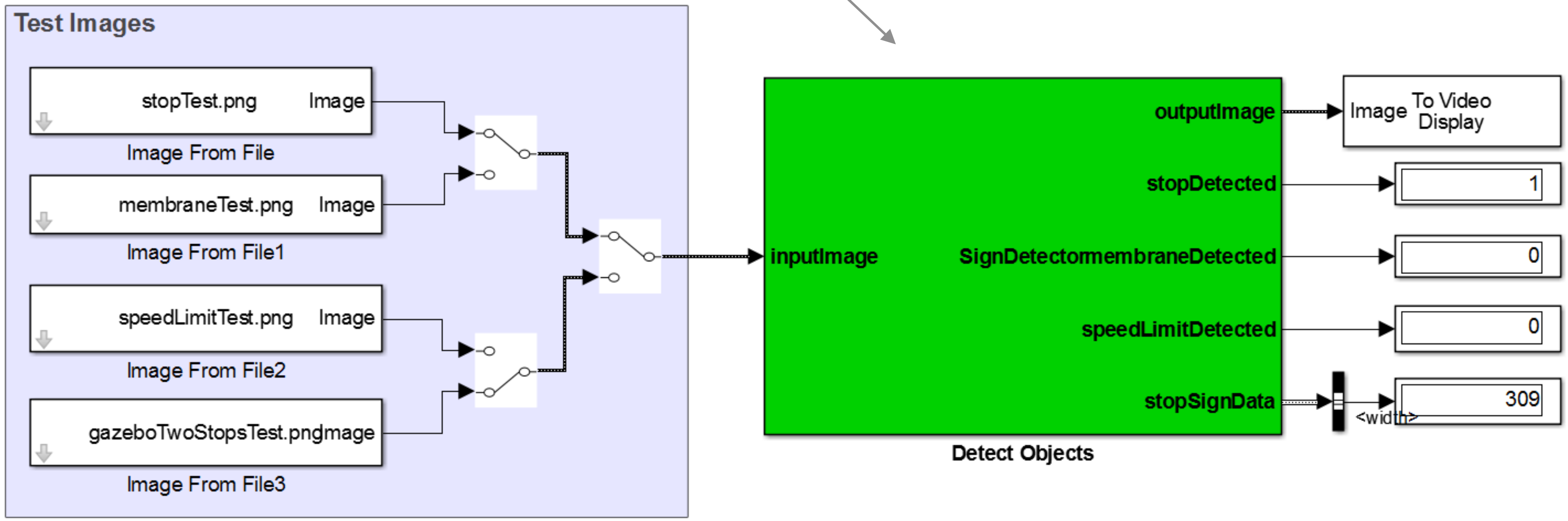
Statistics and Machine Learning Toolbox™

Design and Test Algorithm



```
%% Create classifier object
sd = SignDetector();

%% Test algorithm with two stops signs
Iin = imread('gazeboTwoStopsTest.png');

%% Show algorithm result
[Iout,~,~,~,~] = sd.step(Iin);
imshow(Iout);
```





