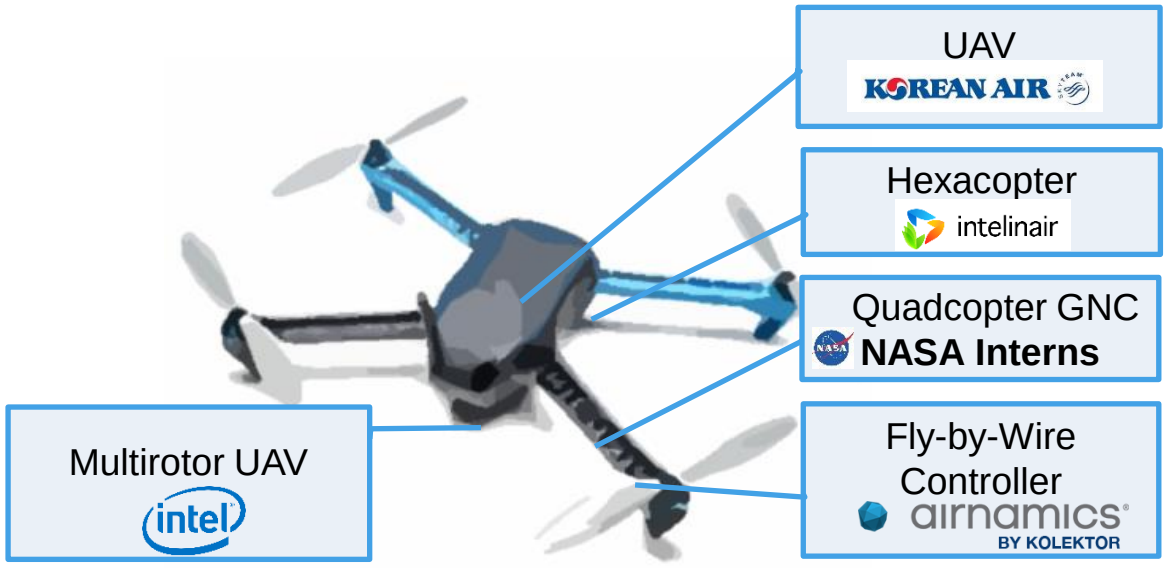
Robotics and Autonomous Systems

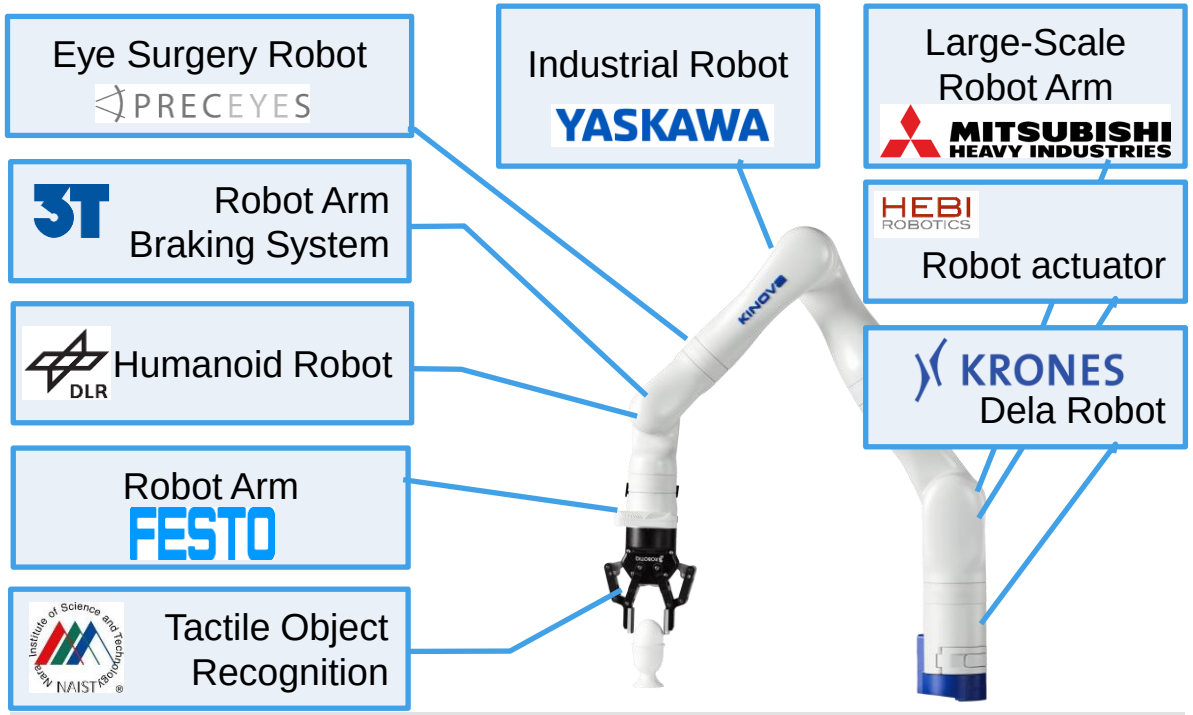
User Stories / User Cases

MathWorks

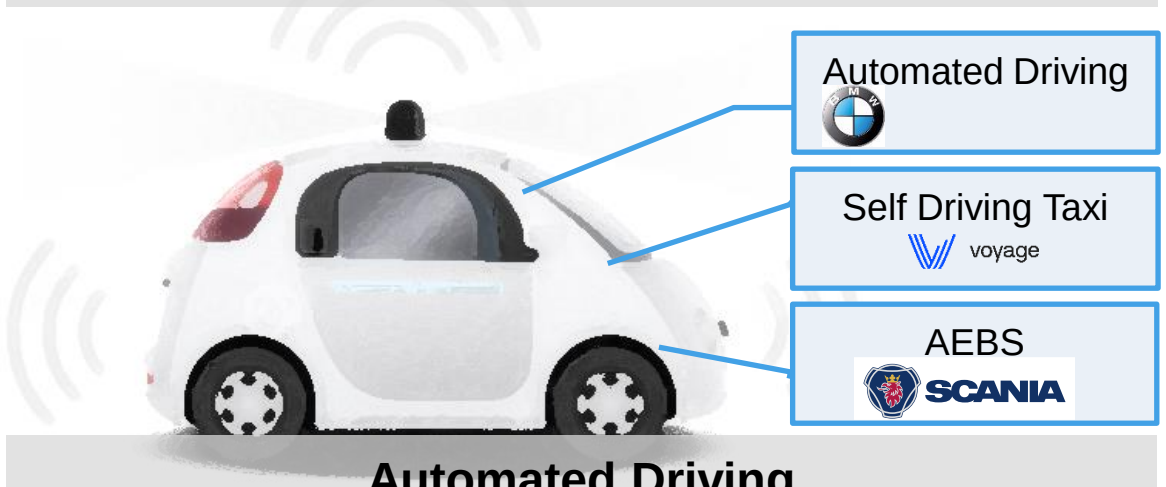
User Example: Robotics and Autonomous System Development



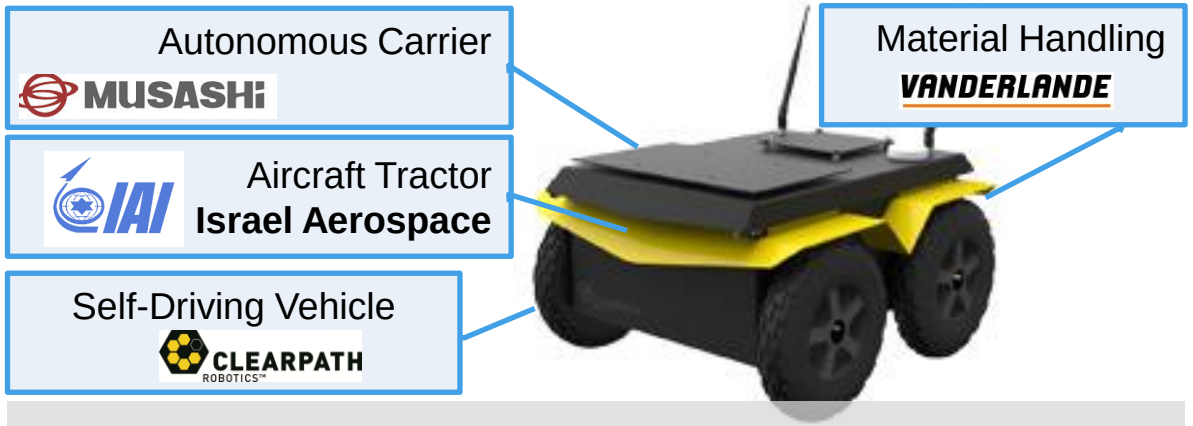
Unmanned Air Vehicle (UAV)



Manipulator – Robot Arm

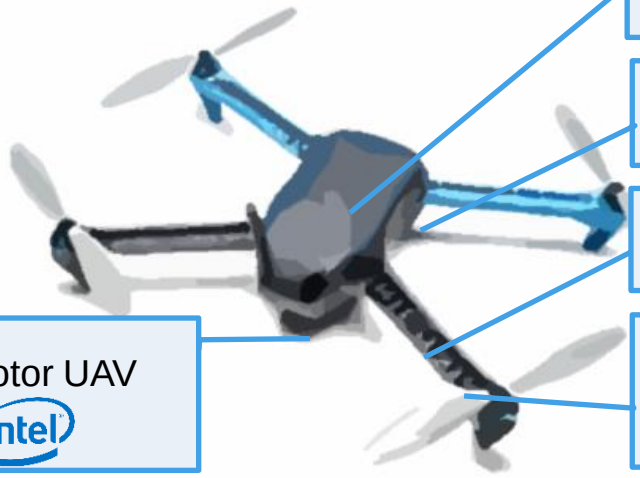


Automated Driving



Autonomous Ground Vehicle (AGV)

