

2023年9月11日(月)

# 実例から学ぶMATLABおよびSimulinkの ロボティクス開発ソリューション

MathWorks Japan  
能戸フレッド

# MATLAB® & SIMULINK®



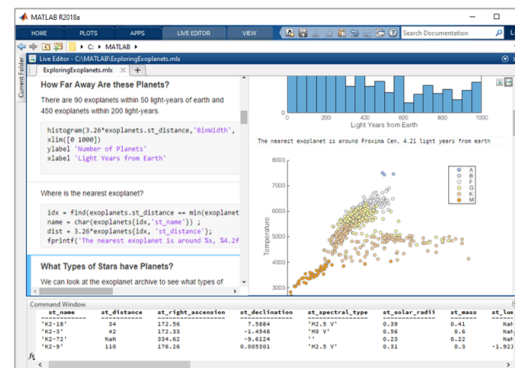
## 製品

MATLABは、アルゴリズム開発、データ解析、可視化、数値計算のためのプログラミング環境

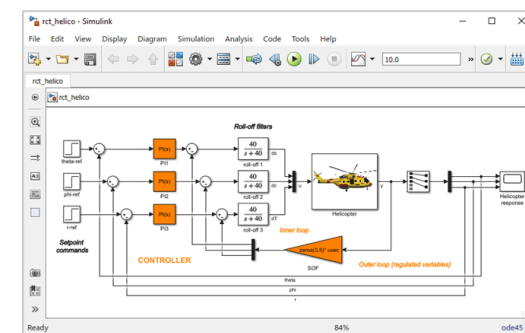
Simulinkは、システム的设计、シミュレーション、テストを行うためのグラフィカル環境