Test Images

stopTest.png Image
Image From File

membraneTest.png Image
Image From File1

speedLimitTest.png Image
Image From File2

gazeboTwoStopsTest.pngImage
Image From File3

inputImage

outputImage

stopDetected

SignDetectormembraneDetected

speedLimitDetected

stopSignData

Detect Objects

Image To Video Display

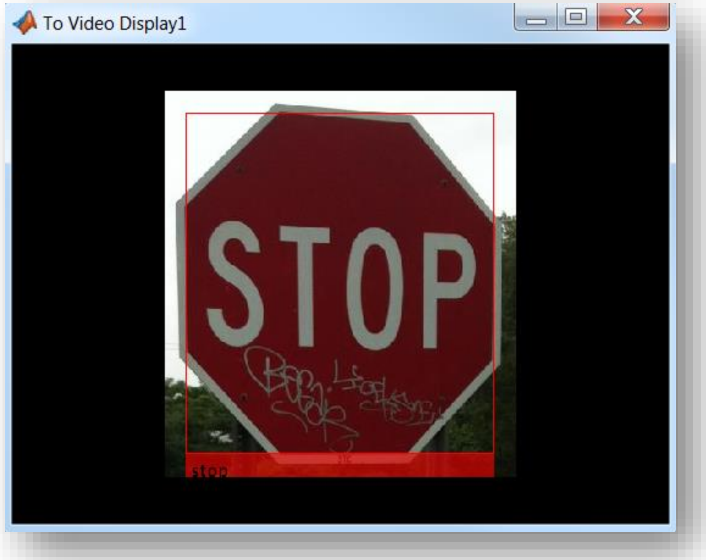
1

0

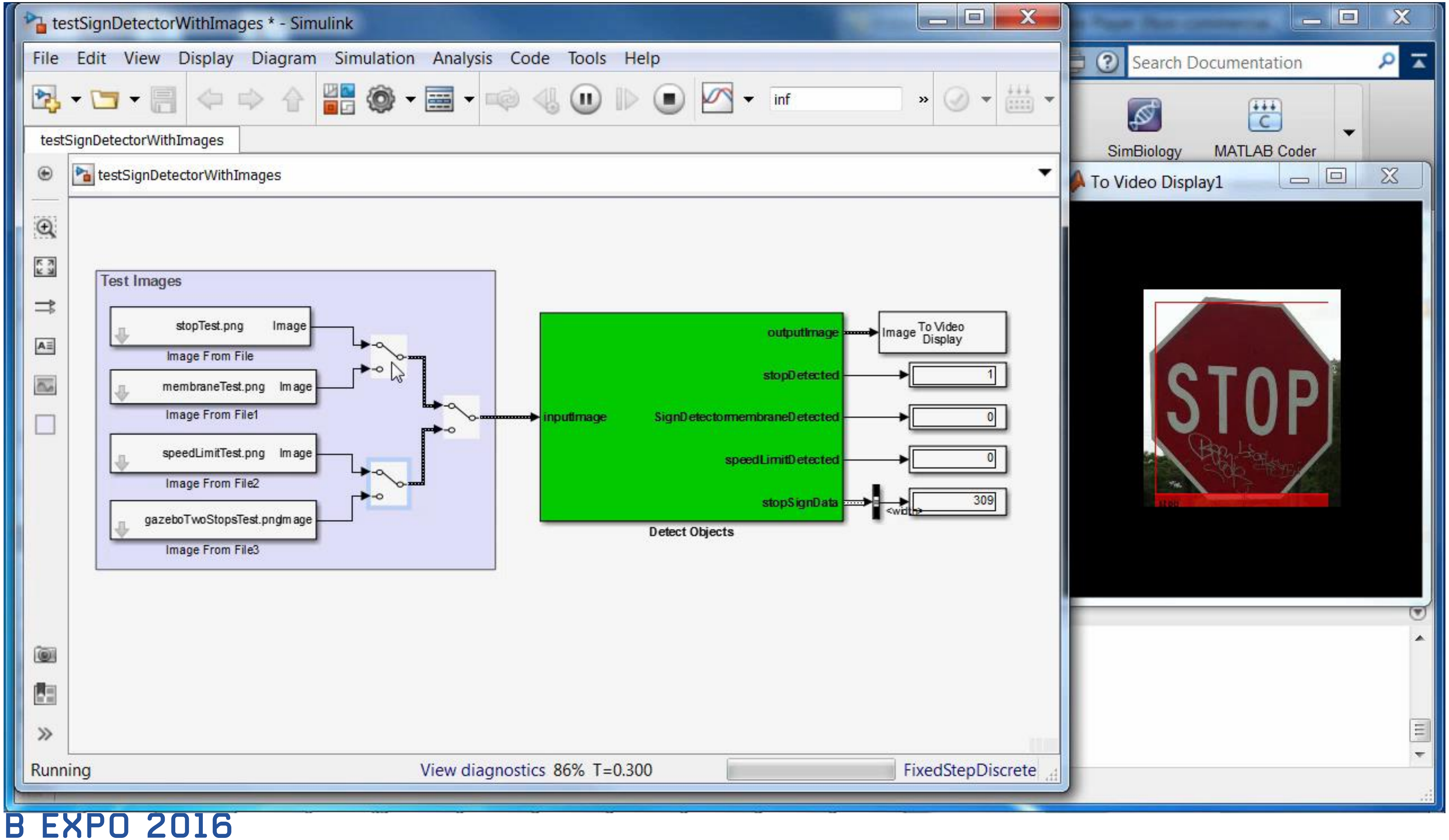
0

<width>

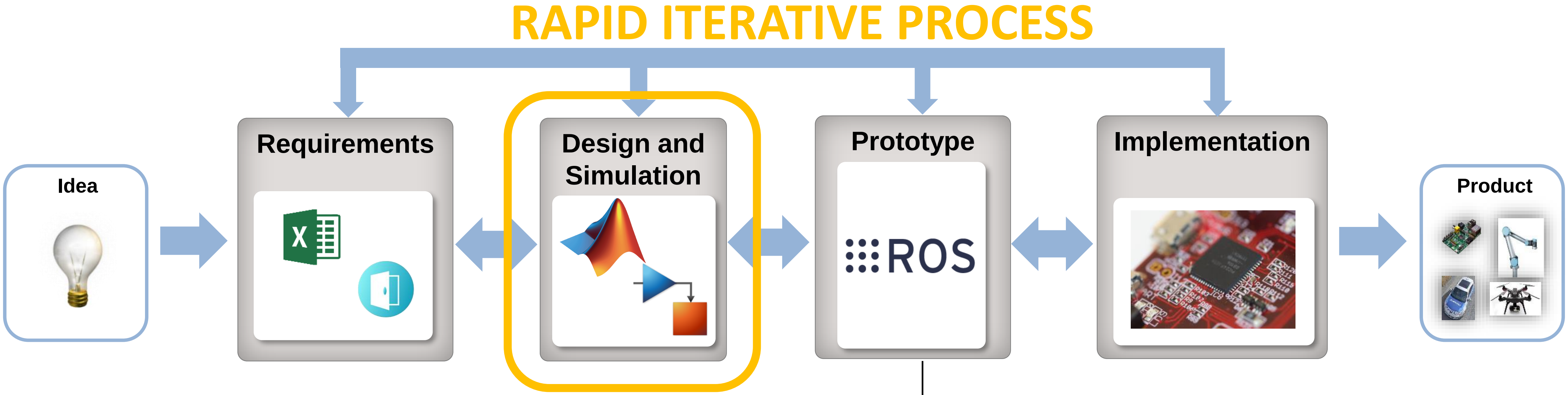
309



Design and Test Algorithm



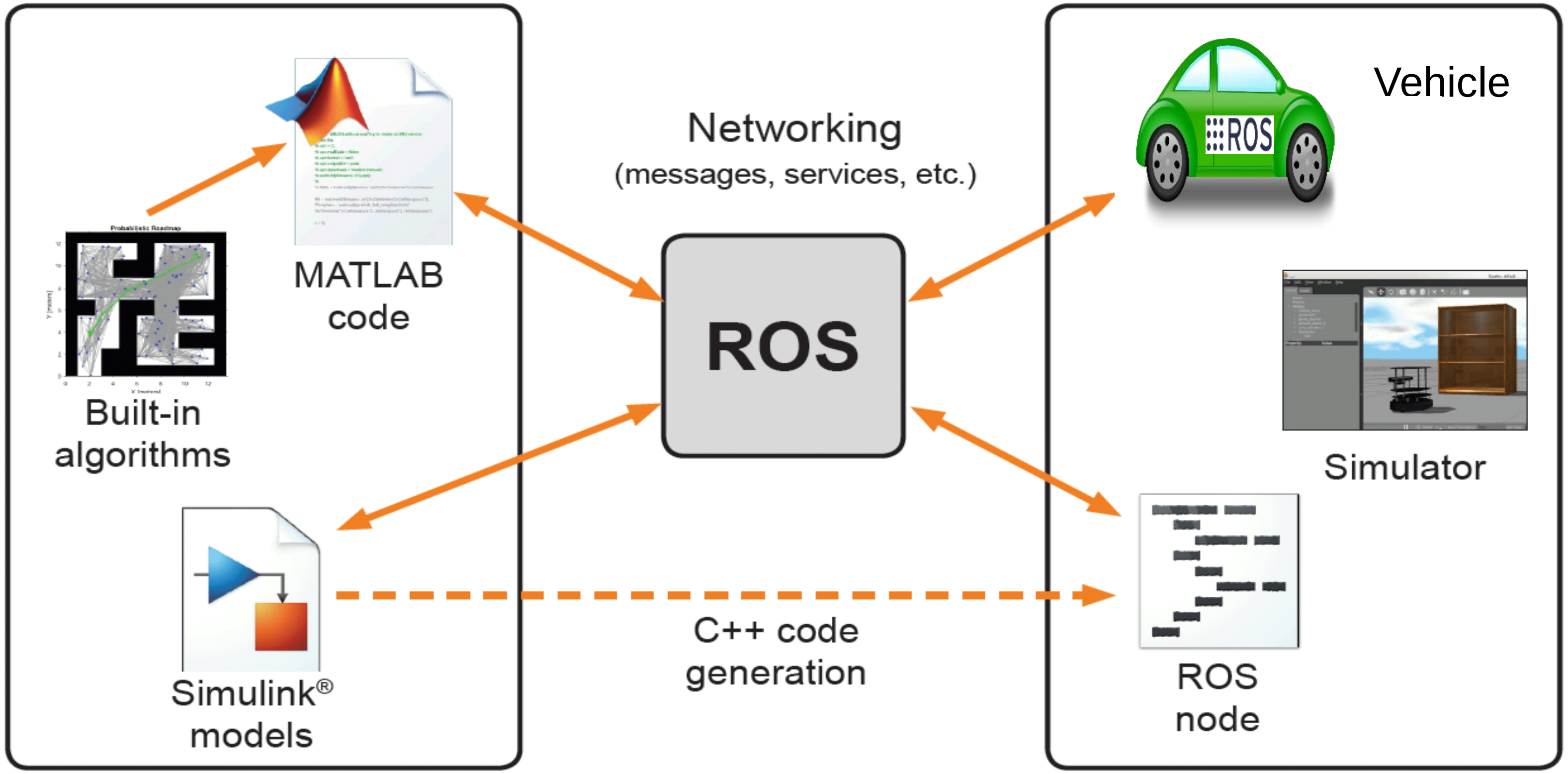
Robotics Development Workflow with MATLAB and ROS



- Test algorithm with an external simulator
- Tune your algorithm
- Integrate with other components