User Example: Robotics and Autonomous System Development

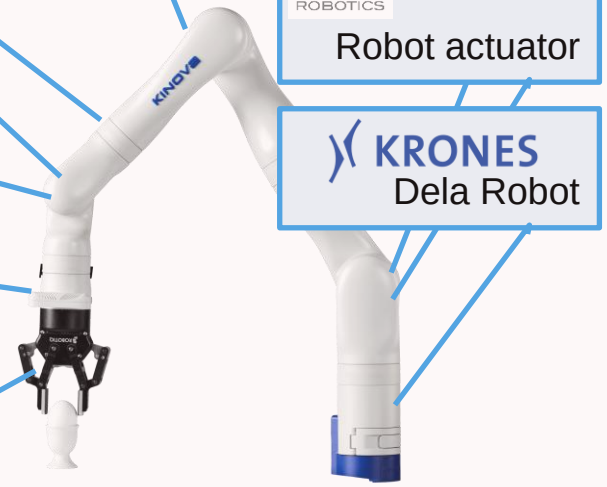


- UAV — KOREAN AIR
- Hexacopter — intelinair
- Quadcopter GNC — NASA Interns
- Multirotor UAV — intel
- Fly-by-Wire Controller — airnamics BY KOLEKTOR

Unmanned Air Vehicle (UAV)

- Automated Driving — BMW
- Self Driving Taxi — voyage
- AEBS — SCANIA

Automated Driving



- Eye Surgery Robot — PRECEYES
- Industrial Robot — YASKAWA
- Large-Scale Robot Arm — MITSUBISHI HEAVY INDUSTRIES
- Robot Arm Braking System — 3T
- Robot actuator — HEBI ROBOTICS
- Humanoid Robot — DLR
- KRONES Dela Robot
- Robot Arm — FESTO
- Tactile Object Recognition — National Institute of Science and Technology NAIST

Manipulator – Robot Arm

- Autonomous Carrier — MUSASHI
- Material Handling — VANDERLANDE
- Aircraft Tractor — IAI Israel Aerospace
- Self-Driving Vehicle — CLEARPATH ROBOTICS

Autonomous Ground Vehicle (AGV)

Autonomous System Examples

German Aerospace Center (DLR)

Autonomous Robots

Senses the environment using stereo cameras and tactile sensors on his skin

Performs human-like tasks

HEBI Robotics Enables Rapid Development of Real-Time Control Algorithms for Robots Assembled from Smart Actuators

Challenge

Enable researchers with diverse programming skills to rapidly develop real-time control software for robots assembled from smart actuator modules

Solution

Develop a MATLAB based API and example code that researchers use to jumpstart development and rapidly implement prototypes for testing control ideas

Results

- Cross-platform interoperability issues eliminated
- Key control system developed in weeks
- Customer ramp-up time shortened

Igor, a self-balancing two-wheeled robot built using the HEBI Robotics platform.

"Some of our demo systems would not be feasible without MATLAB as a starting point. The plotting and debugging tools enable rapid troubleshooting, and the portability of MATLAB code ensures that the system will work properly when our customers use a robotics kit with the MATLAB code and API we provide."

- Dave Rollinson, HEBI Robotics

Link to user story

3T Develops Robot Emergency Braking System with Model-Based Design

A SCARA robot.

Challenge

Design and implement a robot emergency braking system with minimal hardware testing

Solution

Model-Based Design with Simulink and HDL Coder to model, verify, and implement the controller

Results

- Cleanroom time reduced from weeks to days
- Late requirement changes rapidly implemented
- Complex bug resolved in one day

"With Simulink and HDL Coder we eliminated programming errors and automated delay balancing, pipelining, and other tedious and error-prone tasks. As a result, we were able to easily and quickly implement change requests from our customer and reduce time-to-market."

- Ronald van der Meer, 3T

Link to user story

Festo Develops Innovative Robotic Arm Using Model-Based Design

The Festo Bionic Handling Assistant. Image © Festo AG.

Challenge

Design and implement a control system for a pneumatic robotic arm

Solution

Use Simulink and Simulink PLC Coder to model, simulate, optimize, and implement the controller on a programmable logic controller

Results

- Complex PLC implementation automated
- Technology and innovation award won
- New business opportunities opened

“Using Simulink for Model-Based Design enables us to develop the sophisticated pneumatic controls required for the Bionic Handling Assistant and other mechatronic designs. With Simulink PLC Coder, it is now much easier to get from a design to a product.”

Dr. Rüdiger Neumann
Festo

Link to user story

Nara Institute of Science and Technology Researchers Develop Tactile Object Recognition Algorithms for a Robotic Hand

NAIST's ROS-enabled Shadow robot performing tactile object recognition.

Challenge

Enable a robotic hand to identify objects based on tactile sensor input

Solution

Use MATLAB to develop machine learning and object recognition algorithms and use Robotics System Toolbox to establish connectivity between the algorithms and the ROS-enabled robot

Results

- Hundreds of manual steps eliminated
- Opportunities to try new algorithm ideas increased
- Expertise shared with other students and researchers

"With Robotics System Toolbox, we seamlessly connected to and controlled our robot directly from the algorithms we had developed in MATLAB, enabling us to minimize development time. We used the time we gained to further our research into new tactile object recognition algorithms."

Takamitsu Matsubara

Nara Institute of Science and Technology

Link to user story

Mitsubishi Heavy Industries Develops Robotic Arm for Removing Nuclear Fuel Debris

Challenge

Design a multi-axis robot for removing molten fuel debris from the Fukushima Daiichi nuclear power station

Solution

Use MATLAB and Simulink to perform hardware measurement tests and to model and simulate individual robot axes and controllers

Results

- Development time halved
- Positioning accuracy requirement exceeded
- Shared platform for interorganizational collaboration established

Rendering of Mitsubishi Heavy Industries' seven-meter-long robotic arm capable of withstanding up to 2000 kg of processing reaction force

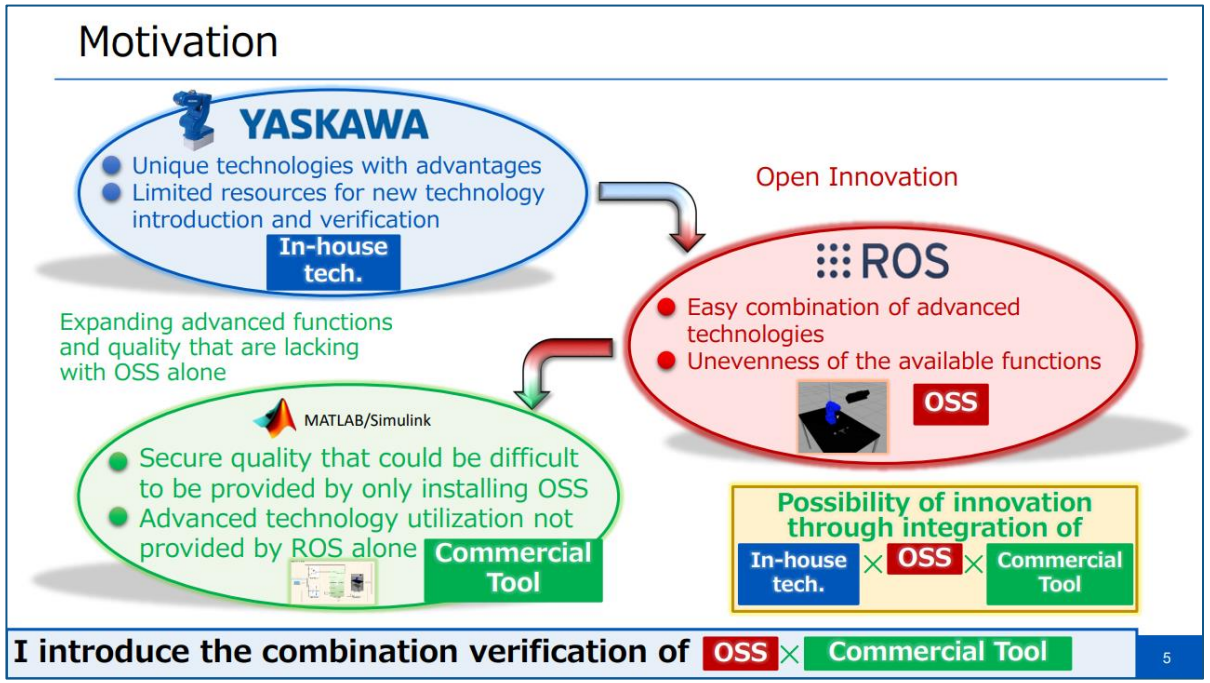
"Model-Based Design with MATLAB and Simulink supports a wide range of options, from classic to modern control, which made it possible to respond easily to any changes in design constraints and to meet the demanding accuracy requirement for this robot."