100以上の専門タスクに特化したアドオンツール

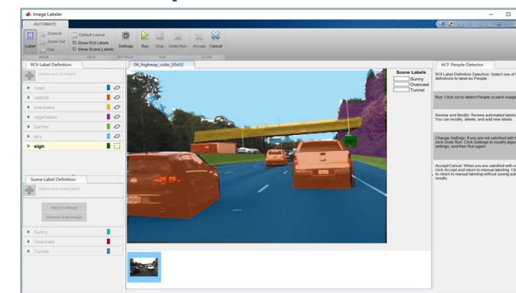
### MATLAB



### Simulink



### Computer Vision Toolbox



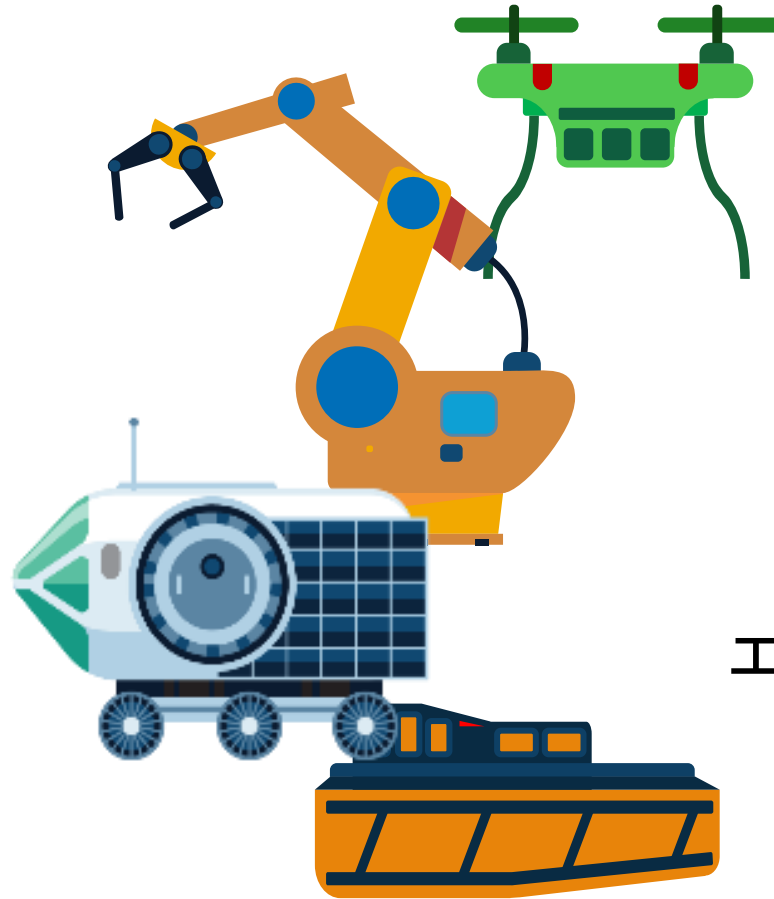
# 自律ロボティクス開発の課題



マルチドメイン  
知識の必要性



アルゴリズム  
複雑性



安全性・品質の  
確保



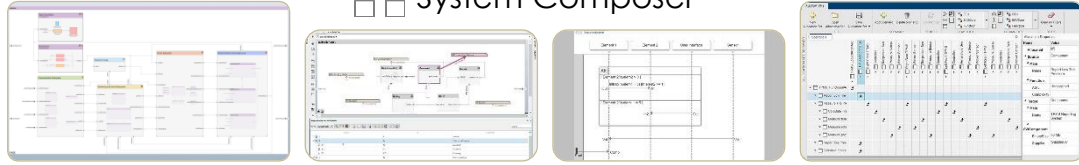
エンド・トゥ・エンド  
ワークフロー





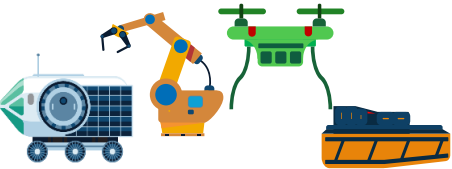
# 自律ロボティクス開発ワークフロー

**MATLAB® Simulink®**  
システムエンジニアリング  
System Composer™

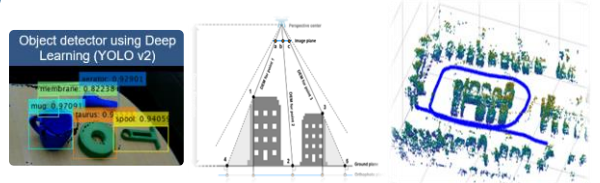


プラットフォーム設計

AI・自律系設計



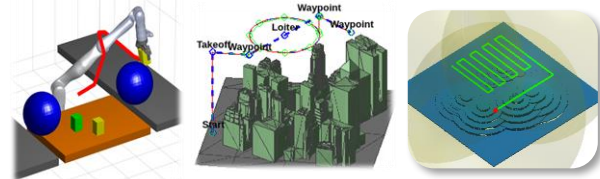
ロボットモデル



認識・自己位置推定

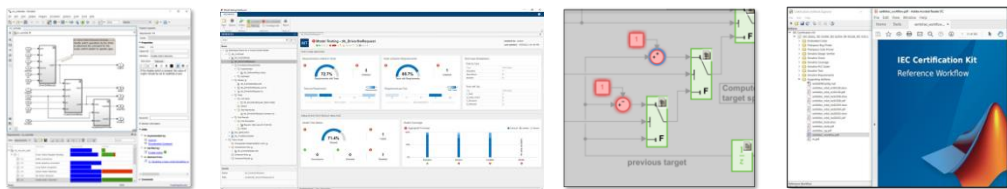


制御



モーションプランニング

検証と妥当性確認



連成

3Dシミュレーション



シーン・環境

オブジェクト・アクター

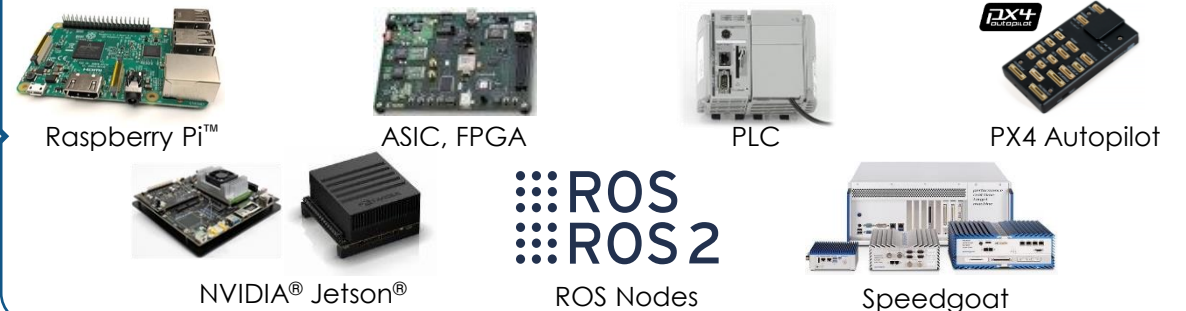
テスト

ロボットとの接続



実装

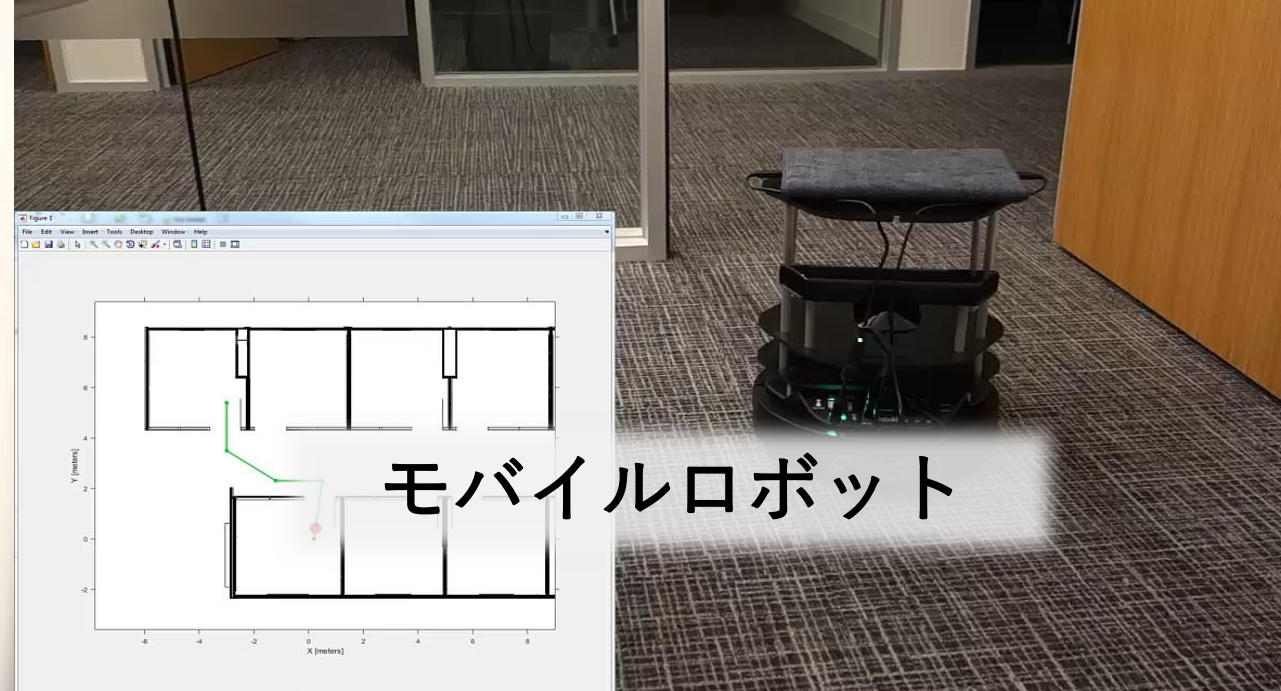
ハードウェアへの実装



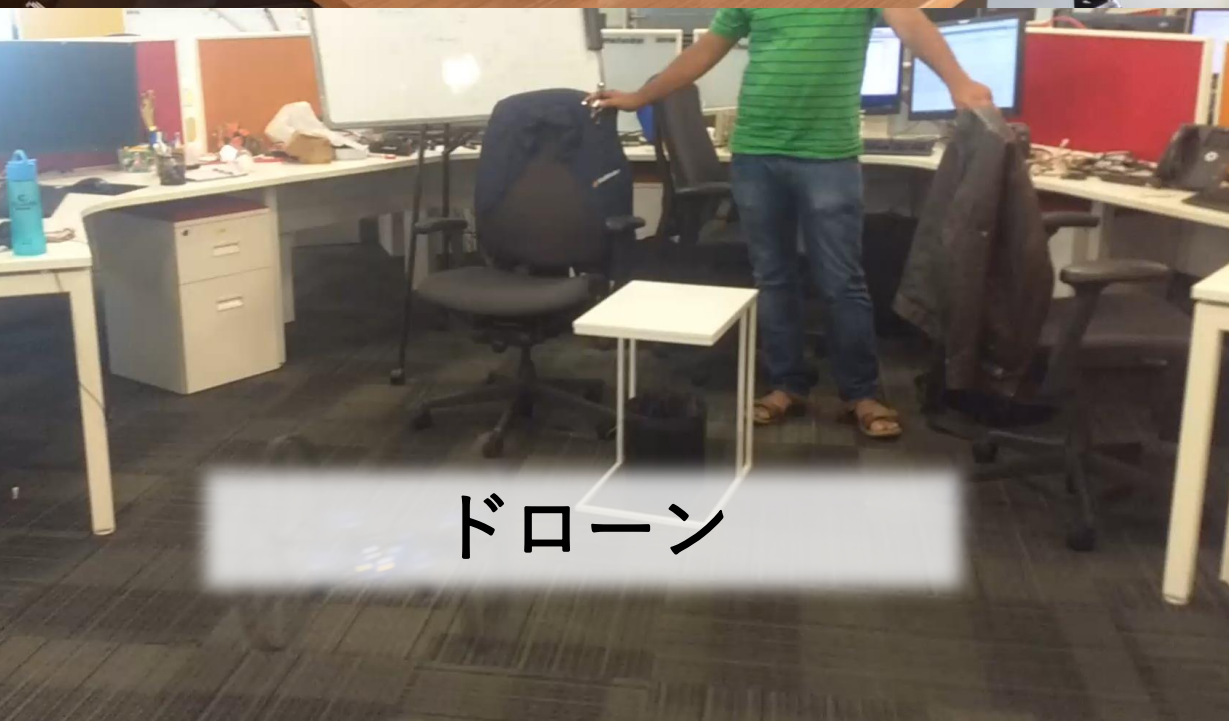




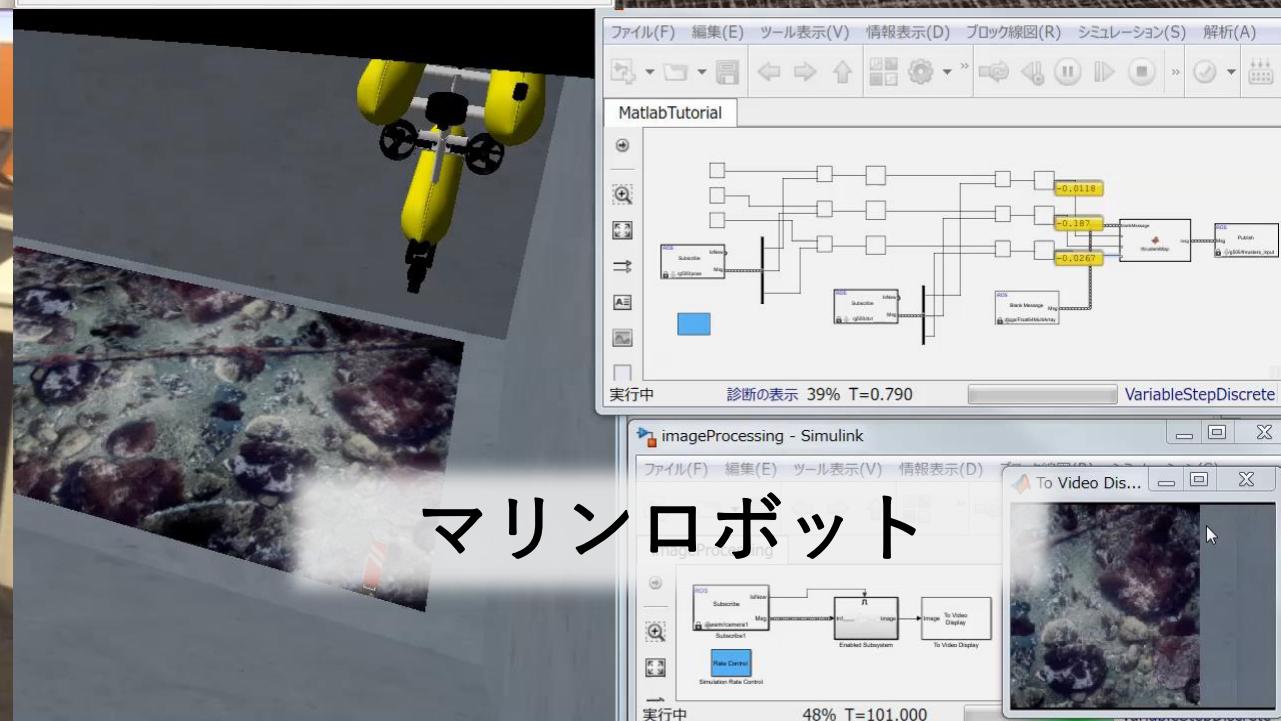
マニピュレータ



モバイルロボット



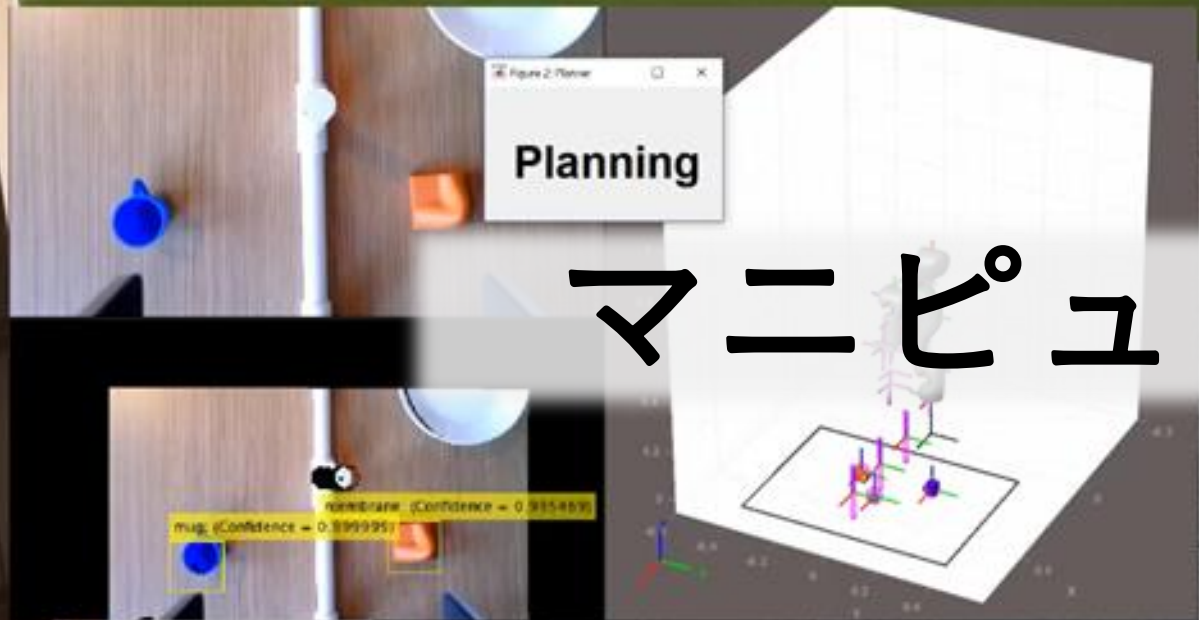
ドローン



マリンロボット



# マニピュレータ



# 京セラ、モデルベースデザインによる人協働多軸ロボットの開発環境を構築

## 課題

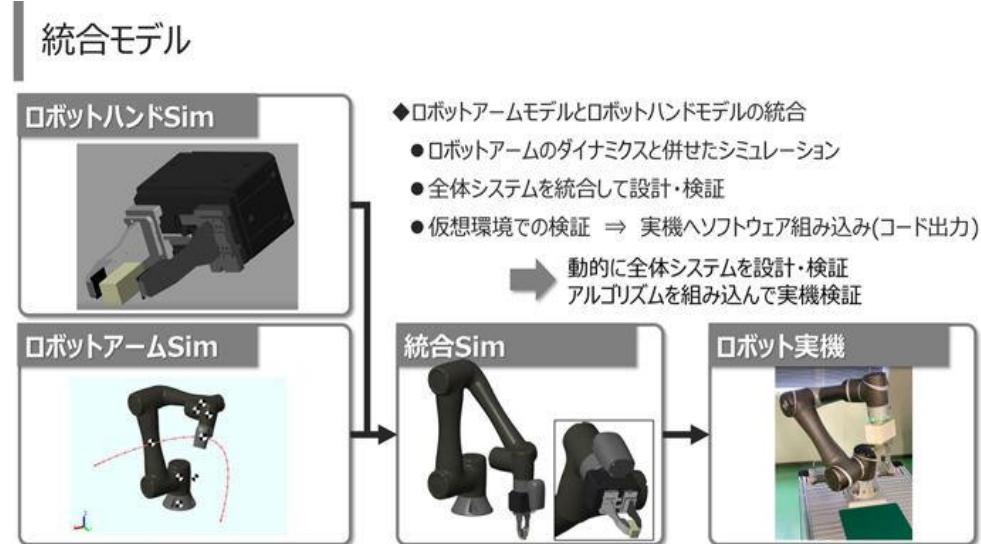
人協働多軸ロボットを用いた多品種生産の現場では、専用ロボットハンドの設計と試作、実機試験を繰り返すことが多く、生産工程の迅速な立ち上げが求められた