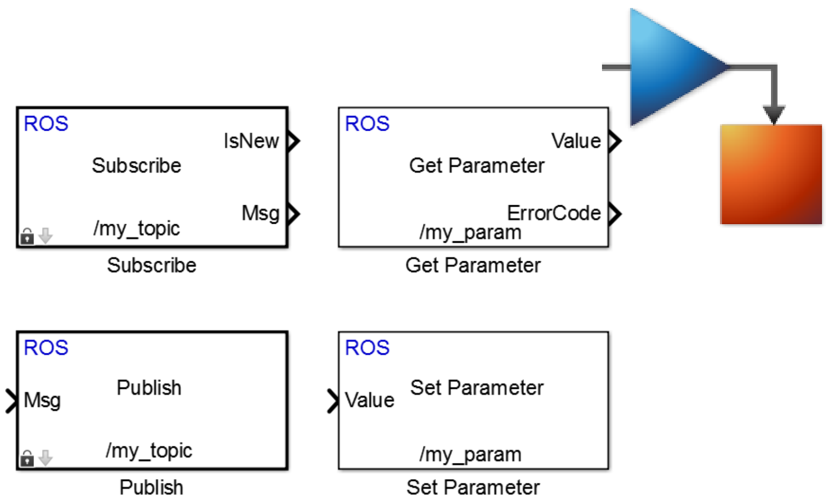
MATLAB and Simulink connect to the ROS network

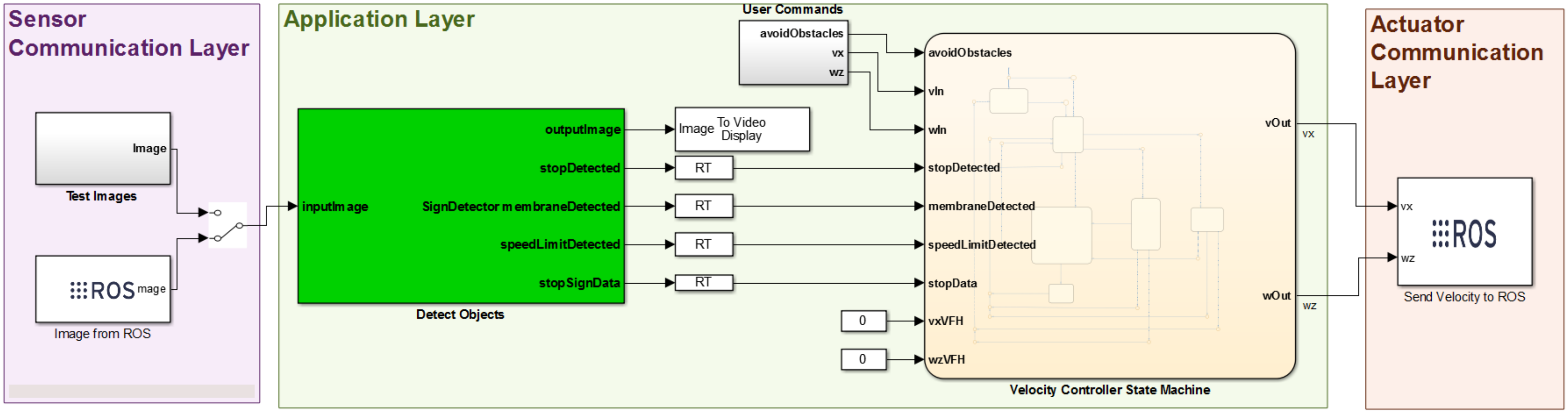
- Multiple master support
- ROS publishers/subscribers
- ROS services
- ROS TF tree
- ROS Parameter server



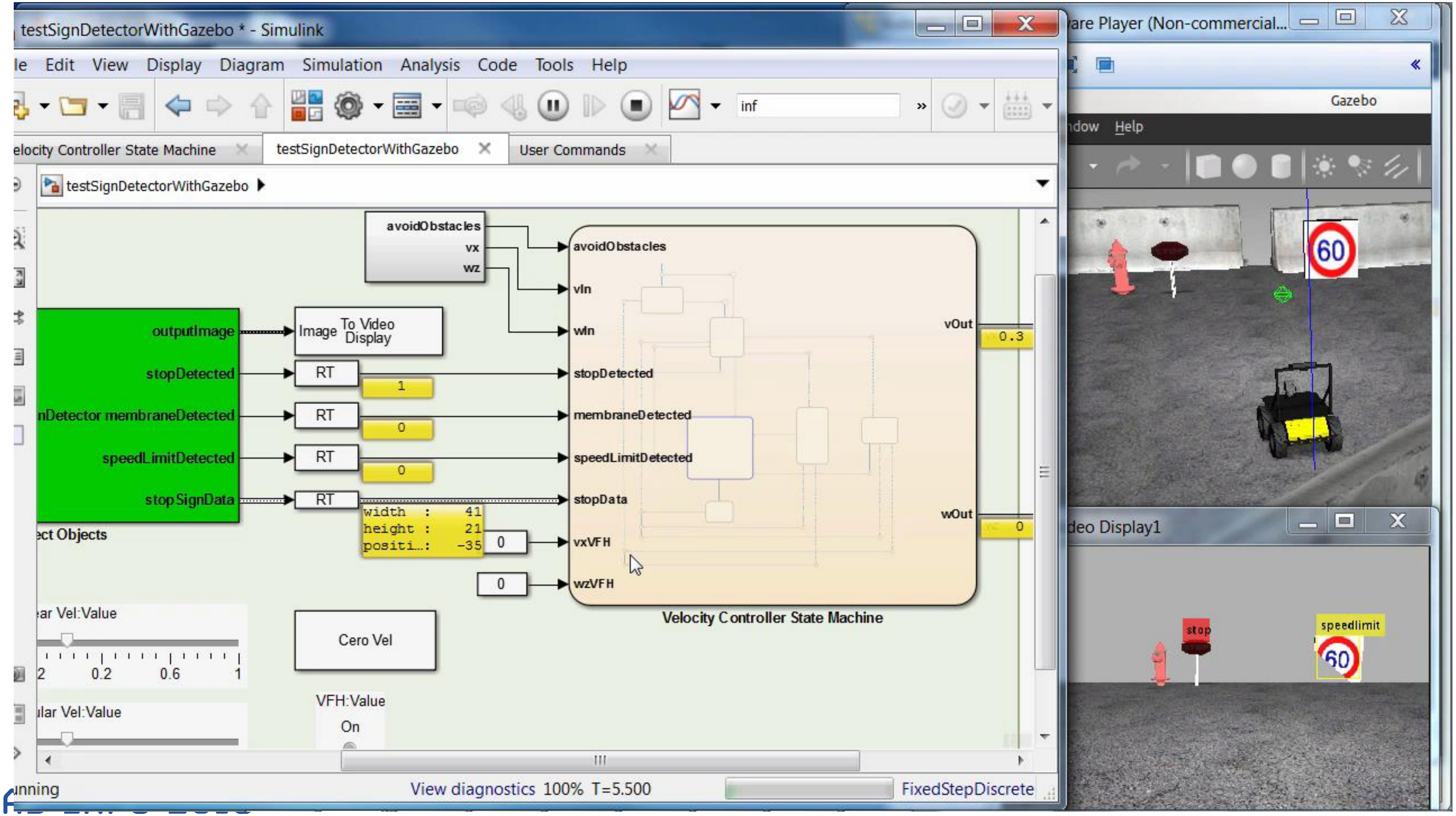
Co-simulation with ROS

```
%% Connect to ROS
rosinit '192.168.204.144';

%% Create subscribers
imSub = rossubscriber('/camera/rgb/image_raw');
scanSub = rossubscriber('/scan');

%% Create publisher
[velPub, velMsg] = rospublisher('/husky_velocity_controller/cmd_vel');
```