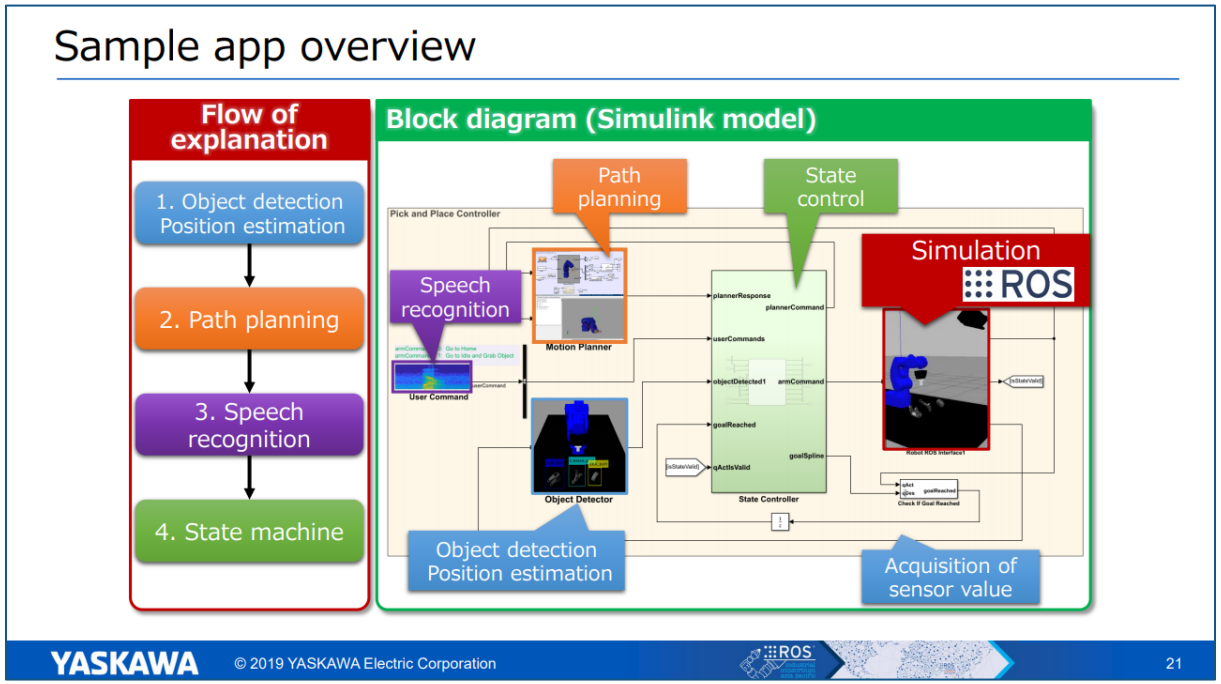
- Tadashi Murata, Mitsubishi Heavy Industries

Link to user story

User Presentation
Industrial Robots using ROS and MATLAB

Yasukawa Electric Co., Ltd. Ken Morita, Author of “ROS programming for practical robot development”





MATLAB EXPO 2019 in Japan

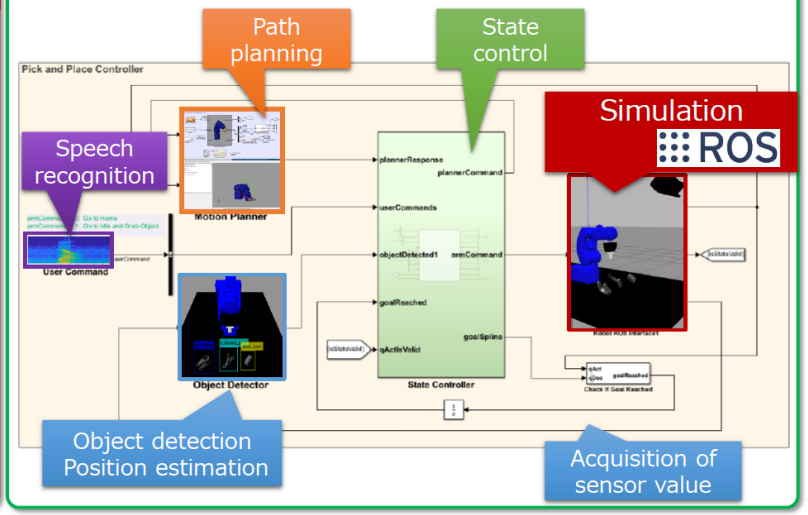
Integrated system-level testing with external simulator

Sample app overview

Flow of explanation

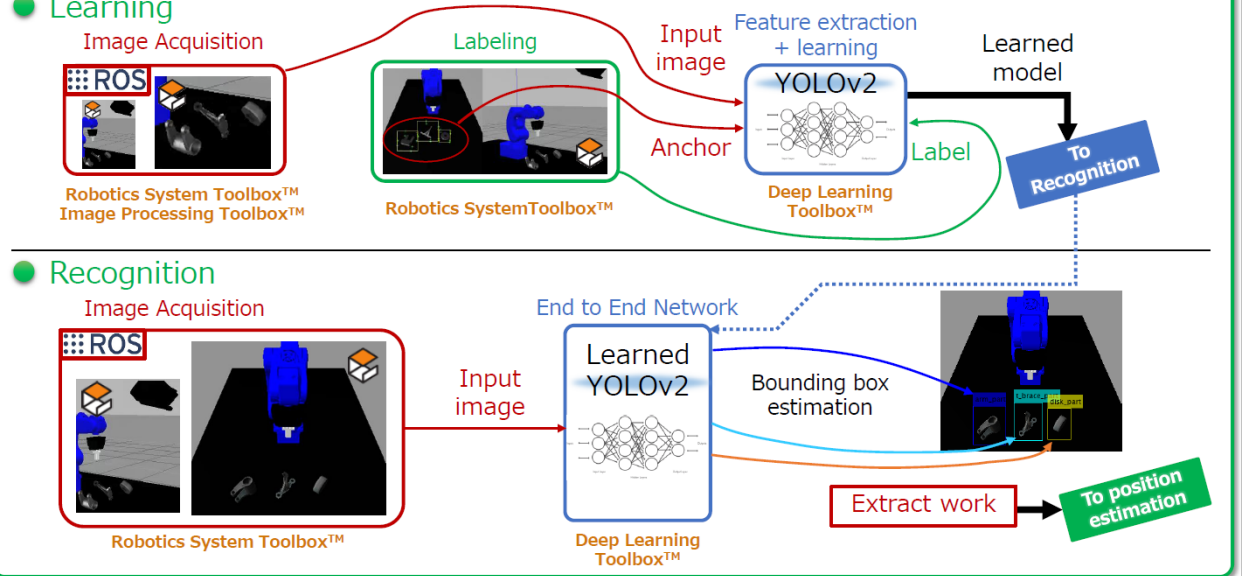
1. Object detection Position estimation
2. Path planning
3. Speech recognition
4. State machine

Block diagram (Simulink model)



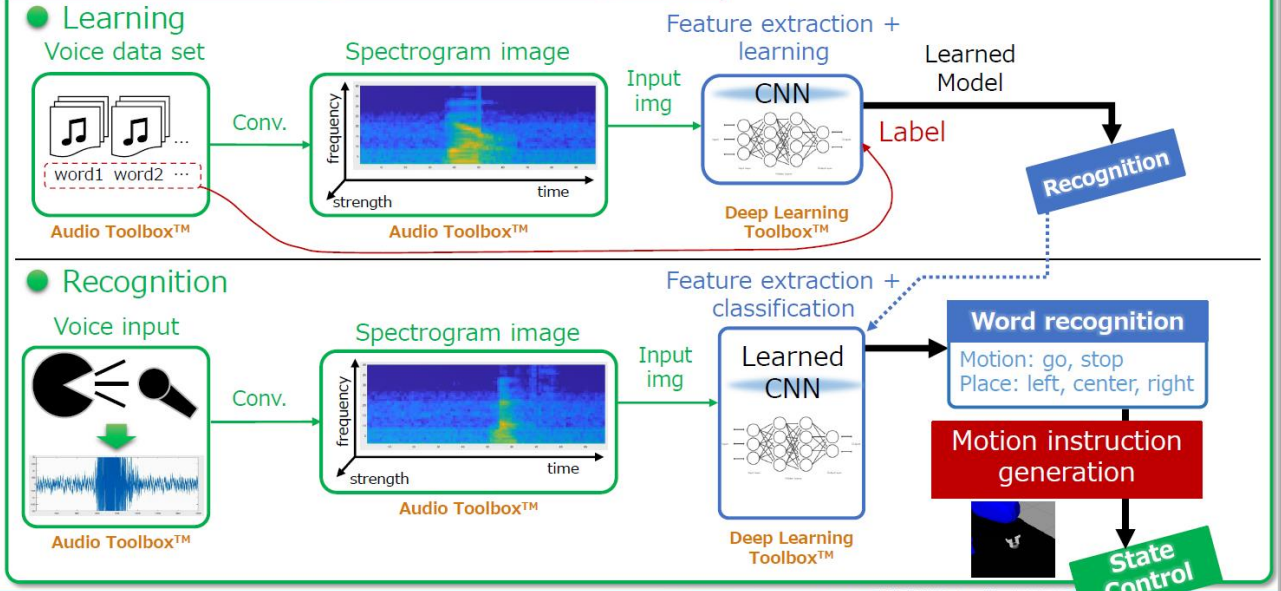
Flow of object recognition by YOLOv2

MATLAB Simulink



Flow of speech recognition w/t CNN

MATLAB Simulink



Italian Institute of Technology Develops Advanced Control Software for the iCub Humanoid Robot

Challenge

Design and implement control algorithms for a humanoid robot with 53 degrees of freedom

Solution

Use Model-Based Design to develop controllers, verify them via cosimulation, and test them on the robot

Results

- Telexistence and agent-robot collaboration control systems developed
- Independent robot operation implemented with generated code
- Control design reused on multiple humanoid robot platforms

Clockwise from top left: iCub shooting arrows, striking a yoga pose, walking while controlled via teleoperation, and working with a human to stand up.

"Our team has created a development workflow based on Simulink and Simulink Coder that makes it possible for even inexperienced team members to rapidly implement new control features, validate them through simulation, and run them on an iCub robot without writing any low-level code."

- Daniele Pucci, Italian Institute of Technology (IIT)

Link to technical article

Krones Develops Package-Handling Robot Digital Twin

Challenge

Increase the performance of an automated beverage-packaging system by incorporating a dynamic tripod robot into the design

Solution

Use Simulink and Simscape Multibody to create an accurate digital twin that supports design optimization, fault testing, and predictive maintenance

Results

- Robot performance increased
- Product development time shortened
- Testing time significantly reduced

The Krones Robobox T-GM package-handling robot.