## ソリューション

- MATLABを活用し、ロボットコントローラ的设计における逆運動学や軌道計画、ロボットハンドやワークとの接触モデルを構築、実機ロボットと連携
- モデルを統合し、試作レスで把持動作を検証

## 結果

- 試作開発の設計から検証までにかかる期間を削減
- バーチャルからリアルまでシームレスに対応できる設計・検証環境を構築
- ROS対応ロボットとの容易な連携



ハンドとアームを統合したシミュレーションおよび実機実装

MATLABとSimulinkのよいところは製品機能の高さと、多彩なインターフェースを持つことです。仮想開発環境を構築し、さらにROSと連携することで迅速な実機展開が可能となりました。

# 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

[Link to technical article](#)



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)*



# JUNIA Develops Autonomous Pediatric Exoskeleton for Children with Severe Neurological Disorders

## Challenge

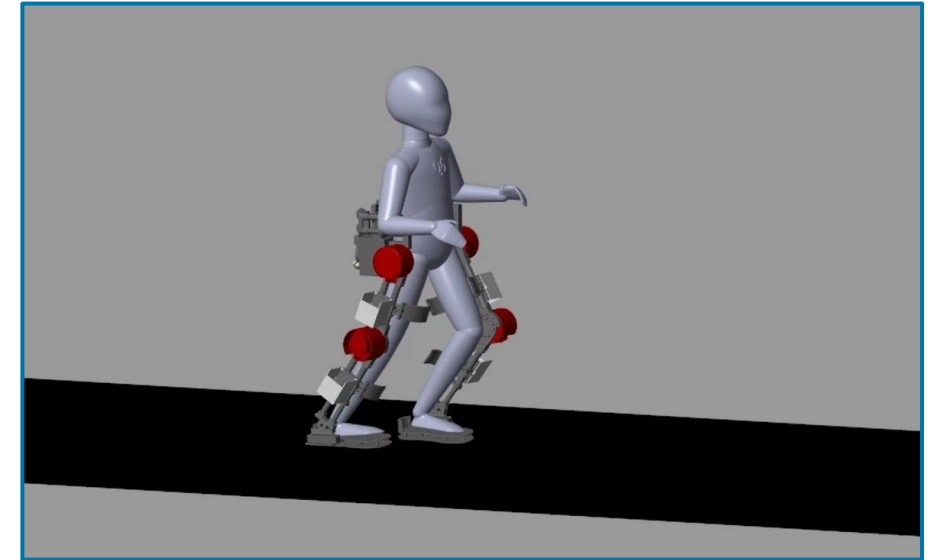
Develop a lower limb exoskeleton to be used as a physical therapy tool for children with cerebral palsy.

## Solution

Use MATLAB, Simulink, and Simscape Multibody to model motor dynamics and design motor controllers. Conduct real-time testing of prototype hardware with Speedgoat.

## Key Outcomes

- Development was accelerated through Model-Based Design with MATLAB and Simulink
- Total prototyping time was cut in half using Simulink Real-Time and Speedgoat real-time target machine
- Collaboration was successful between clinicians and partners in exoskeleton testing using the designed interface with MATLAB App Designer



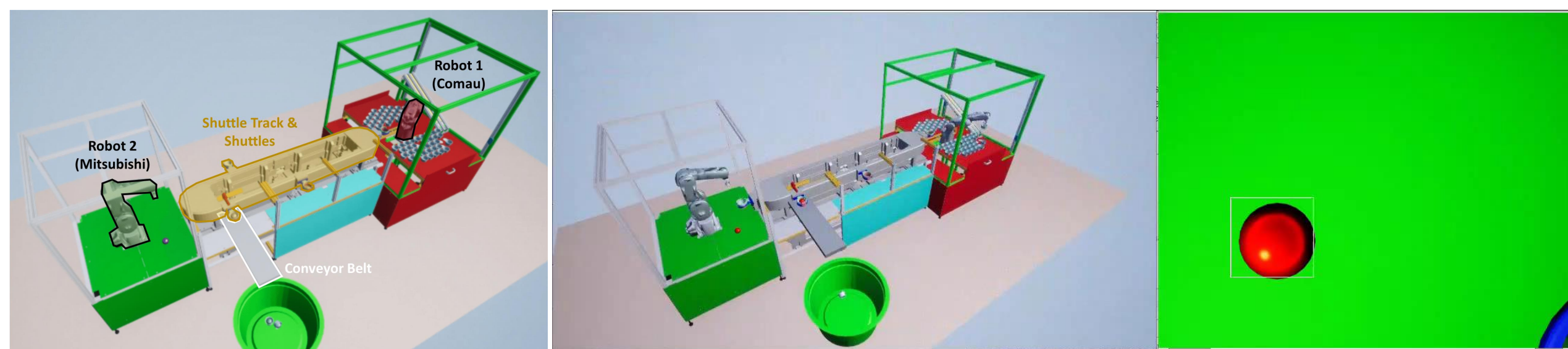
Model of the JUNIA exoskeleton. (Image credit: JUNIA HEI)

*“We found that Simulink Real-Time is the best choice because you can design whatever model you want, whatever control algorithm you want, and you can apply it quickly to your prototype.”*  
— Yang Zhang, postdoctoral researcher at JUNIA HEI

# ゲーミングエンジン連携による3Dシミュレーション

ロボットモデルをインポートして3Dシミュレーションで動作確認

- システムの挙動を3次元空間で可視化し直感的に動作を確認
- センサーモデルも含めた閉ループシミュレーションによる検証



[Automate Virtual Assembly Line with Two Robotic Workcells - MATLAB & Simulink - MathWorks 日本](#)

ドローン





# ヤマハ発動機、ドローン向けシリーズハイブリッドシステムをモデルベースデザインで開発し、飛行試験を実施

## 課題

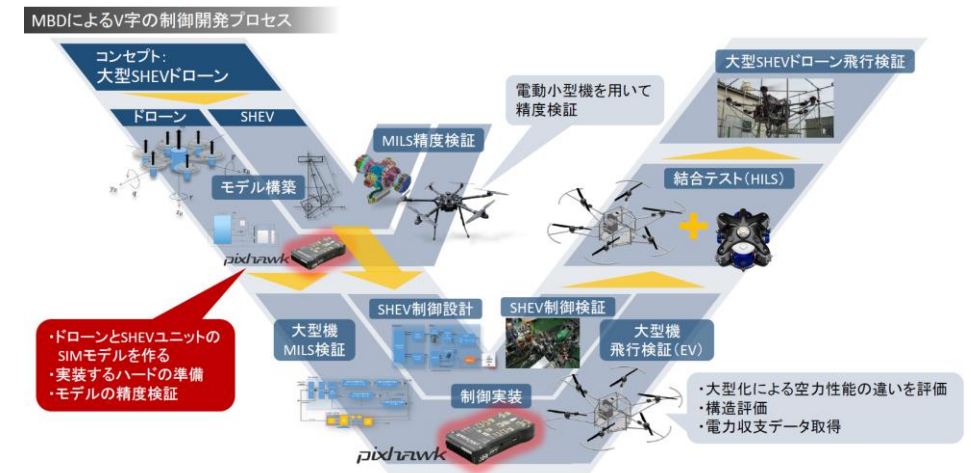
- ドローン向けのシリーズハイブリッド(SHEV)システムを開発し大型ドローンでの搭載検討を行う

## ソリューション

- プロペラや機体の特性を同定しモデルを構築
- 飛行アルゴリズムを自動コード生成でPixhawk®に実装
- シリーズハイブリッドシステムと結合テスト(HILS)

## 結果

- ドローンおよびシリーズハイブリッドシステムをモデル化することで机上で制御成立性を検証
- 自動コード生成と結合テスト(HILS)により実機試験を安全かつ効率的に実施



大型SHEVドローンのためのV字制御開発プロセス

ドローン、シリーズハイブリッドシステムをモデル化することで制御の机上検討が可能となりました。  
UAV ToolboxとPixhawk®を活用することでVプロセスに沿ったドローンのモデルベースデザインを実現することができました。

# University of Naples Researchers Simulate Autonomous Landing for UAVs

## Challenge

Improve autonomous approach and landing for UAVs in challenging, low-visibility conditions such as fog, rain, or night