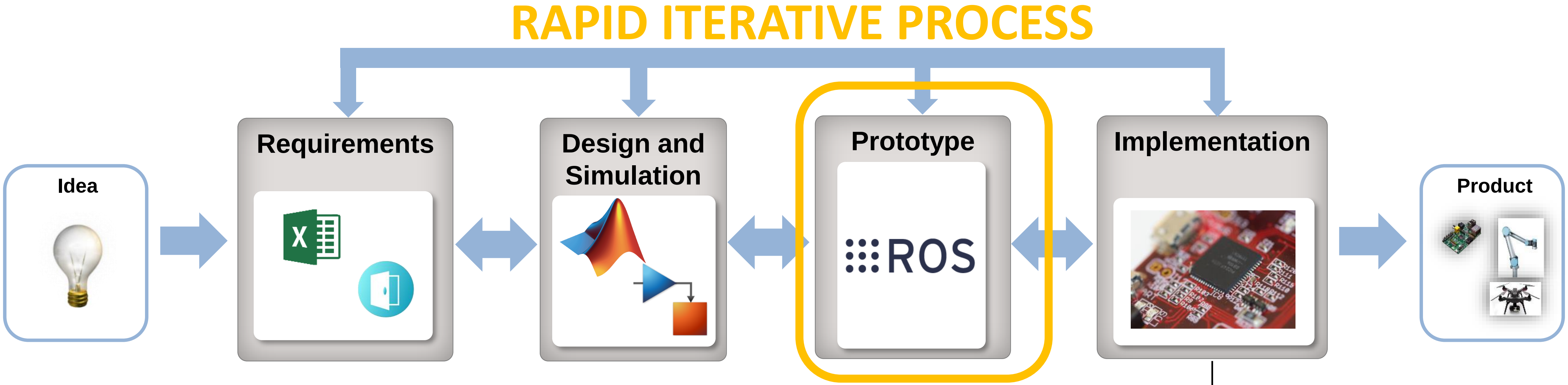




Co-simulation with ROS

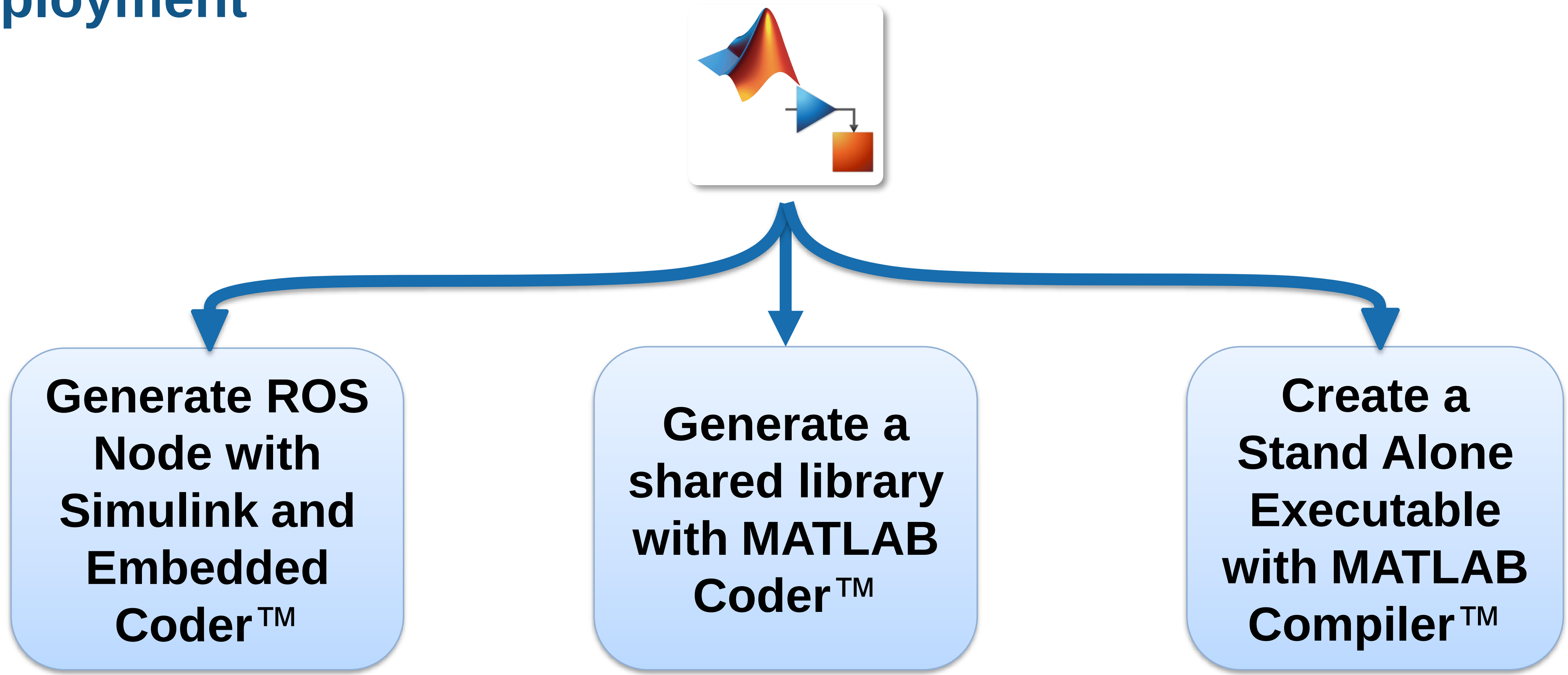


Robotics Development Workflow with MATLAB and ROS



System level design to target a different middleware or framework

Deployment



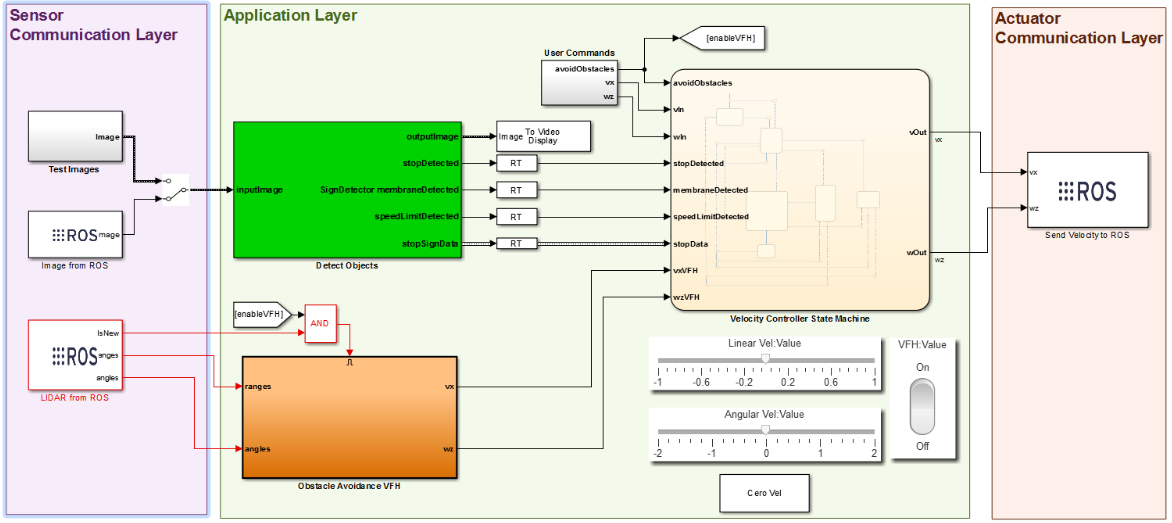
Generate ROS Node with Simulink and Embedded Coder™