"Simulations of the digital twin in Simulink enabled us to obtain data and insights that would be either impossible to get via hardware tests or simply too costly and time-consuming. Visualizing forces and moments helped us to understand the effects of individual components on a highly dynamic robot."
- Benedikt Böttcher, Krones

Link to user story

Musashi Seimitsu Industry Uses Deep Learning for Visual Inspection of Automotive Parts

Musashi Seimitsu Industry Co., Ltd.

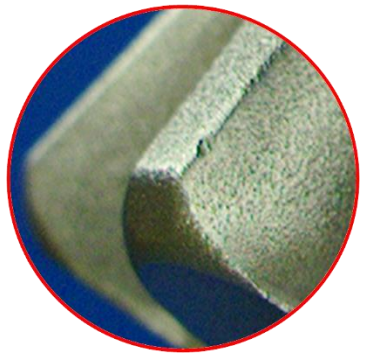
Musashi Seimitsu Industry Co., Ltd. prototyped an anomaly detection system using deep learning with MATLAB® for inspecting bevel gears used in automotive parts. After verifying accuracy and speed, they are now carrying out demonstration experiments at the manufacturing site. This approach is expected to reduce the workload and cost for manually operated visual inspection of 1.3 million parts per month.

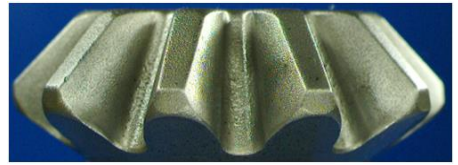
In this project, Musashi Seimitsu worked with MathWorks consultants to build a camera connection setup, preprocess images, create a custom annotation tool using App Designer, and improve the model accuracy. They generated code for the trained model using GPU Coder™, implemented it on NVIDIA® Jetson™, and leveraged the results to the PLC.

Advantages of using MATLAB and Simulink:

- Enable a seamless development workflow from image capture to implementation on embedded GPU
- Estimate and visualize the defect area using Class Activation Map
- Create custom user interfaces (App Designer) for improving labeling efficiency
- Leverage consulting services to maximize the benefits of using MATLAB

> “Using camera connection, preprocessing, and various pretrained models in MATLAB enabled us to work on the entire workflow. Through discussions with consultants, our team gained many tips for solving problems, growing the skills of our engineers.”





Explore MATLAB for deep learning

Deep Learning for Computer Vision

Watch video

Kyocera establishes development platform for collaborative robots using Model-Based Design

Challenges

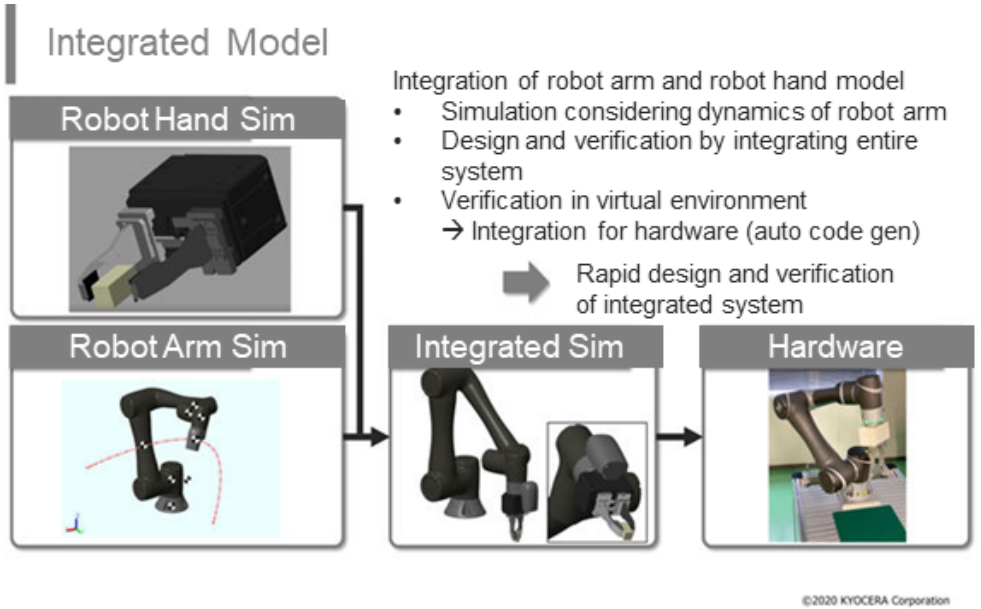
- Rapid development of robot hands for collaborative robots in high-mix optimal-volume production lines

Solution

- Develop inverse kinematics and trajectory planning for robot control system and simulate contact forces between robot hand and work pieces
- Connect to robot hardware by using MATLAB
- Verify grasping behavior through integrated simulation

Results

- Time for prototyping from design to verification reduced
- Development platform established which supports seamless transition from virtual to real
- Integration with ROS-enables robots simplified



Integrated simulation and hardware implementation of robot hand and robot arm

MATLAB and Simulink provide useful capabilities and a variety of interfaces. By creating a virtual development platform and linking it with ROS, we were able to rapidly develop and deploy algorithms to hardware.

Details (Japanese)

ASTRI Accelerates Development of Robotic Manipulation System Using MBSE Digital Twin

Challenge

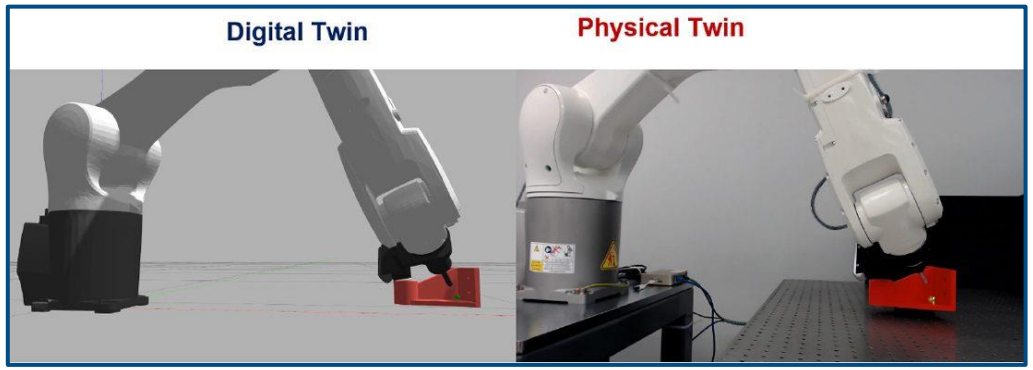
Reduce development time, manual processes, and costs

Solution

Adopt model-based systems engineering and develop a digital twin with MATLAB, Simulink, and Deep Learning Toolbox

Results

- Integration time reduced by 40%
- Issues resolved in the design stage
- Teams worked collaboratively



ASTRI created a digital twin to design, build, and validate its robotic welding system.

"The integration of MATLAB, Simulink, and Deep Learning Toolbox gave us the confidence to move forward with the MBSE digital twin project."

- Dr, T. John Koo, ASTRI

Link to user story

Hitachi prototypes flexible production system using Node-RED and MATLAB

Challenges

Need a platform to build complex production systems from edge device development to process management, including AI and robot control

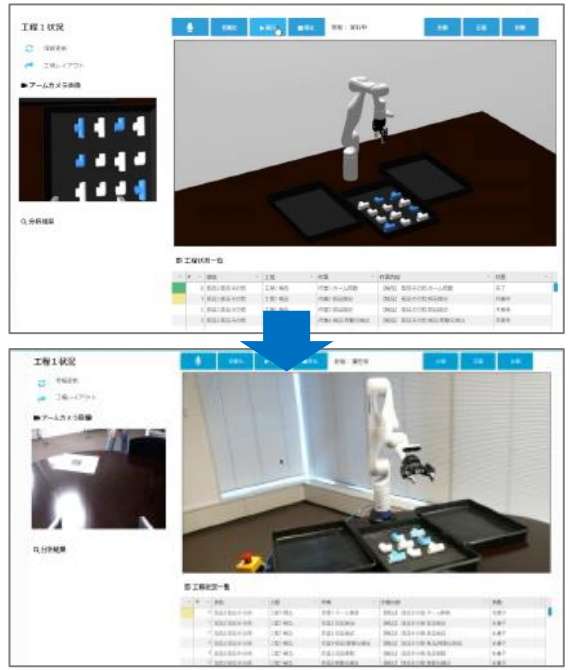
Solutions

Integrate MATLAB/Simulink and Node-RED, which is a low-code programming tool, to build entire system from edge AI to process control system

Use Gazebo simulator through ROS to leverage a digital twin of the hardware and improve development efficiency

Results

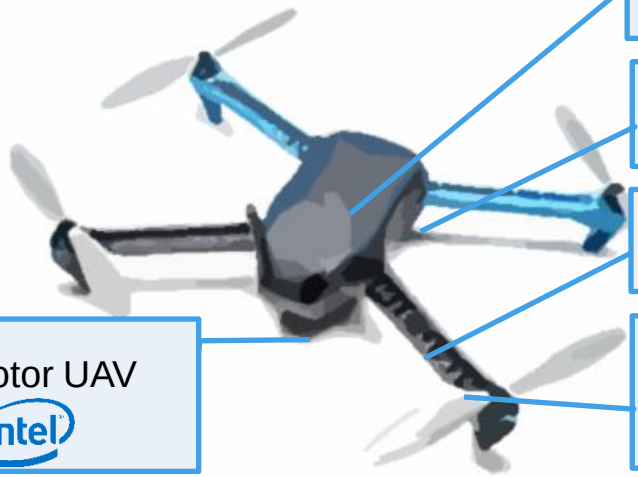
- Built a prototype of a complex production system that can address changes of production plan and process rapidly



By integrating Node-RED with MATLAB and Simulink, it allows us to develop IoT and robotics systems efficiently.