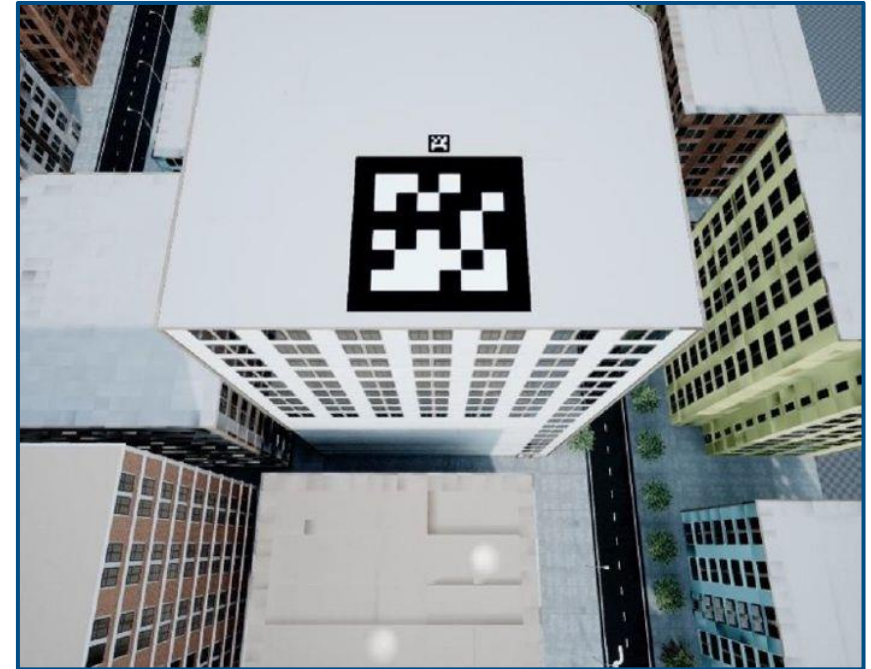
## Solution

Develop perception algorithms using MATLAB and simulate landing conditions using Simulink to support design, development, and testing of solutions

## Results

- Developed vision-aided, multisensor-based navigation architecture to support autonomous UAV landing
- Created simulations that are easier to design and control, can include a wide variety of system parameters, and are better representative of real-world urban environments
- Developed perception algorithms capable of producing reliable pose estimates under low-visibility conditions and capable of crosschecking the integrity of GNSS measurements

[Link to article](#)



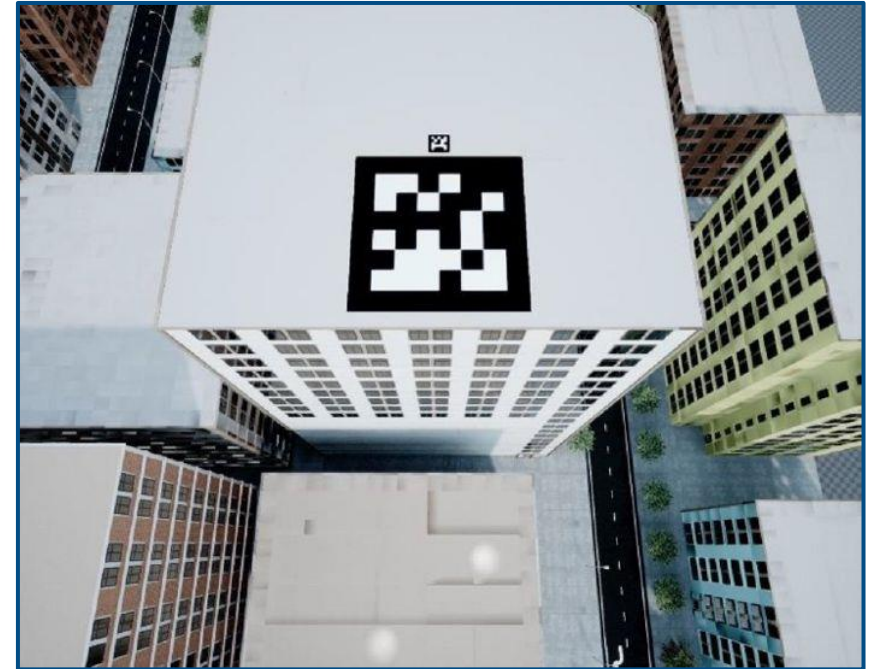
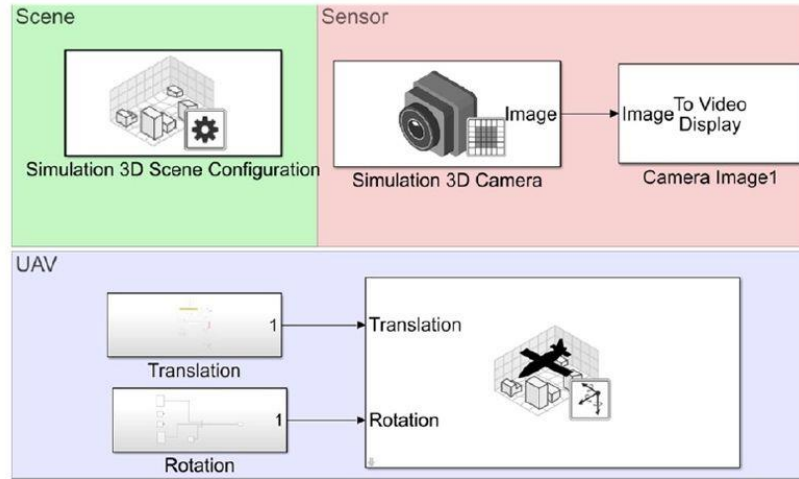
A landing pad equipped with an AprilTag marker to facilitate an autonomous aircraft's pose estimation.

*“One key feature of MATLAB that makes it very useful for our work is the support, help, and functionality of MATLAB, which is very clear, comprehensive, and easy to use. For us and our students, this makes MATLAB useful.”*

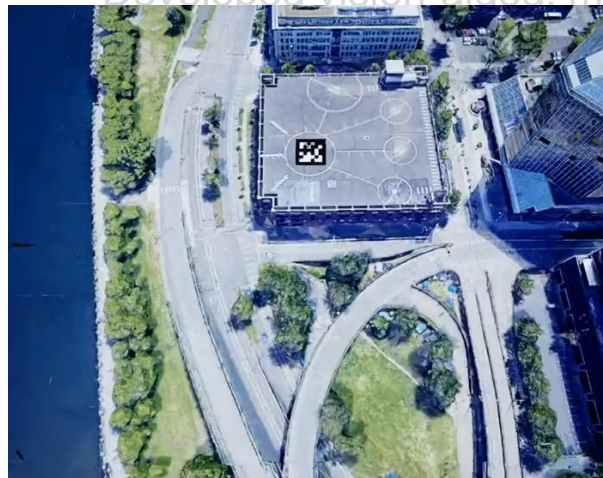
*- Roberto Opromolla, PhD, University of Naples*

# University of Naples Researchers Simulate Autonomous Landing for UAVs

Unreal Engine連携  
機能を使用した様々な  
テストケースでの  
自律着陸アルゴリズム  
の評価



A landing pad equipped with an AprilTag marker to facilitate an autonomous aircraft's pose estimation.



*“One key feature of MATLAB that makes it very useful for our work is the support, help, and functionality of MATLAB, which is very clear, comprehensive, and easy to use. For us and our students, this makes MATLAB useful.”*

*- Roberto Opromolla, PhD, University of Naples*

[Link to article](#)





# Cesiumとの連携

## Unrealのための3次元地理空間システム

- WGS84測地系でのフルスケールかつ地理空間的に正確なモデル
- 地上から軌道高度までシームレスに視覚化
- 実行時に指定された場所の現実世界の3次元地形トポロジを作成
- シーンを正確なマップ テクスチャでオーバーレイ
- 実行時にシーン内に建物を生成



# センサーモデルと統合されたシミュレーション

The image is a composite of three windows from a MATLAB/Simulink simulation:

- Top Left: Simulink Model Editor**
  - Menu: SIMULATION, DEBUG, MODELING, FORMAT, APPS
  - Toolbox: Open, Save, Library Browser, Log Signals, Add Viewer, Signal Table, Stop Time, Step Back, Pause, Step Forward, Stop, Data Inspector, Logic Analyzer, Bird's-Eye Scope.
  - Model Browser: simFlight20b
  - Model Diagram:
    - Inputs: pos, orientEuler
    - Blocks: Translation, Rotation, Simulation 3D Scene Configuration, Simulation 3D Camera, Simulation 3D Lidar
    - Outputs: Image, Point cloud
    - Final Blocks: Image -> Video Viewer, Point cloud -> xyzPoints -> VisualizePointCloud
  - Status: Running, 109%, T=24.200, 24%
- Top Right: Video Viewer**
  - Menu: File, Tools, View, Simulation, Help
  - Content: A 3D rendered street scene with buildings and a drone flying in the distance.
  - Status: Running, RGB: 720x1280, T=24.000
- Bottom Right: Figure 1: Point Cloud Player**
  - Menu: File, Edit, View, Insert, Tools, Desktop, Window, Help
  - Content: A 3D point cloud visualization of the street scene, showing the drone's sensor data as a blue and green point cloud.
  - Axes: X, Y, Z
- Bottom Left: Full View of Street Scene**
  - A large, detailed 3D rendered view of the street scene, showing buildings, a road, and a drone flying in the center.