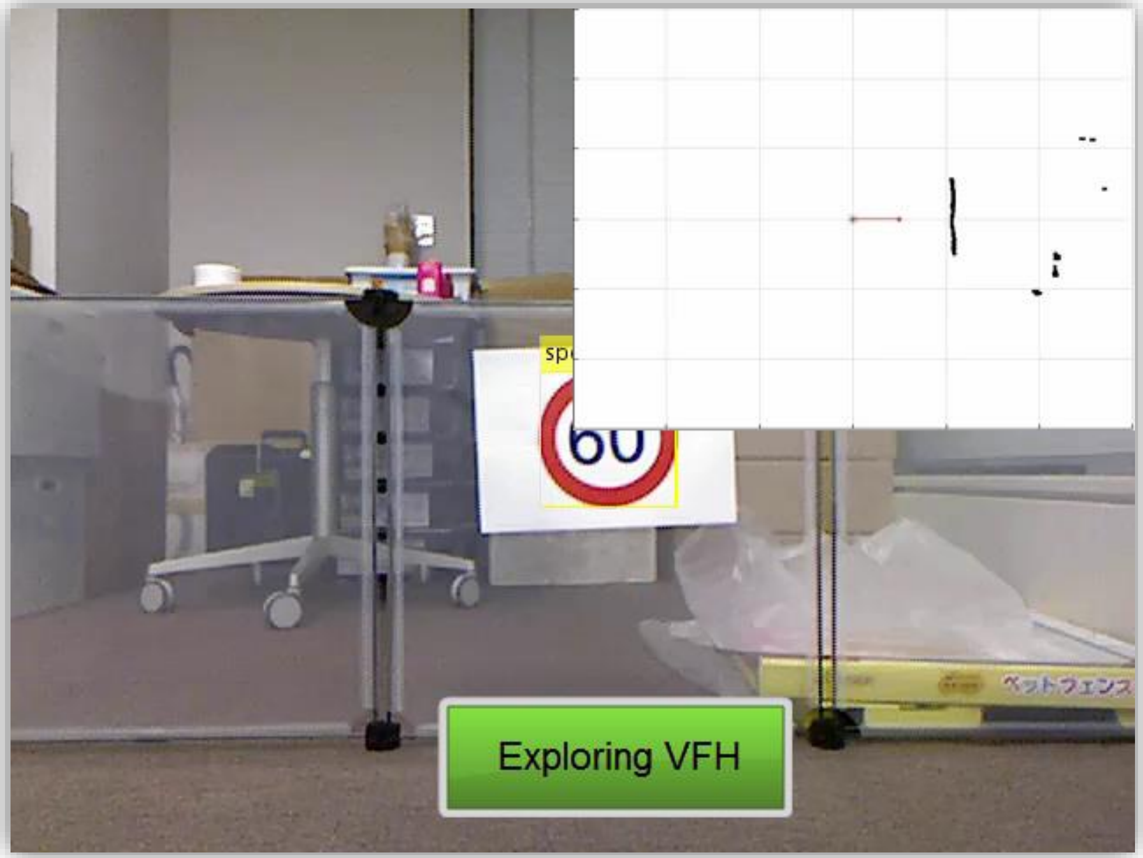
Generate a shared library with MATLAB Coder™

Create a Stand Alone Executable with MATLAB Compiler™

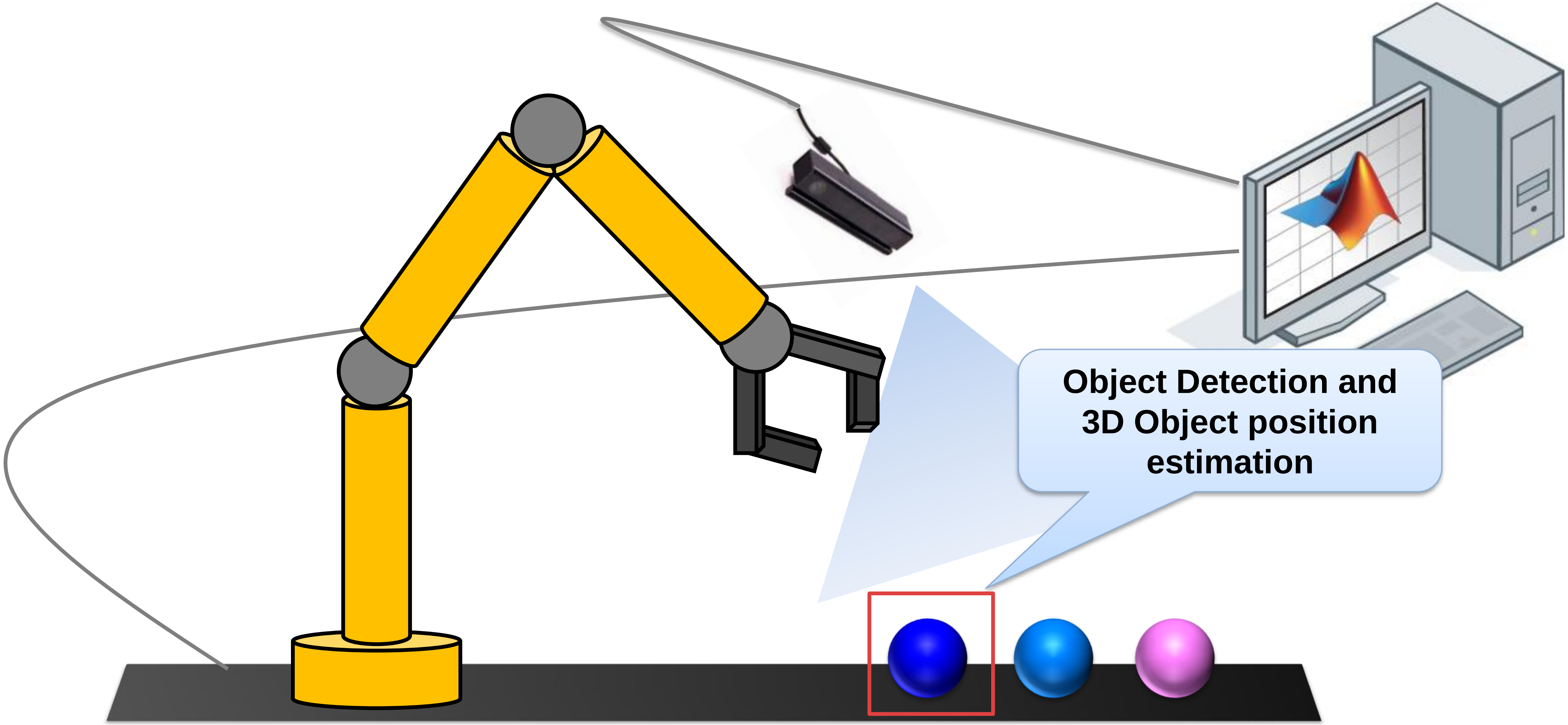
Determine deployment methods based on application