Yaoko Nakagawa and Hiroyasu Nishiyama
R&D Group, Hitachi

User Example: Robotics and Autonomous System Development



Multirotor UAV
intel

UAV
KOREAN AIR

Hexacopter
intelinair

Quadcopter GNC
NASA Interns

Fly-by-Wire Controller
airnamics BY KOLEKTOR

Unmanned Air Vehicle (UAV)

Eye Surgery Robot
PRECEYES

Robot Arm Braking System
3T

Humanoid Robot
DLR

Robot Arm
FESTO

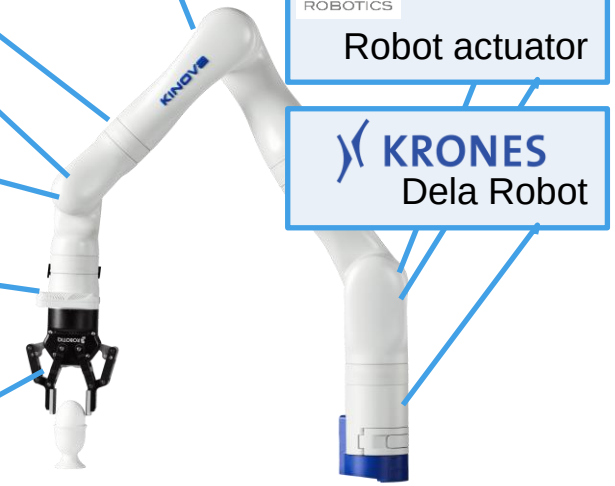
Tactile Object Recognition
Nara Institute of Science and Technology NAIST

Industrial Robot
YASKAWA

Large-Scale Robot Arm
MITSUBISHI HEAVY INDUSTRIES

Robot actuator
HEBI ROBOTICS

KRONES Dela Robot



Manipulator – Robot Arm

Automated Driving

Self Driving Taxi
voyage

AEBS
SCANIA

Automated Driving

Autonomous Carrier
MUSASHI

Aircraft Tractor
IAI Israel Aerospace

Self-Driving Vehicle
CLEARPATH ROBOTICS

Material Handling
VANDERLANDE

Autonomous Ground Vehicle (AGV)

Musashi Seimitsu Industry Co., Ltd. Develops Prototype of Industrial Autonomous Carrier Robot in Six Months

Haruki Takemoto and Kenneth Renny Simba, Musashi Seimitsu Industry Co., Ltd.

Musashi Seimitsu Industry Co. Ltd. developed a prototype system of an industrial autonomous carrier robot in six months. Developing an independent four-wheel steering system, especially tuning the motion controller iteratively, is very challenging. To minimize iterative development, Musashi Seimitsu Industry adopted Model-Based Design.

Musashi Seimitsu engineers used Simulink® to model the complex vehicle dynamics and used simulation to optimize the motion controller parameters.

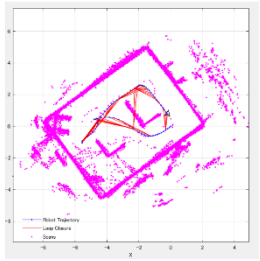
In addition to sensors, the autonomous driving vehicle will need to have maps of the environments and the ability to estimate an egocentric pose of the vehicle within the map. Using Robotics System Toolbox™, the engineers built two-dimensional maps offline and tested the localization algorithm with the maps in simulation.

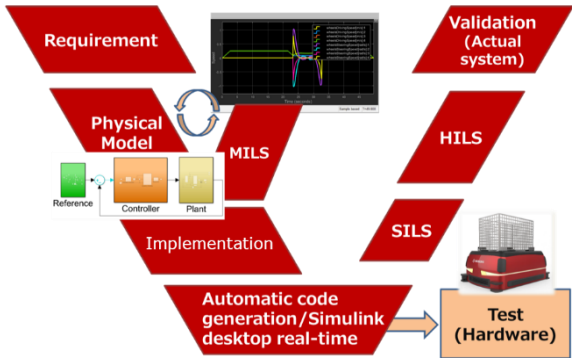
The next challenge is localization with lidar and other sensors. Musashi Seimitsu engineers will work on sensor fusion to integrate the IMU and odometry.

Advantages of using MATLAB® and Simulink:

- Shortened development time for tuning controller algorithms by performing simulation
- Tested lidar-based localization algorithm in desktop simulation

> “Using MATLAB and Simulink, **we designed a prototype for the motion controller and tested it on the hardware within a month**. The localization algorithm was evaluated and **challenges were clarified by performing simulations**.”





» Learn how to implement SLAM with lidar scans

Learn More

» Robotics and autonomous systems

Clearpath Robotics Accelerates Algorithm Development for Industrial Robots

Challenge

Shorten development times for laser-based perception, computer vision, fleet management, and control algorithms used in industrial robots

Solution

Use MATLAB to analyze and visualize ROS data, prototype algorithms, and apply the latest advances in robotics research

Results

- Data analysis time cut by up to 50%
- Customer communication improved
- Cutting-edge SDV algorithms quickly incorporated

An OTTO self-driving vehicle from Clearpath Robotics.

"ROS is good for robotics research and development, but not for data analysis. MATLAB, on the other hand, is not only a data analysis tool, it's a data visualization and hardware interface tool as well, so it's an excellent complement to ROS in many ways."

- Ilia Baranov, Clearpath Robotics

Link to user story

Israel Aerospace Industries Develops DO-178B Level B Certified Software for a Hybrid-Electric Aircraft Tractor

Challenge

Develop control software for the world's first certified aircraft-taxiing vehicle

Solution

Use Model-Based Design to model the control loops, application logic, and plant; run simulations and HIL tests; and generate DO-178B certified production code

Results

- Development time halved
- 50% of models reused
- DO-178B certification simplified

The TaxiBot from Israel Aerospace Industries

"We initially intended to develop only control loops with Model-Based Design, but the process proved so efficient that we decided to use Model-Based Design for the application layer as well. Being able to run a model, see that it's working right, and then generate certifiable code is a big advantage."