# リアルタイムハードウェアによるHILSテスト

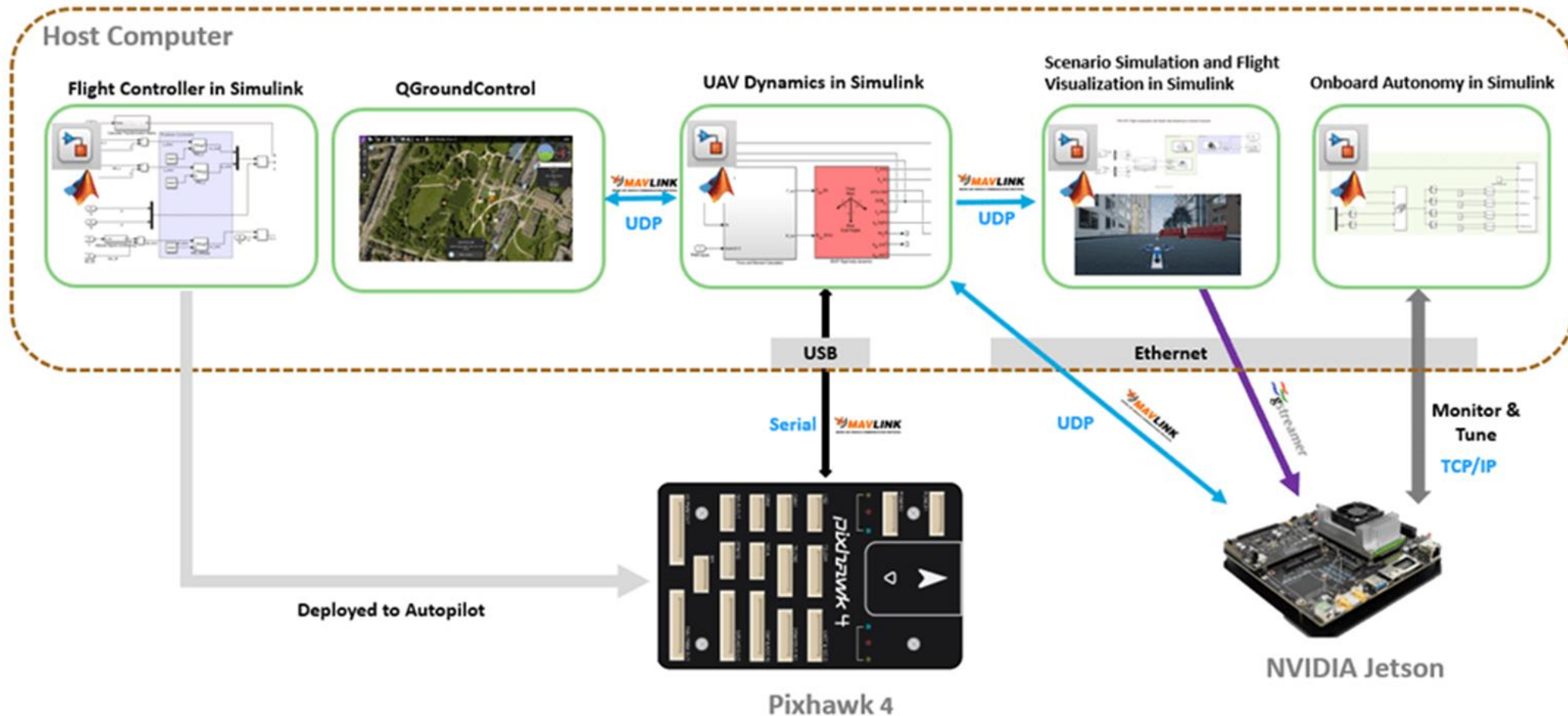
PX4オートパイロットへの実装

Unrealシミュレーション実行

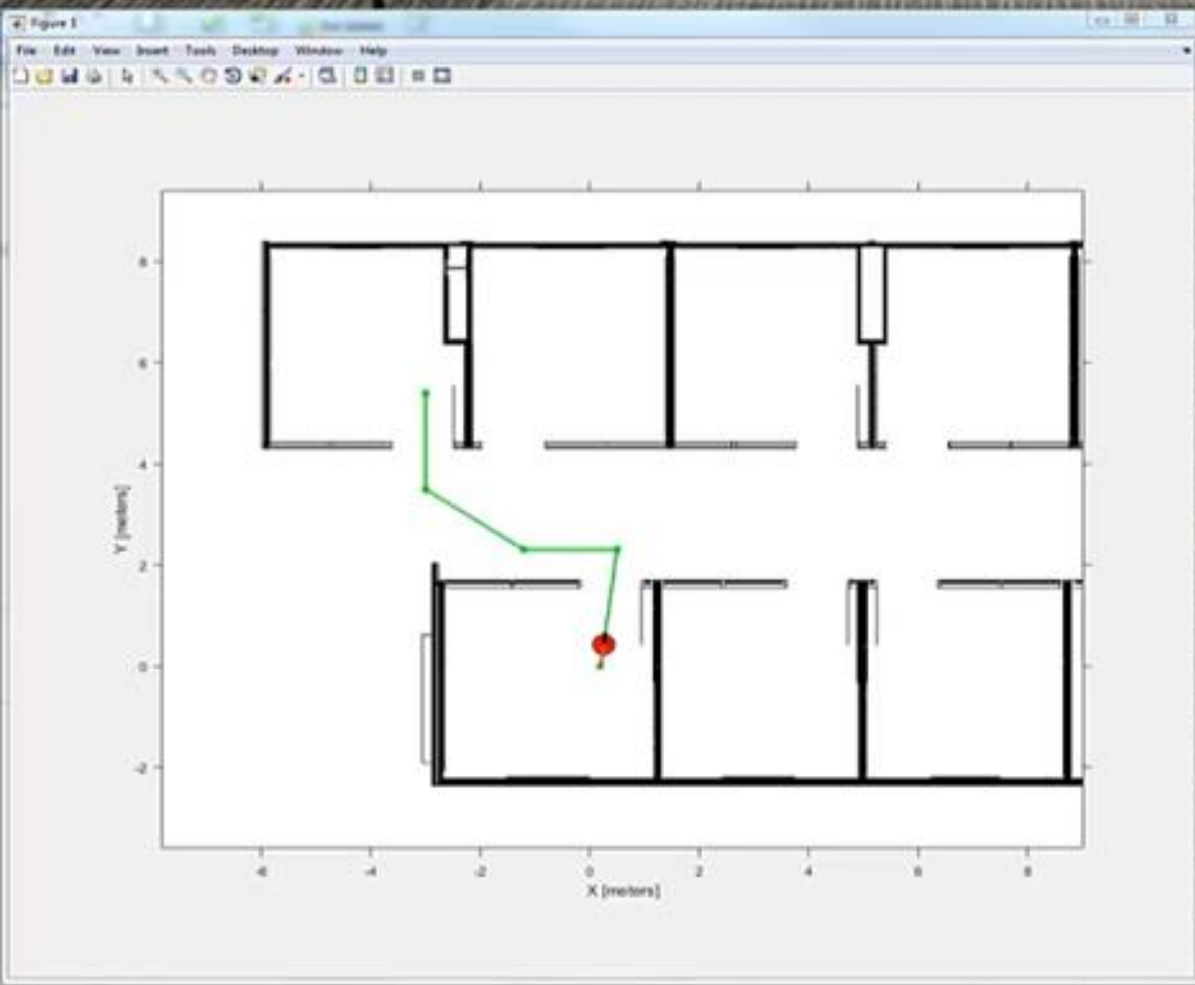
Jetsonに自律アルゴリズムを実装

Simulink機体ダイナミクスを実行

QGroundControlでミッション開始



# モバイルロボット



# Airbus Develops Autonomous Control Systems for the Mars Sample Fetch Rover Using Model-Based Design

Airbus Defence and Space engineers are using Simulink and Embedded Coder to accelerate the development of computer vision, machine learning, and autonomous control algorithms for the Mars Sample Fetch Rover.

## Key Outcomes

- Design, verification, and testing of autonomous control software accelerated
- Human errors in implementation reduced via code generation
- Machine learning-based visual detection algorithms developed to locate samples



As part of the Mars Sample Return (ESA-NASA) mission, Airbus Defence and Space UK is developing the Mars Sample Fetch Rover.

*Fast development is key in this project. Airbus has been developing algorithms in MATLAB that serve as the basis for automatically generating the flight software, as well as all the documentation that is needed.*



# MIT Develops More Reliable Algorithm for Robots Navigating New Spaces

## Challenge

Ensure that simultaneous localization and mapping (SLAM) algorithms produce correct results in challenging real-world conditions

## Solution

Develop the graduated nonconvexity (GNC) algorithm, which reduces the random errors and uncertainties in SLAM results

## Key Outcomes

- GNC produces correct results where existing methods “get lost.”
- GNC algorithm made available to MATLAB users in Navigation Toolbox.



A delivery robot.

*In real applications, the robot faces many outliers, which can make up more than 90% of all observations. That's where the GNC algorithm comes in and outperforms all competitors.*