Implementation



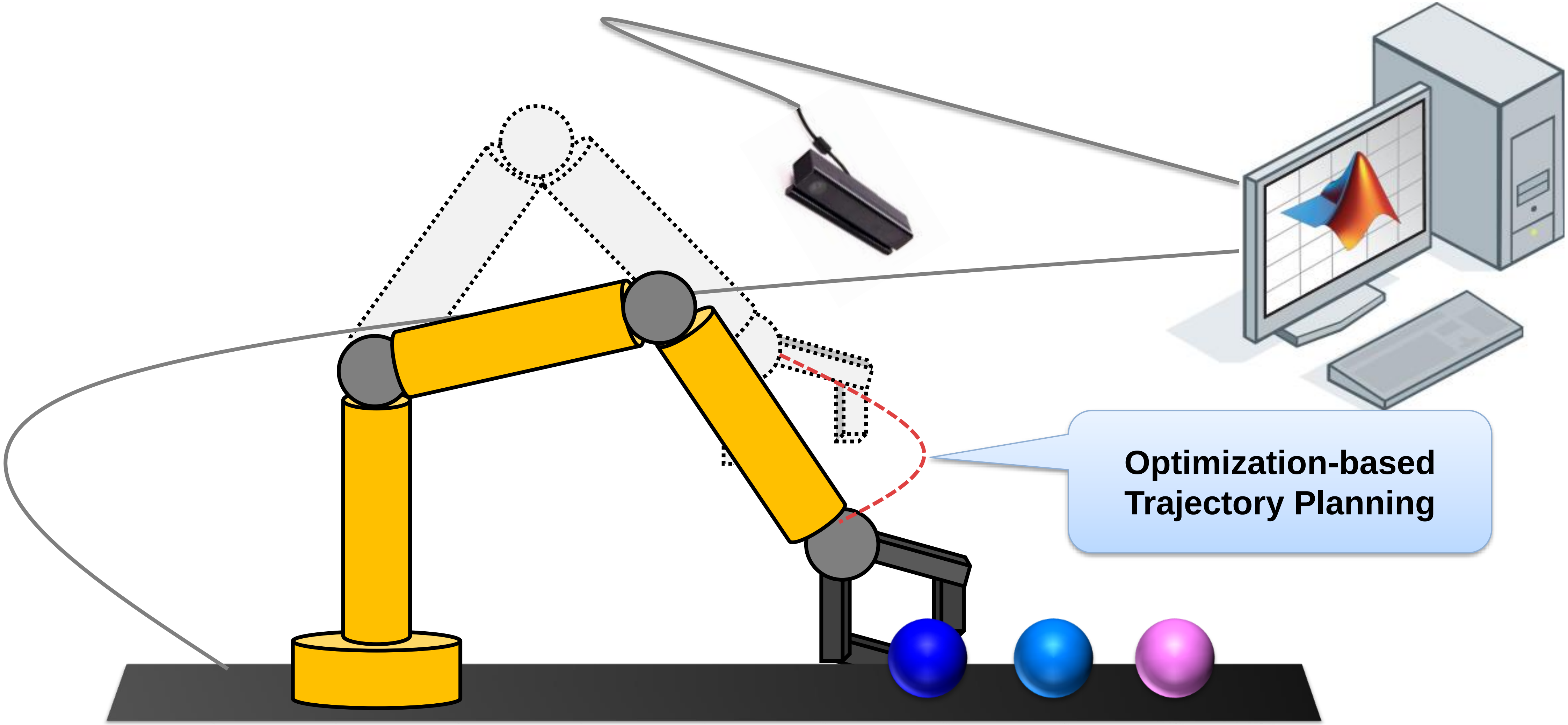


Part1: Object Detection and Position Estimation



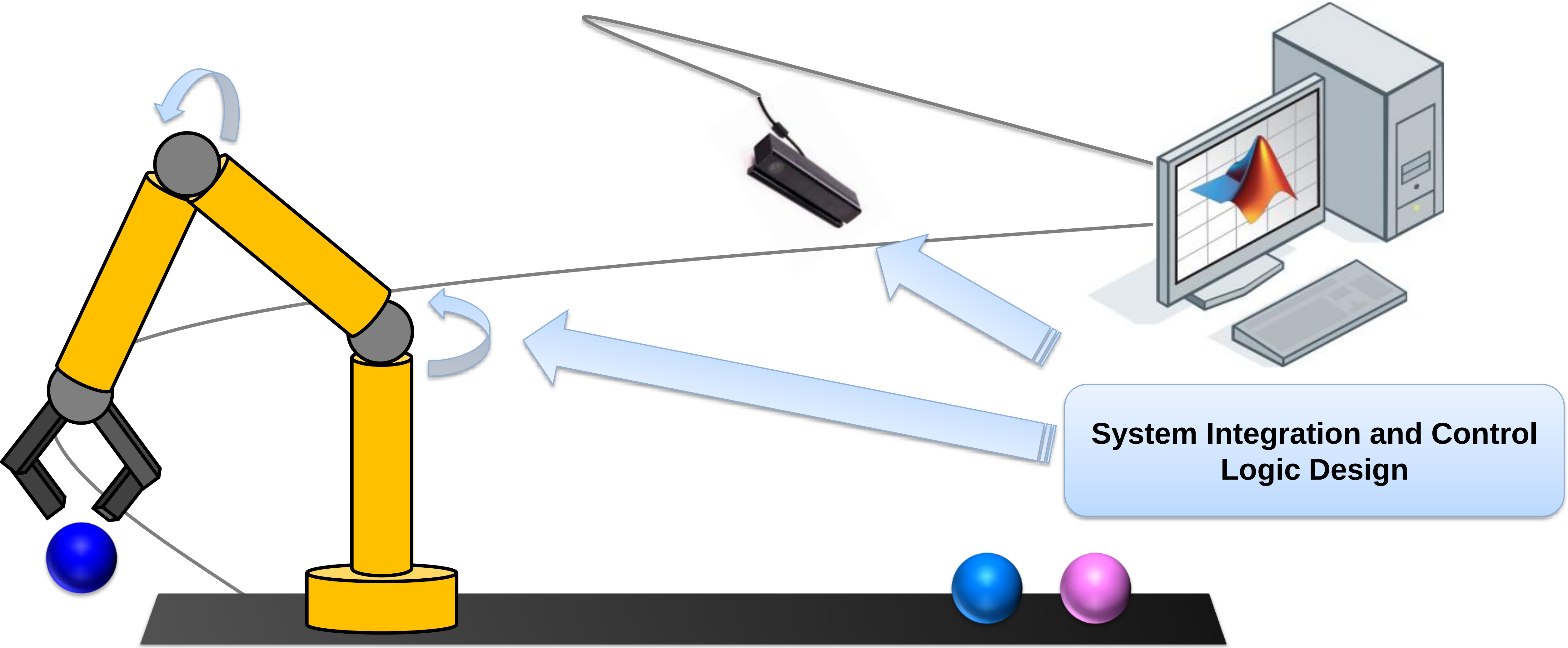
Object Detection and 3D Object position estimation