- Zeev Gabbin, Israel Aerospace Industries

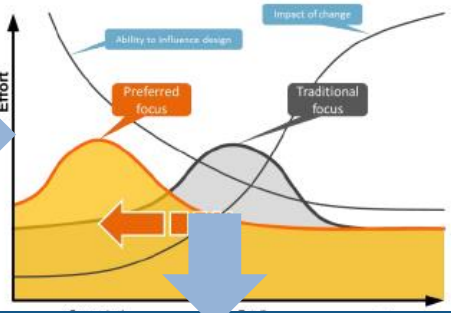
Link to user story

Model-Based Design in Vanderlande

Complexity

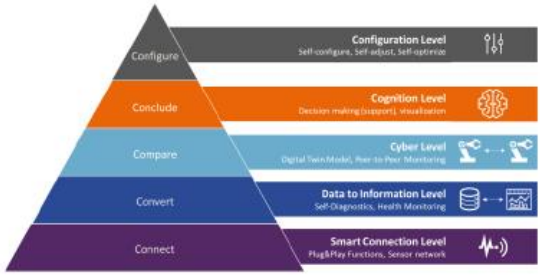
Complex Cyberphysical Systems

Shift of Focus

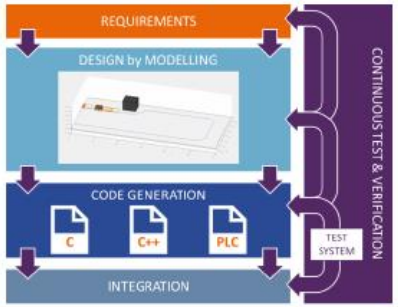


Model-Based Design

Architecture



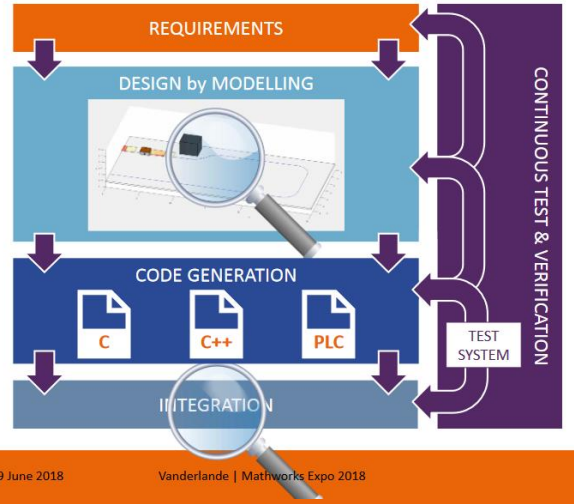
Design Automation



32 19 June 2018 Vanderlande | Mathworks Expo 2018

VANDERLANDE

Model Based Design: Focus on design & integration

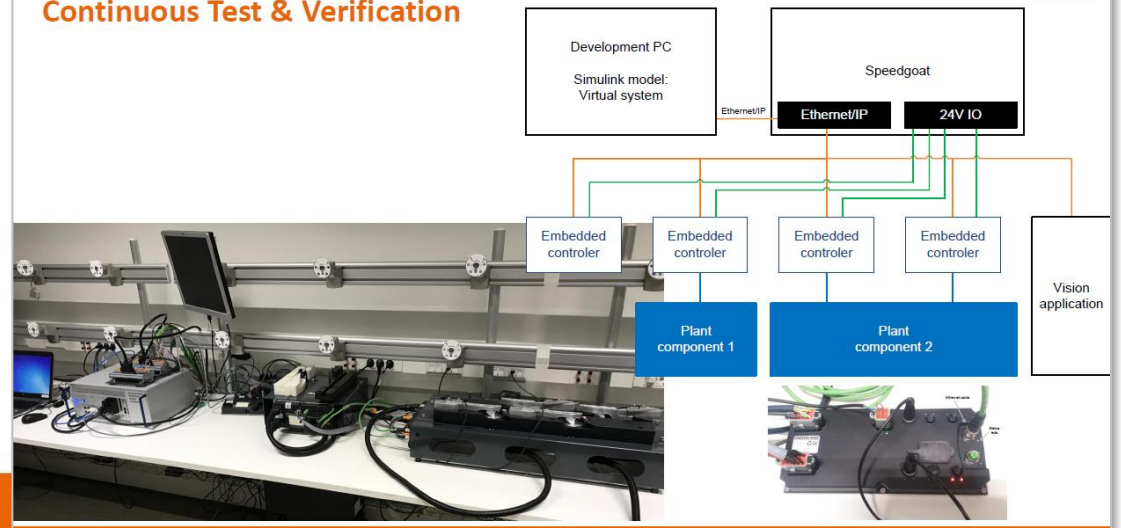


MATLAB® & SIMULINK®
MathWorks®

26 19 June 2018 Vanderlande | Mathworks Expo 2018 MOVING YOUR BUSINESS FORWARD

VANDERLANDE

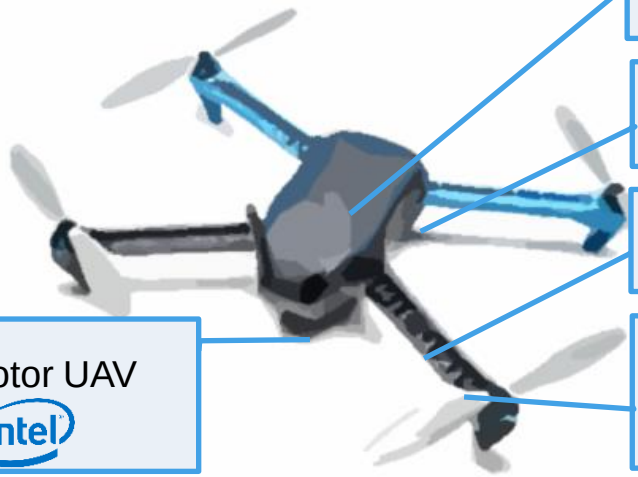
Continuous Test & Verification



29 19 June 2018 Vanderlande | Mathworks Expo 2018 MOVING YOUR BUSINESS FORWARD

https://www.matlabexpo.com/content/dam/mathworks/mathworks-dot-com/images/events/matlabexpo/nl/2018/how-to-ride-the-wave-of-innovative-mechatronics-embracing-complexity.pdf

User Example: Robotics and Autonomous System Development



UAV
KOREAN AIR

Hexacopter
intelinair

Quadcopter GNC
NASA Interns

Multirotor UAV
intel

Fly-by-Wire Controller
airnamics BY KOLEKTOR

Unmanned Air Vehicle (UAV)

Eye Surgery Robot
PRECEYES

Industrial Robot
YASKAWA

Large-Scale Robot Arm
MITSUBISHI HEAVY INDUSTRIES

3T Robot Arm Braking System

HEBI ROBOTICS Robot actuator

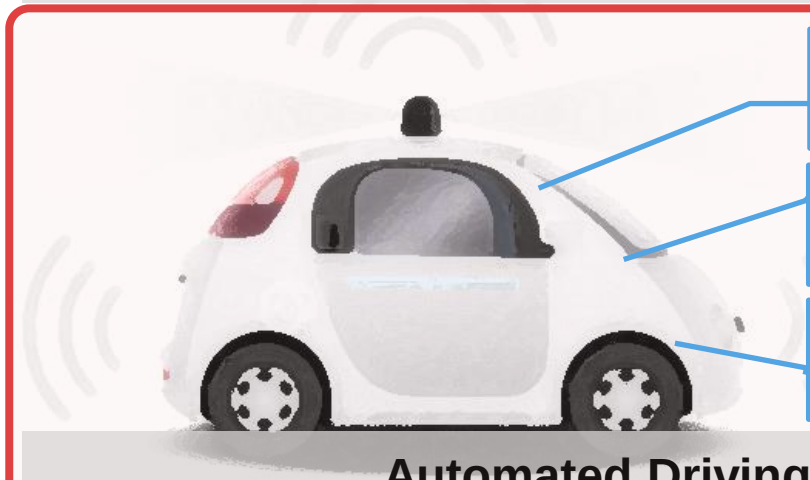
DLR Humanoid Robot

KRONES Dela Robot

Robot Arm
FESTO

Tactile Object Recognition
Institute of Science and Technology NAIST

Manipulator – Robot Arm



Automated Driving

Self Driving Taxi
voyage

AEBS
SCANIA

Automated Driving

Autonomous Carrier
MUSASHI

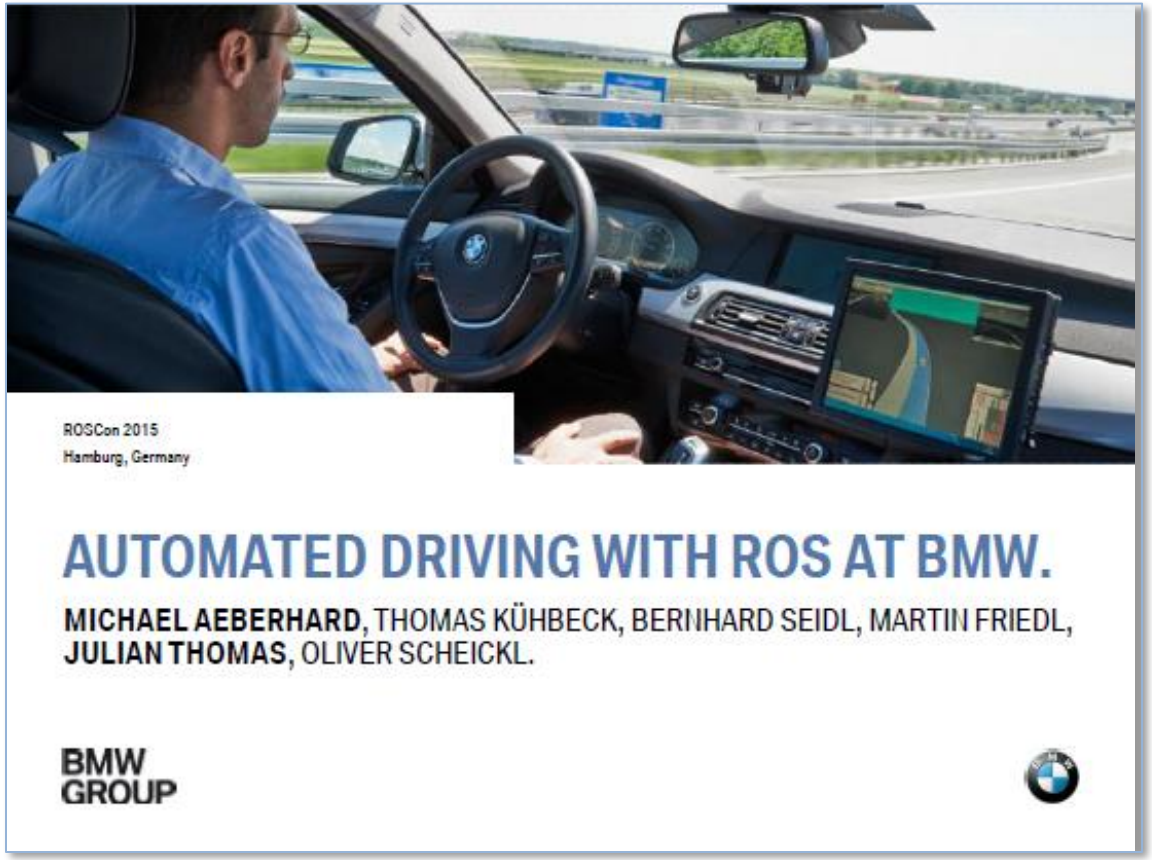
Material Handling
VANDERLANDE

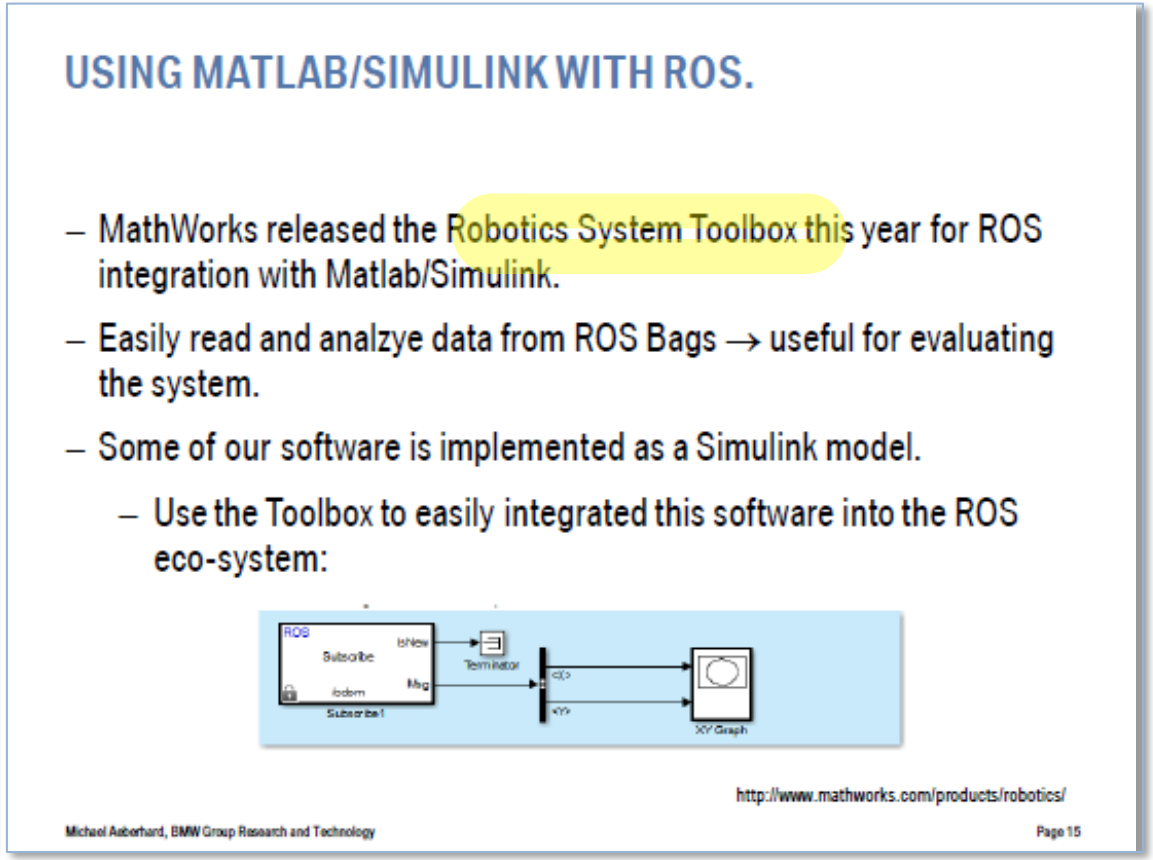
IAI Aircraft Tractor
Israel Aerospace

Self-Driving Vehicle
CLEARPATH ROBOTICS

Autonomous Ground Vehicle (AGV)