# デジタル標高図を使ったオフロードパスプランニング

地表面を定義

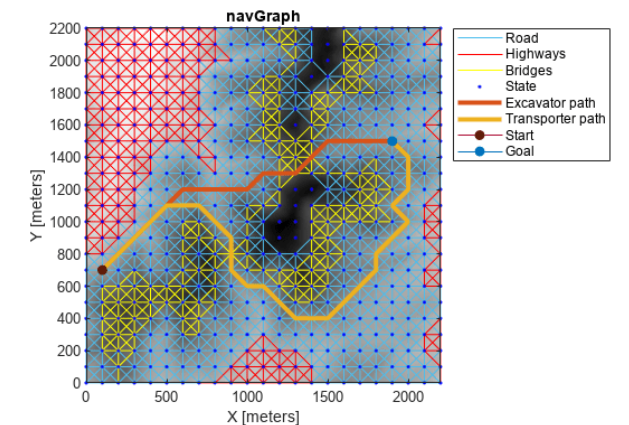
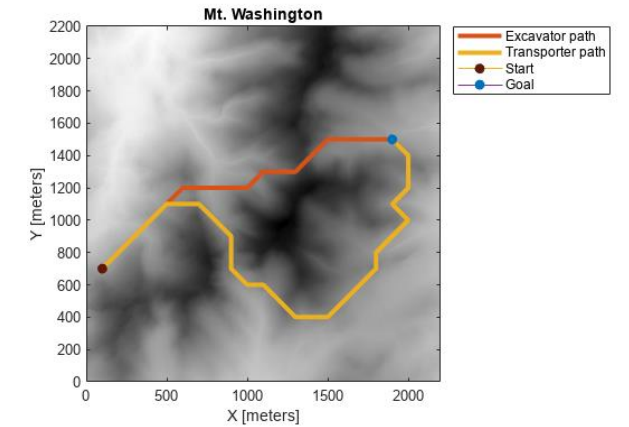
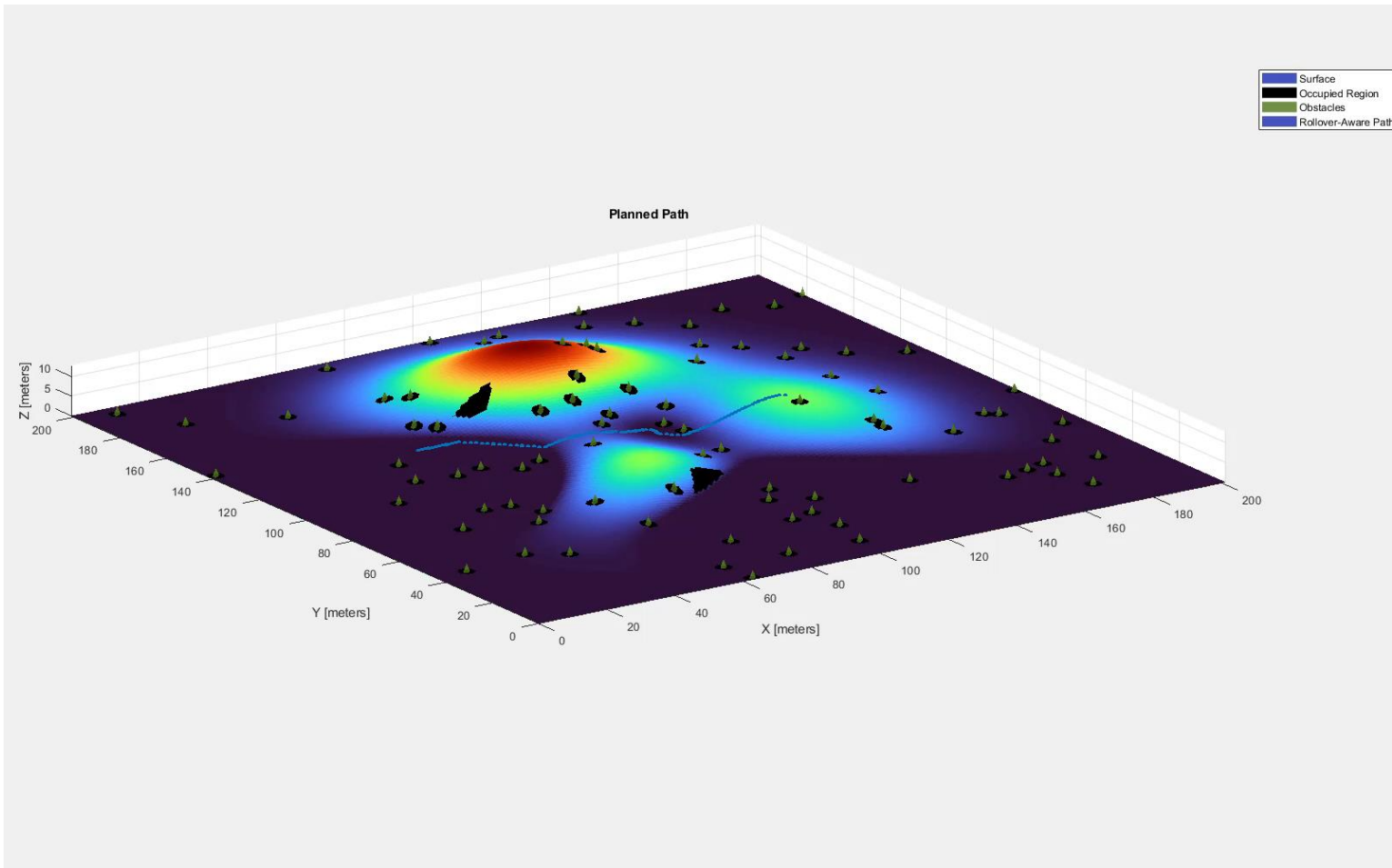
障害物を定義

マップレイヤ  
に格納

パスプラン  
ニング

結果の比較  
解析

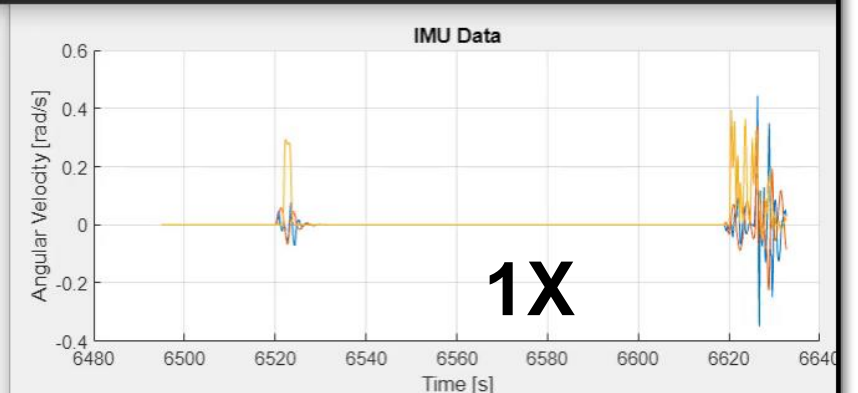
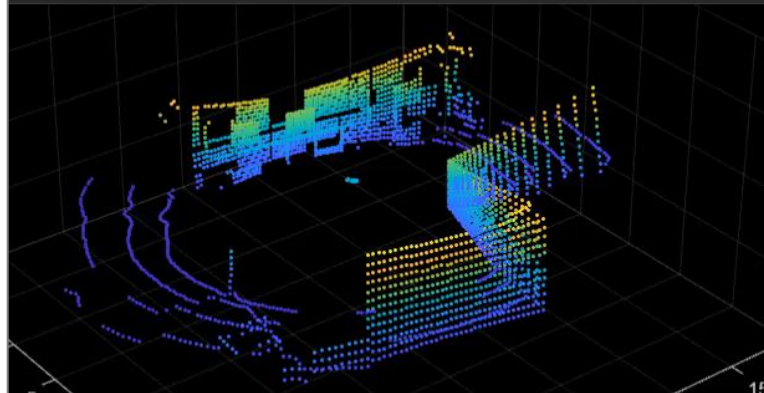
現実のDEM  
上でのプラン  
ニング



[Link to Example](#)

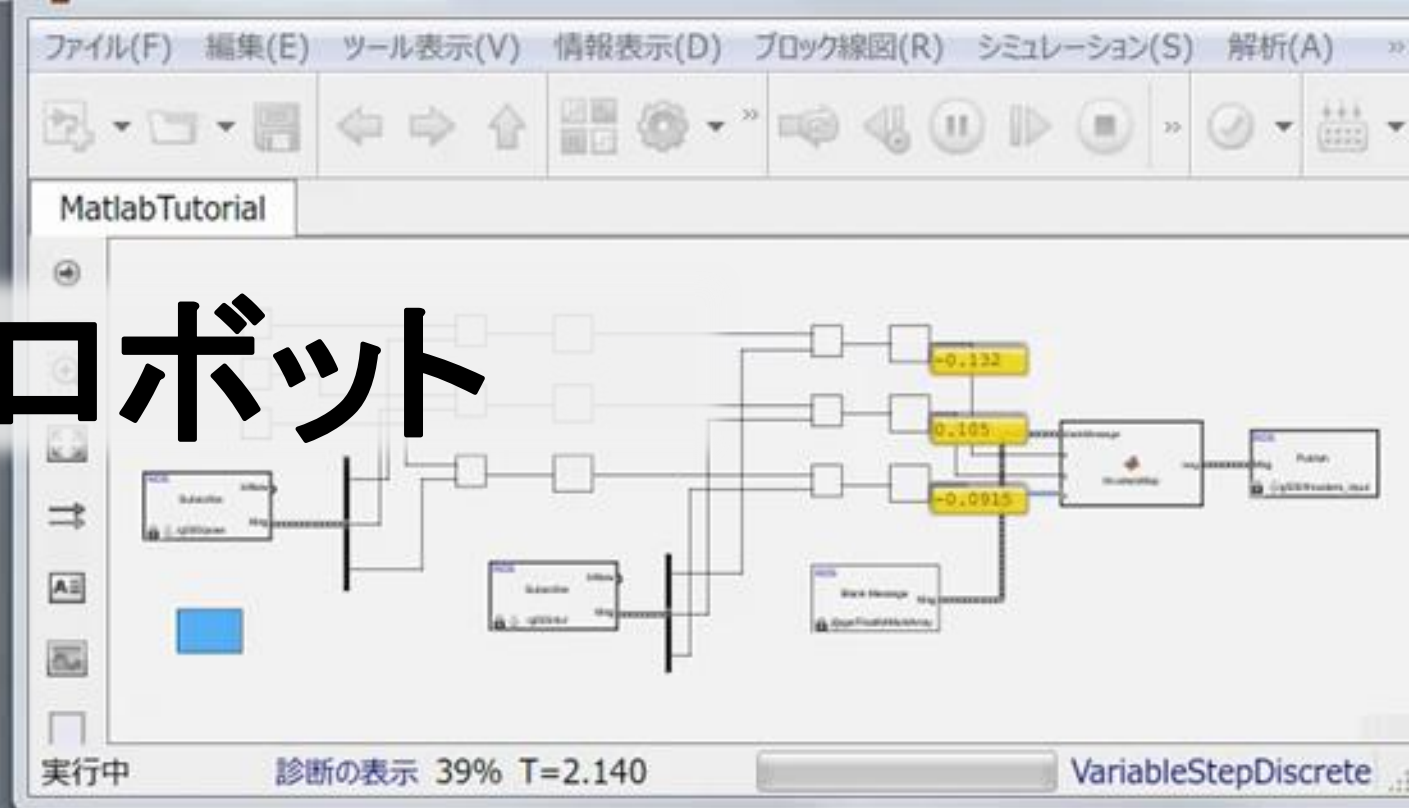


# オフロード車両の3Dシミュレーション





# マリンロボット



# KTH Royal Institute of Technology Researchers Develop Control Algorithms to Simulate AUV Hydrobatatics

## Challenge

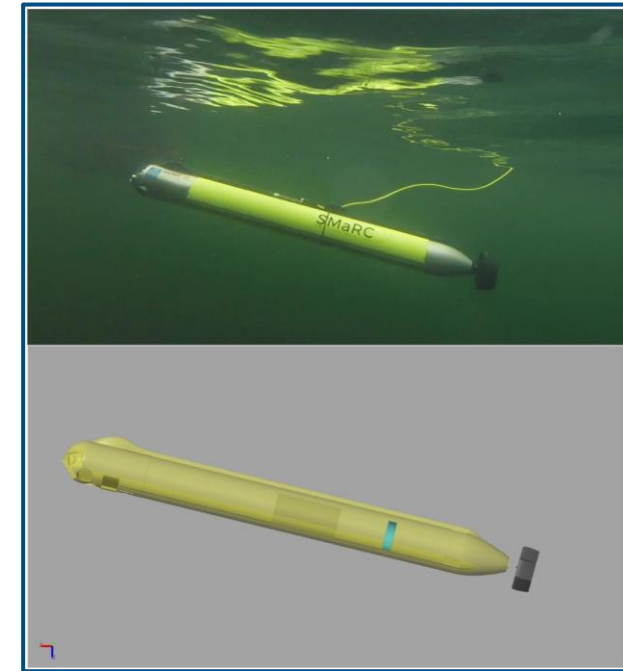
Improve the autonomous capabilities of AUVs that operate remotely and at depth

## Solution

Develop control algorithms using MATLAB and Simulink that allow researchers to simulate complex AUV maneuverability

## Results

- Developed effective control strategies that help the AUV perform complex maneuvers in real-world tests
- Developed vehicle motion simulations in Simulink that run in close-to-real-time and are sufficiently accurate to model qualitative behaviors
- Increased the pace of AUV research by rapidly simulating, optimizing, and implementing control algorithms before field-testing



Submerged hydrobatatic AUV SAM (above) and a 3D representation of the vehicle (below).