Part2: Trajectory Planning



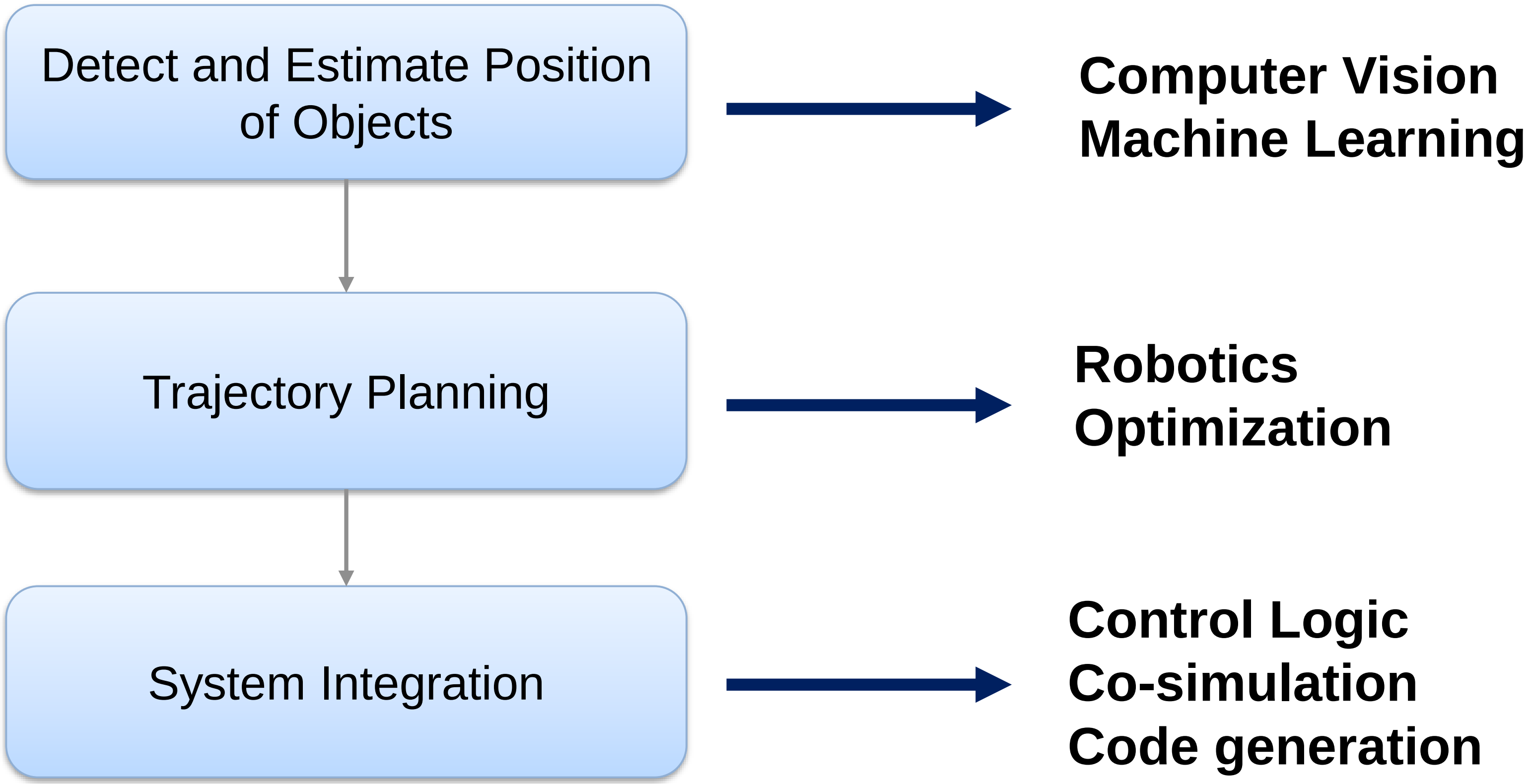
Optimization-based Trajectory Planning

Part3: System Integration



System Integration and Control Logic Design

Advanced Robotics Application Requires Multiple Technologies



Detect and Estimate Position of Objects

Trajectory Planning

System Integration

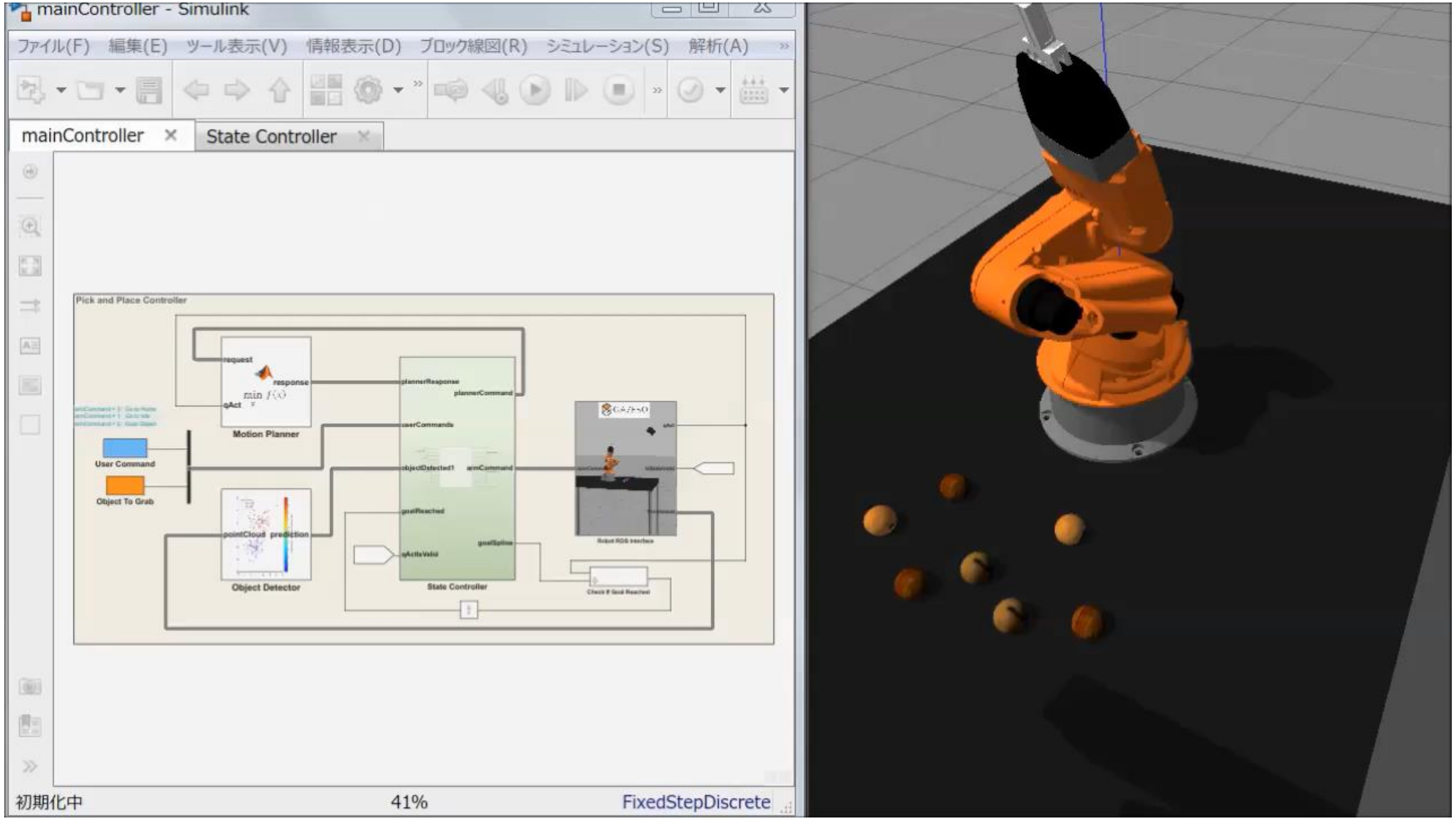
Computer Vision
Machine Learning

Robotics
Optimization