User Presentation
Automated Driving with ROS at BMW





M. Aeberhard, et.al., ROSCon 2015, Hamburg, Germany, Oct. 3-4, 2015

PDF: http://roscon.ros.org/2015/presentations/ROSCon-Automated-Driving.pdf
Video: https://vimeo.com/142622935

Voyage develops longitudinal controls for self-driving taxis

Challenge

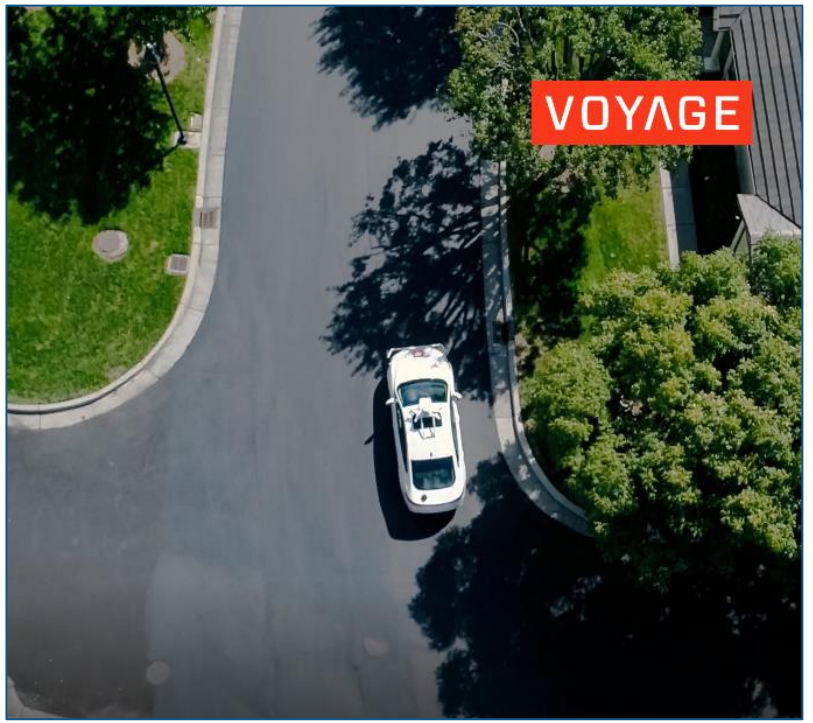
Develop a controller for a self-driving car to follow a target velocity and maintain a safe distance from obstacles

Solution

Use Simulink to design a longitudinal model predictive controller and tuned parameters based on experimental data imported into MATLAB using Robotics System Toolbox. Deploy the controller as a ROS node using Robotics System Toolbox. Generate source code using Simulink Coder into a Docker Container.

Results

- Development speed tripled
- Easy integration with open-source software
- Simulink algorithms delivered as production software



Voyage's self driving car in San Jose, California.

"We were searching for a prototyping solution that was fast for development and robust for production. We decided to go with Simulink for controller development and code generation, while using MATLAB to automate development tasks."

- Alan Mond, Voyage

Link to user story

Voyage Auto Develops Longitudinal Controls for a Self-Driving Taxi Using Model-Based Design

Challenge

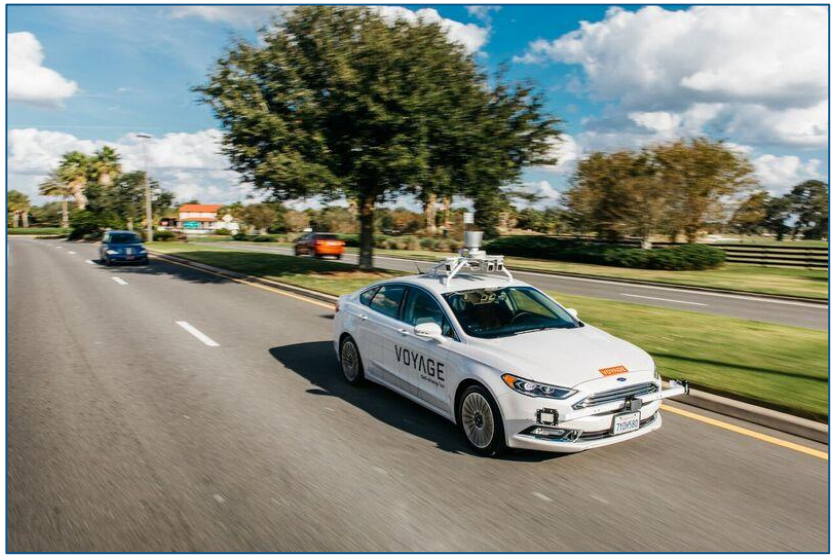
Reduce time-to-market for a self-driving taxi service by speeding control design and implementation

Solution

Use Model-Based Design to model, simulate, and generate C++ code for the taxi's longitudinal control system

Results

- Initial control design implemented in three days
- Gigabytes of ROS data imported, filtered, and analyzed
- ROS node for simulating virtual barriers developed in hours



A Voyage self-driving taxi on the road in Florida.

"After downloading [the MATLAB adaptive cruise control system example] model and running some preliminary simulations in Simulink, I generated C++ code from the model for a standalone ROS node with Robotics System Toolbox and Simulink Coder… Within three days we were running the generated code for the ACC in our vehicle."

- Alan Mond, Voyage Auto

Link to technical article

Tata Motors Accelerates Development of Autonomous Vehicle Control Algorithms with Model-Based Design

Challenge

Build and demonstrate an autonomous vehicle for the UK Autodrive project

Solution

Use Model-Based Design to model, simulate, and generate embedded code for motion planning and vehicle control algorithms

Results

- Real-time controller implementation accelerated
- Debugging simplified
- Developmental time focused on design

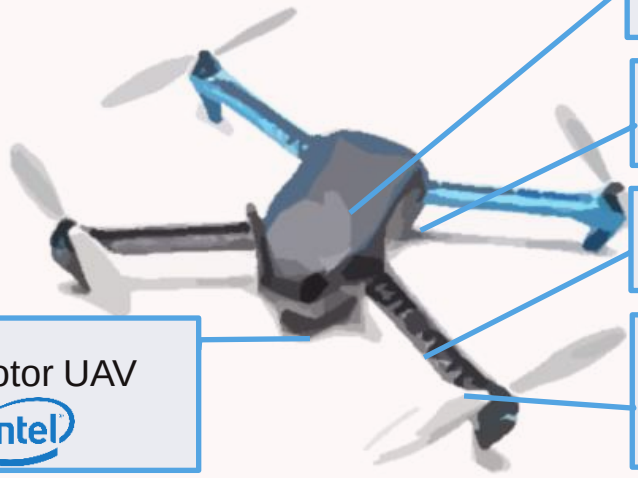
Trials for Tata's autonomous vehicle in Coventry, UK.

"A small team of engineers pulled together an autonomous vehicle with off-the-shelf hardware and control algorithms developed and implemented with Model-Based Design. Though the system isn't production-ready, it does demonstrate important design concepts with a pragmatic design approach."