*“Finding an effective control strategy for complex maneuvers would have been extremely difficult if we were limited to solely conducting tests with the AUV itself. Via simulation, however, we were able to rapidly try various MPC designs and combinations of PID controllers, as well as generate detailed plots of the simulation results in MATLAB to see which approach worked best.”*

*- Ivan Stenius and Sriharsha Bhat, KTH Royal Institute of Technology*

# Drass Develops Deep Learning System for Real-Time Object Detection in Maritime Environments

## Challenge

Help ship operators monitor sea environments and detect objects, obstacles, and other ships

## Solution

Create an object-detection deep learning model that can be deployed on ships and run in real time

## Results

- Data labeling automated
- Development time reduced
- Flexible and reproducible framework established



First day of object detection tests with optronic system prototype.

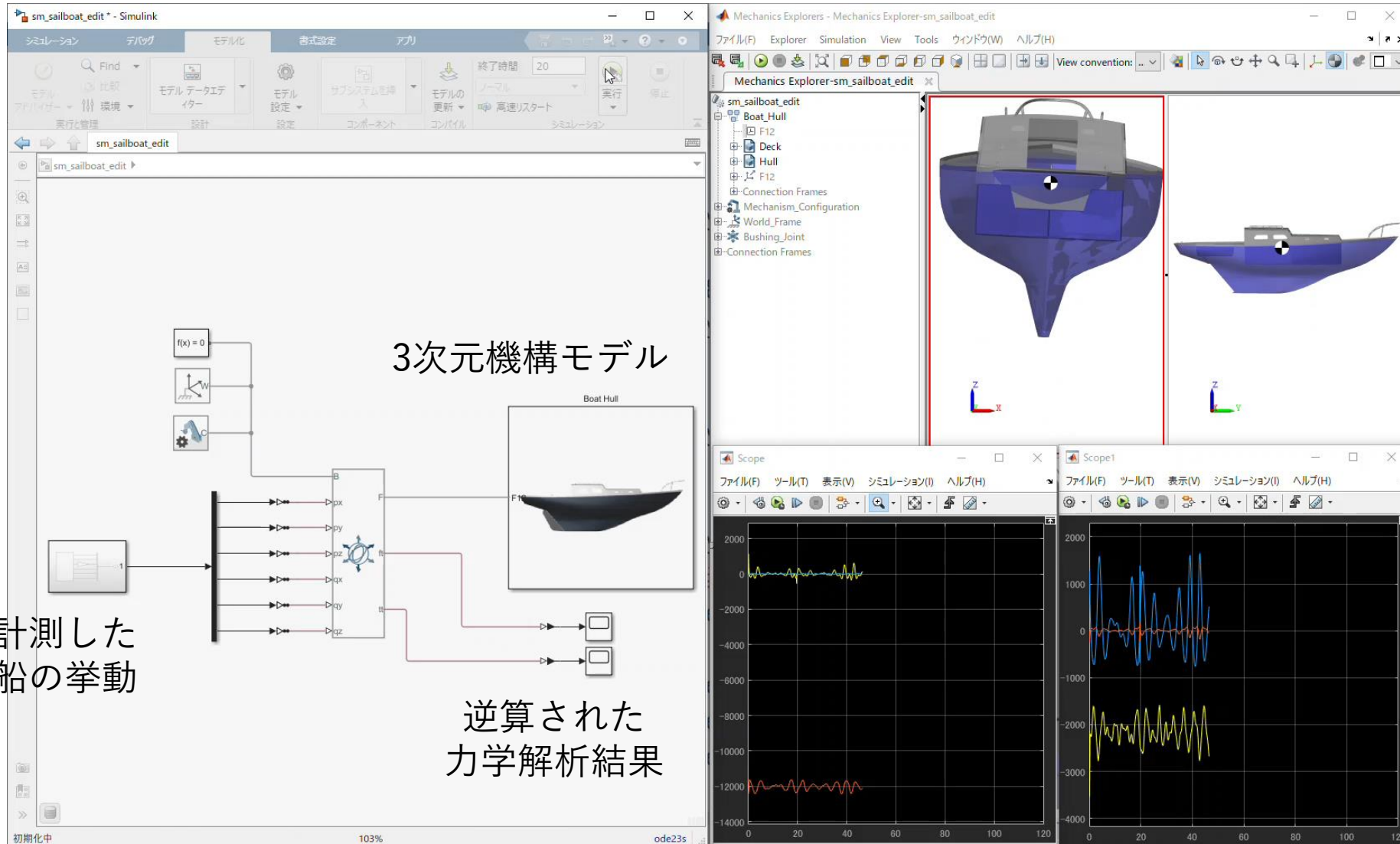
*“From data annotation to choosing, training, testing, and fine-tuning our deep learning model, MATLAB had all the tools we needed—and GPU Coder enabled us to rapidly deploy to our NVIDIA GPUs even though we had limited GPU experience.”*