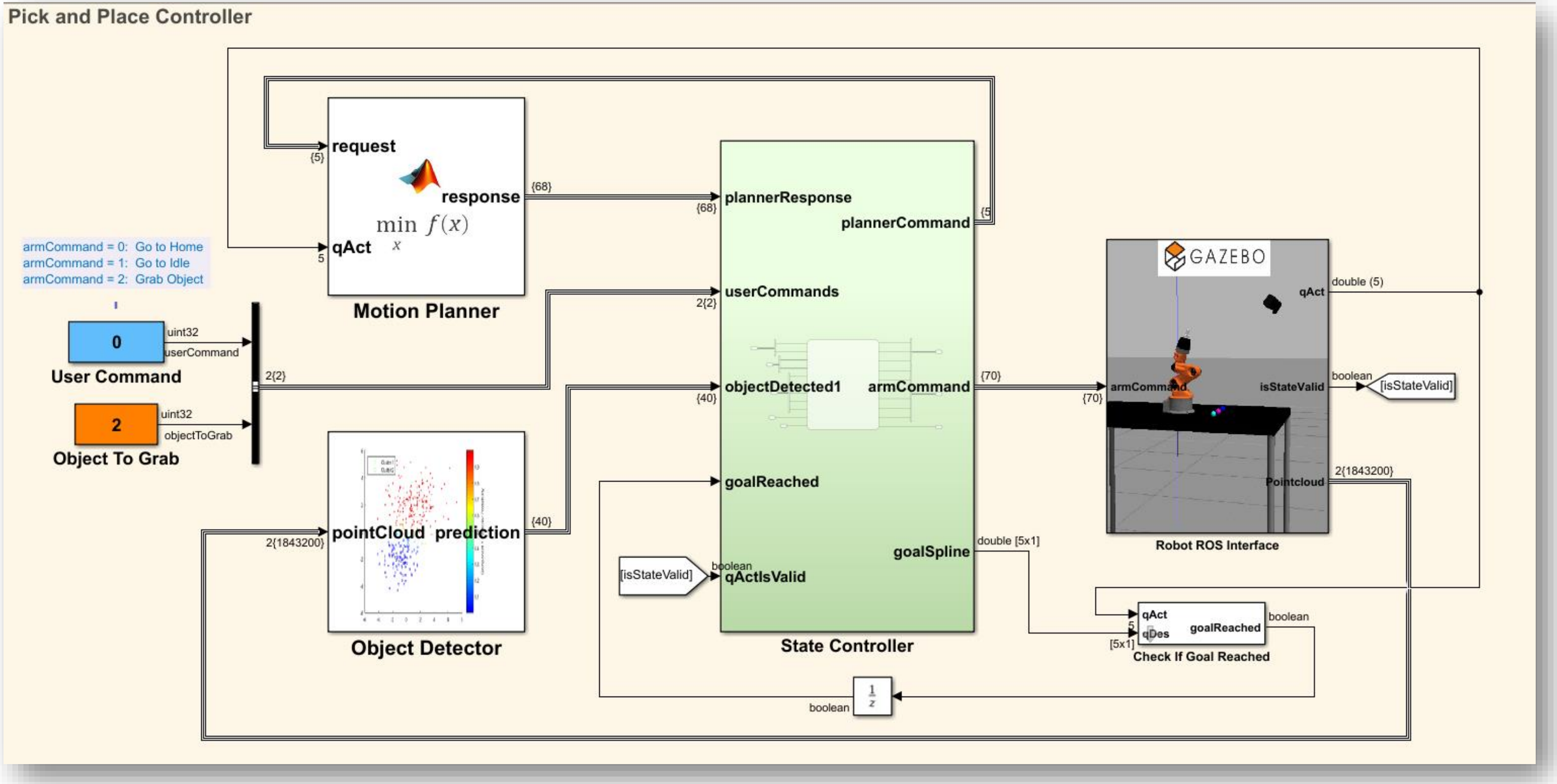
Control Logic
Co-simulation
Code generation

MATLAB and Simulink*: very powerful tools to design advanced robotics applications*

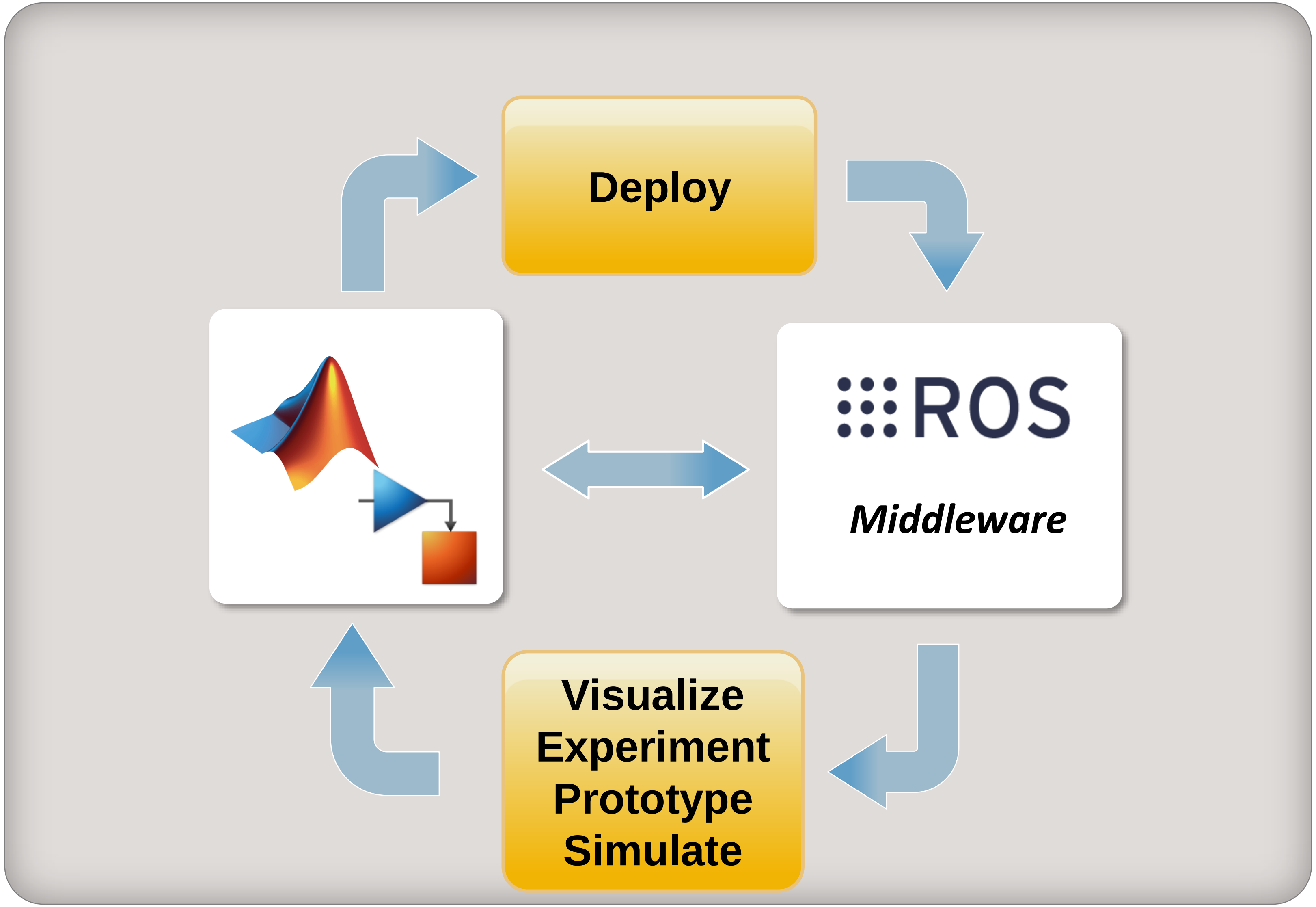
Trajectory Planning with RGB-D Sensor



System Level Design



Robotics Development Workflow with MATLAB and Simulink



Deploy

ROS

Middleware

Visualize
Experiment
Prototype
Simulate

Questions

```
% Thank you
```