- Dr. Mark Tucker, TMETC

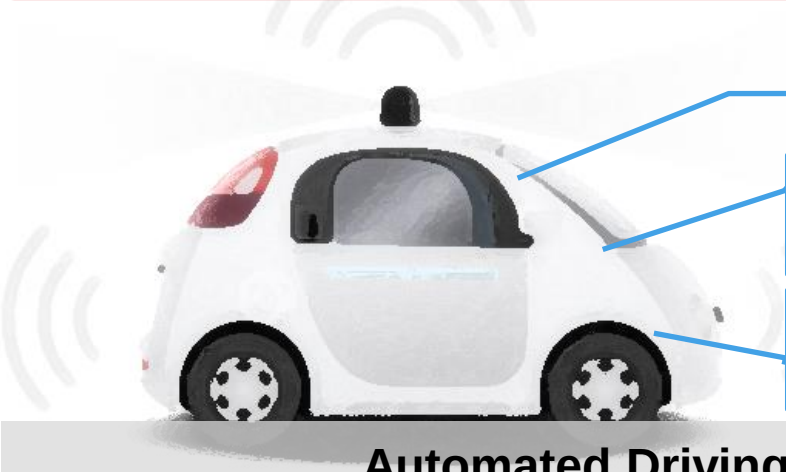
Link to user story

User Example: Robotics and Autonomous System Development



- UAV — KOREAN AIR
- Hexacopter — intelinair
- Quadcopter GNC — NASA Interns
- Multirotor UAV — intel
- Fly-by-Wire Controller — airnamics BY KOLEKTOR

Unmanned Air Vehicle (UAV)



- Automated Driving
- Self Driving Taxi — voyage
- AEBS — SCANIA

Automated Driving

- Eye Surgery Robot — PRECEYES
- Robot Arm Braking System — 3T
- Humanoid Robot — DLR
- Robot Arm — FESTO
- Tactile Object Recognition — National Institute of Science and Technology NAIST

- Industrial Robot — YASKAWA
- Large-Scale Robot Arm — MITSUBISHI HEAVY INDUSTRIES
- Robot actuator — HEBI ROBOTICS
- KRONES Dela Robot

Manipulator – Robot Arm

- Autonomous Carrier — MUSASHI
- Aircraft Tractor — IAI Israel Aerospace
- Self-Driving Vehicle — CLEARPATH ROBOTICS

- Material Handling — VANDERLANDE

Autonomous Ground Vehicle (AGV)

Yamaha Motor Co., Ltd. develops series hybrid system for drones and executes flight testing using Model-Based Design

Challenges

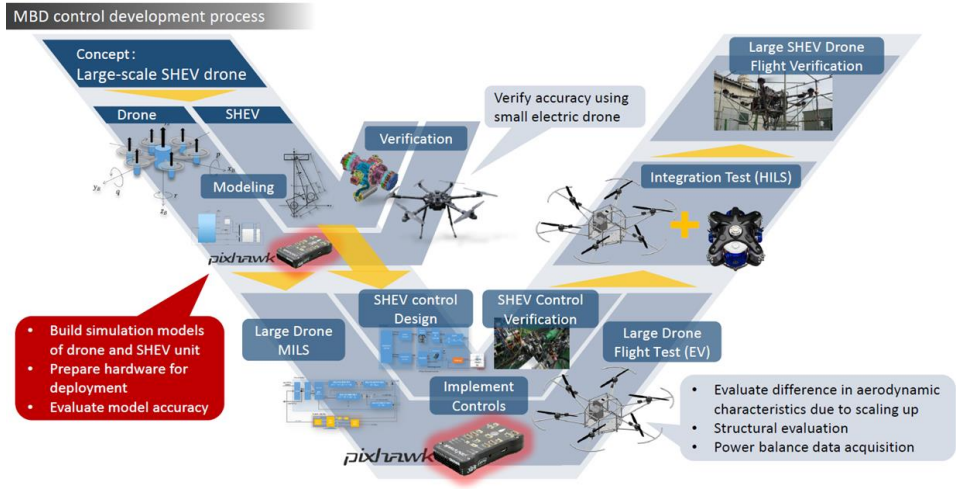
- Development of a Series Hybrid (SHEV) system for implementing on a large-scale drone

Solution

- Identify airframe and propellers parameters for modeling
- Implement flight algorithms onto Pixhawk® using automatic code generation
- Integrated testing of SHEV system and drone (HILS)

Results

- Verify control system in simulation by using drone and SHEV models
- Efficient and safe flight testing by utilizing automatic code generation and hardware integration test (HILS)



Large-scale SHEV drone development V-process

By modeling and simulating the drone and series hybrid system, it was possible to evaluate the control system virtually. We were able to utilize Model-Based Design through the V-process by using UAV Toolbox and Pixhawk®.

Details (Japanese video)

RangeAero Develops Pilotless Helicopter to Support Last-Mile Deliveries

Challenge

Supply chain slowdowns, e-commerce growth, and labor shortages put strain on last-mile delivery of large payloads

Solution

Develop coaxial helicopter using MATLAB and Simulink to design workflows with Model-Based Design and analyze flight data, systems health, and communications logs. Create “what if” scenarios to identify and correct deficiencies and weaknesses.

Key Outcomes

- The RangeAero small footprint autonomous coaxial helicopter can land and take off in congested areas with limited space and where precise delivery is required
- Scaled versions with payload capacity up to 10 kg pave the way for the Range100, designed to carry 100 kg for 124 miles

The RangeAero autonomous helicopter delivers large payloads with precision. (Image credit: RangeAero)

“When we developed the brain of the aircraft, including things like control systems and other integrated systems, we used MATLAB intensively. We wanted to develop rapidly, test it, and deploy it.” — Rohit Gupta, co-founder and CTO of RangeAero

Link to feature story

DCNS Models and Simulates SAMAHE Helicopter Handling System

The SAMAHE helicopter handling system.

Challenge

Accelerate the development and qualification of a handling system for moving helicopters between the flight deck of a navy ship and an onboard hangar

Solution

Use Simulink and SimMechanics to model the handling system, helicopter, and ship and to simulate the system under conditions that would be difficult or unsafe to test at sea

Results

- System stability verified under numerous operating conditions
- Simulations accelerated by a factor of 18
- Critical development knowledge transferred

"MATLAB and Simulink helped us to quickly verify technical concepts, accurately estimate the performance of our design, minimize system qualification time, and deliver a system that answers program needs."

Jérôme Vouillat
DCNS

NASA Interns Develop Guidance, Navigation, and Control Software for Quadcopter with Model-Based Design

Challenge

Give engineering interns hands-on work experience in the aerospace field

Solution

Create a program in which interns use Model-Based Design to model, simulate, implement, and fly GNC algorithms for multirotor aircraft

Results

- GNC algorithms developed and implemented in 10 weeks
- Hardware integration streamlined
- Practical engineering experience acquired

NASA intern working with the quadcopter vehicle and ArduPilot Mega 2.5 hardware.

Model-Based Design makes both working engineers and interns at NASA MSFC more productive. The students have more fun because they can run the GNC algorithms they create in Simulink on a real processor and quickly get things done.

Link to user story

Airnamics Develops Unmanned Aerial System for Close-Range Filming with Model Based Design

Challenge

Design and develop an unmanned aerial camera motion system for close-range aerial filming

Solution

Use Model-Based Design with MATLAB and Simulink to accelerate the design, debugging, and implementation of the vehicle's fly-by-wire and flight management system software

Results

- Time-to-market shortened by up to an order of magnitude
- Test flight anomalies quickly resolved
- Debugging time reduced from weeks to hours

Airnamics co-founders Marko Thaler and Zoran Bjelić with the R5 MSN1 prototype after its first flight.

"With Model-Based Design our three-engineer team found more than 95% of control software bugs before the first flight. We used the test flights to increase our Simulink models' fidelity and isolate remaining bugs with high precision. The result is a safer, more reliable, and higher-quality product."

- Marko Thaler, Airnamics

Link to user story

Flying-Cam Develops Autonomous Mini-Helicopter Controller with MathWorks Tools

Flying-Cam's autonomous mini-helicopter capturing footage from a Formula 1 test run.

Challenge

Develop an autonomous helicopter control to support aerial camera shots in the film industry

Solution

Use MathWorks tools to design a complete control system, automatically generate code, and run hardware-in-the-loop simulations

Results

- Development time reduced
- Real-time controller implemented without errors
- Learning curve eliminated

"With MathWorks tools, I know that if the helicopter is working on my laptop, then the real-time implementation will work, too."

Marco La Civita
Flying-Cam

Link to user story

Intel Creates Dynamic Simulation Environment for Testing GNC Algorithms for Multirotor UAVs

Challenge

Develop and test advanced control algorithms for existing and next-generation multirotor UAVs

Solution

Use MATLAB and Simulink to model multirotor UAV dynamics, verify control algorithms via simulation, and evaluate control design ideas

Results

- Complex calculations verified before flight testing
- Design ideas evaluated in days, not weeks
- Design iterations and testing time reduced

The eight-rotor Intel® Falcon™ 8+.

"Modeling and simulation with Simulink is the only way that we can get the results we need with the speed and quality that's expected in our industry today. If we had to do everything by hand and rely solely on flight testing, we would require more bug fixing iterations and need more testing time per iteration. The problem would grow intractable. There's no other way."

- Jan Vervoorst, Intel

Link to user story

Airbus Helicopters Accelerates Development of DO-178B Certified Software with Model-Based Design

The Airbus Helicopter EC130 helicopter.

Challenge

Speed the development, validation, and verification of DO-178B certified helicopter flight software

Solution

Use Model-Based Design to model the system design and software design, and to generate flight code

Results

- Software testing time cut by two-thirds
- Requirements stabilized earlier
- Certified flight software automatically generated

"We use our system design model in Simulink for ARP4754 to establish stable, objective requirements. We save time by using the model as the basis for our software design model for DO-178—from which we generate flight code—and reusing validation tests for software verification."

Ronald Blanrue
Airbus Helicopters

Link to user story

MathWorks® is a **Leader** in the Gartner Magic Quadrant for 2020 Data Science and Machine Learning Platforms

MathWorks has the furthest completeness of vision in the Leaders quadrant

We believe this recognition demonstrates our ability to:

- Empower your team, including those with limited AI or data science experience
- Provide complete workflows for data preparation, AI modeling, system design, and production
- Deploy AI models on embedded devices, edge, enterprise systems, and the cloud
- Use Simulink to tackle integration challenges and reduce risk in designing AI-driven systems

*Gartner Magic Quadrant for Data Science and Machine Learning Platforms, Peter Krensky, Erick Brethenoux, Jim Hare, Carlie Idoine, Alexander Linden, Svetlana Sicular, 11 February 2020 .

MathWorks®

is a **Leader** in the Gartner Magic Quadrant for 2020 Data Science and Machine Learning Platforms

Figure 1. Magic Quadrant for Data Science and Machine Learning Platforms

Source: Gartner (February 2020)

*Gartner Magic Quadrant for Data Science and Machine Learning Platforms, Peter Krensky, Erick Brethenoux, Jim Hare, Carlie Idoine, Alexander Linden, Svetlana Sicular, 11 February 2020 .