*- Valerio Imbriolo, Drass Group*



# 計測データを使った波力解析

Simscape Multibody



CAD取り込み  
3次元的な  
イナーシャの再現

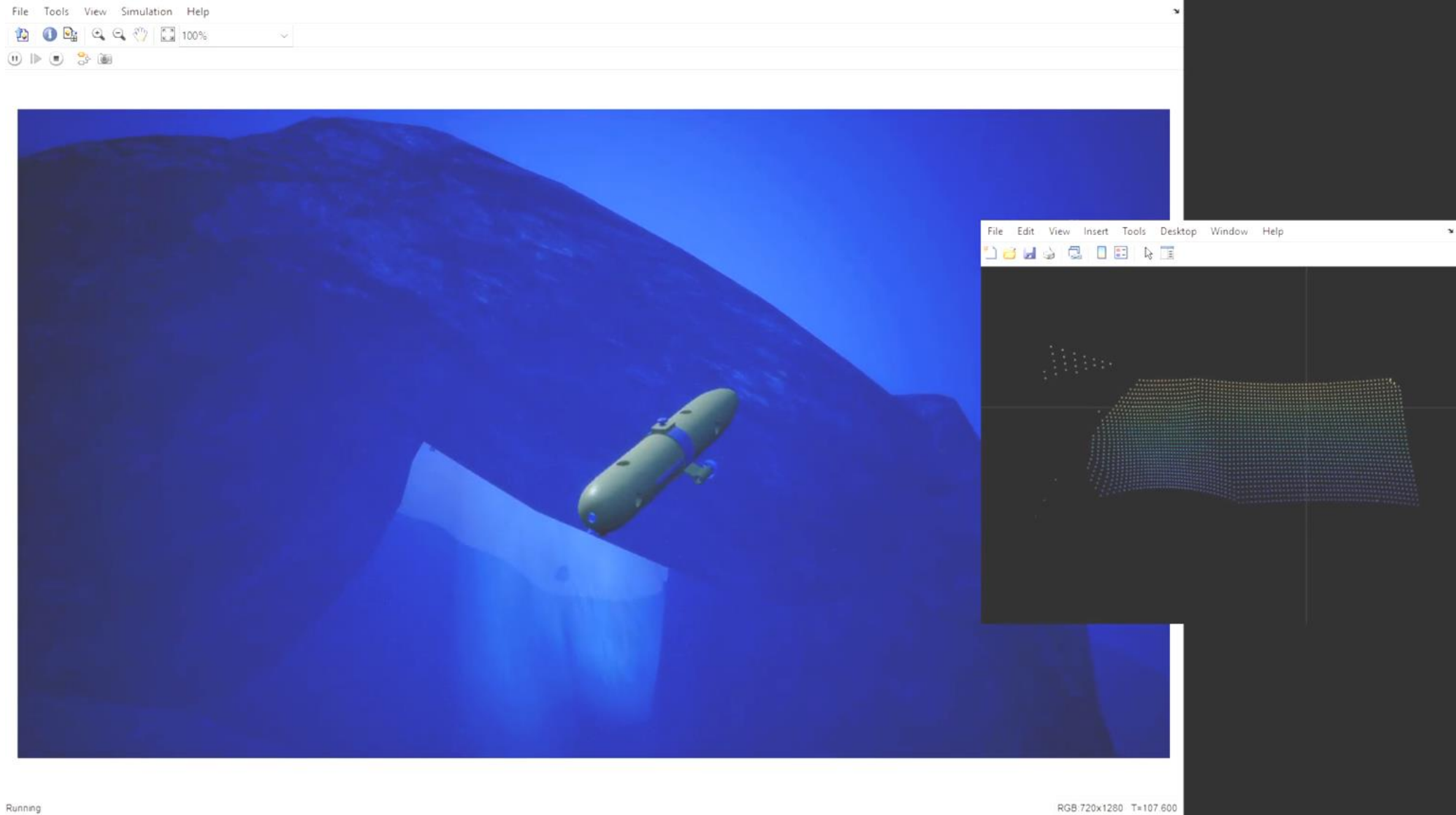


IMU等計測データ  
から逆動力学解析  
⇒波から受ける力  
及びトルクを逆算

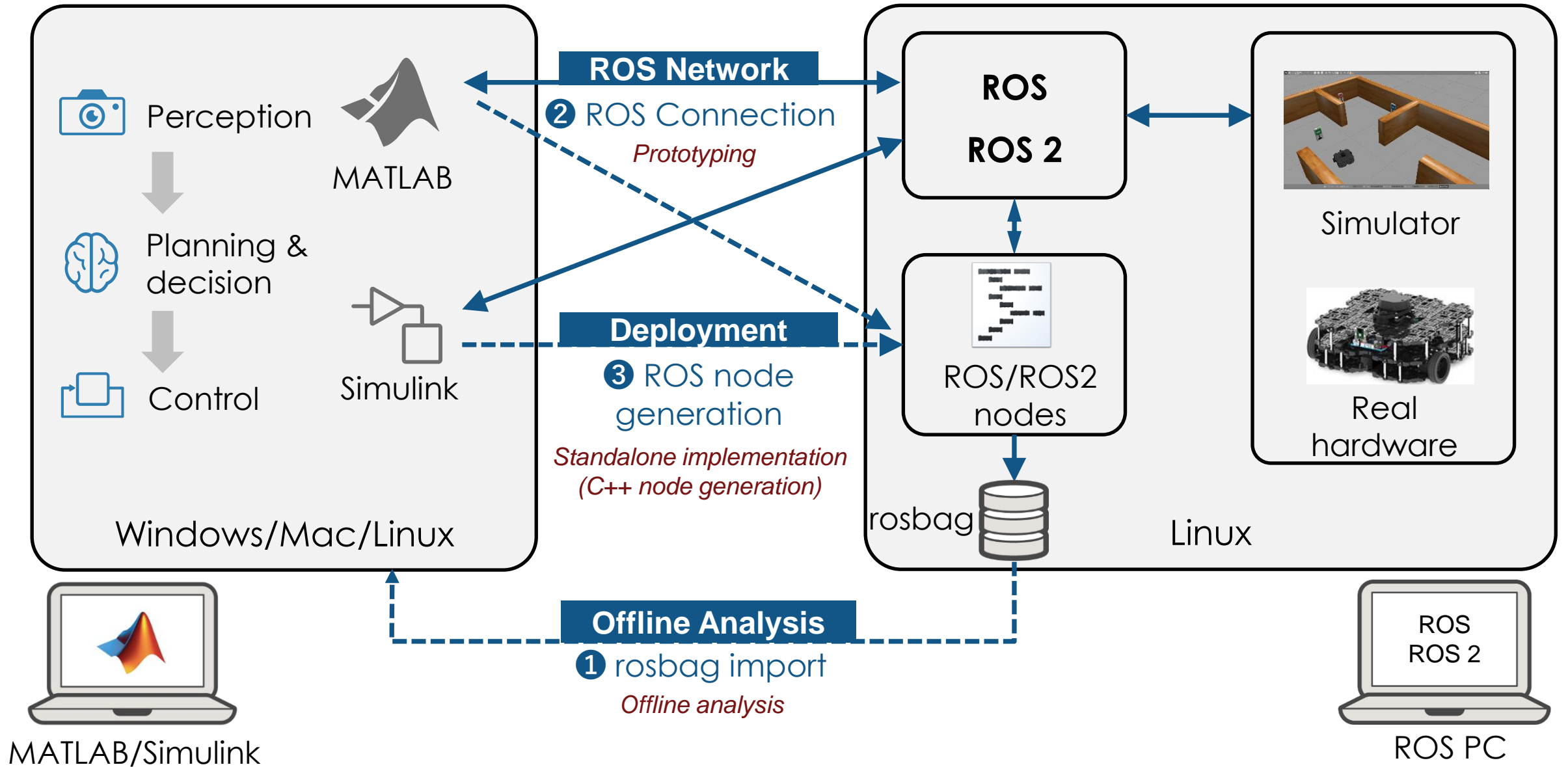


重心位置の設計や  
波のモデリングに活用

# マリンロボットの3Dシミュレーション



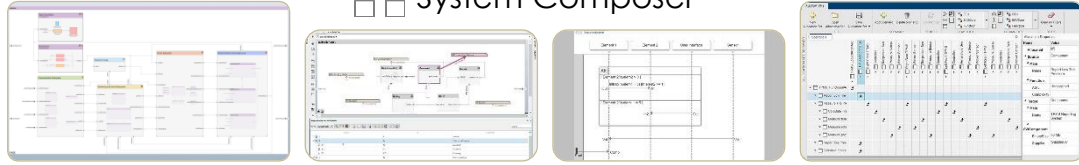
# Bridging ROS / ROS 2 with MATLAB and Simulink





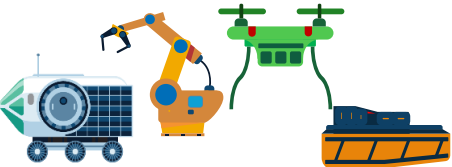
# 自律ロボティクス開発ワークフロー

**MATLAB® Simulink®**  
 システムエンジニアリング  
 System Composer™

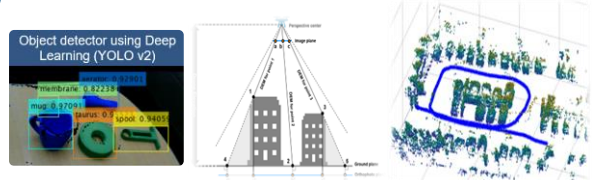


プラットフォーム設計

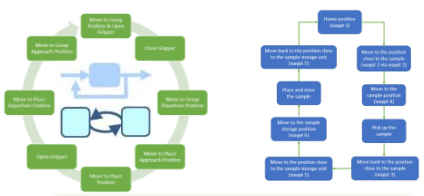
AI・自律系設計



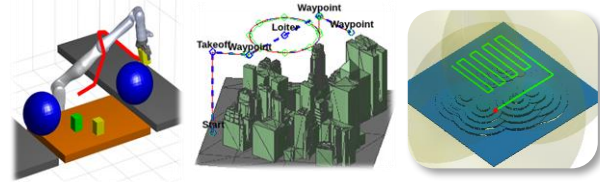
ロボットモデル



認識・自己位置推定

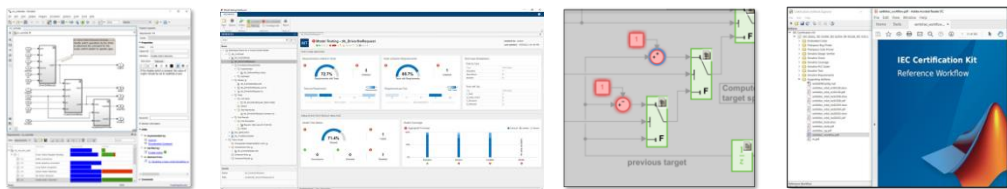


制御



モーションプランニング

検証と妥当性確認



連成

3Dシミュレーション



シーン・環境

オブジェクト・アクター

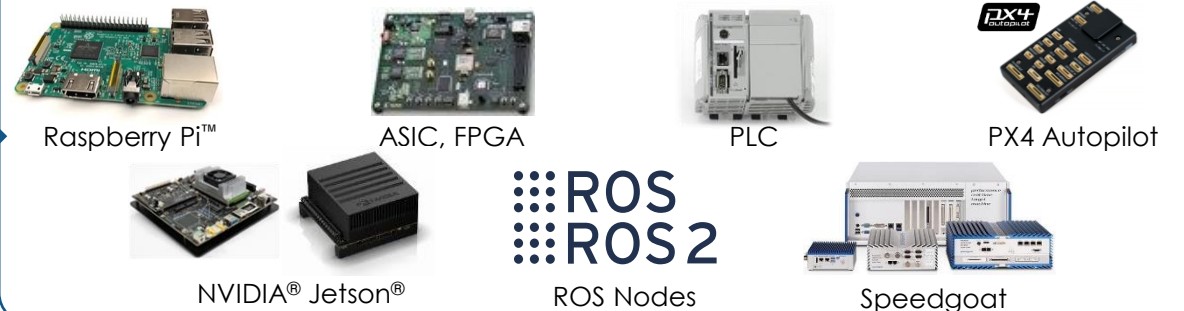
テスト

ロボットとの接続



実装

ハードウェアへの実装



# 自律ロボティクス開発ワークフロー



## システムの定義

### System Composer™



- システム要求仕様の策定・改善・管理
- システムアーキテクチャモデリング
- アーキテクチャ分析とトレードオフ分析
- インターフェース定義
- システム動作の検証

連成

## 3Dシミュレーション



シーン・環境

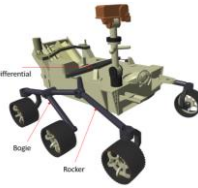
- 動作環境
- 照明条件
- 可変レイアウト
- シーンアセット作成
- 外部3Dシミュレータ (Gazebo, Unreal Engine®, Unityなど)
- ROS/ROS 2による接続



オブジェクト・アクター

- 環境要素、物体や障害物
- 動的障害物の軌道定義
- シナリオアクター (人間、他ロボット、等)

## プラットフォーム設計



- 車体ダイナミクス
- マルチボディモデリング
- 車輪接触・摩擦
- CAD/URDFのインポート
- ロット運動学モデル
- センサーモデル

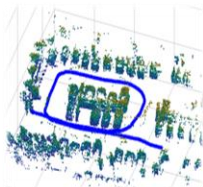
### ロボットモデル



- 調停(スーパーバイザー)設計
- モーター制御
- PID制御
- モデル予測制御
- 最適化

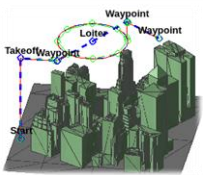
### 制御

## AI・自律系設計



- センサーフュージョン
- コンピュータビジョン/ライダー処理
- 物体・障害物の検出・トラッキング
- セグメンテーション/分類
- 物体姿勢推定
- SLAM (自己位置推定 & マップ作成)

### 認知



- 経路計画・最適化
- 障害物/衝突回避
- 経路の動的再計画
- パスメトリクス
- 強化学習

### モーションプランニング

テスト

## ロボットとの接続



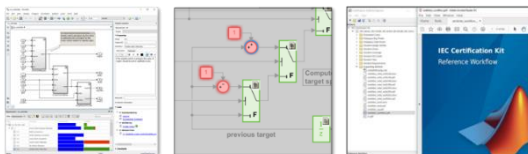
ROS  
ROS2



- ハードウェアによるアルゴリズムのテストと検証
- ロボットハードウェアの制御
- 実センサーデータの取得
- 他のモバイルロボットのROS/ROS 2経由での接続

実装

## 検証と妥当性確認



- 要求項目のトレーサビリティ
- モデル標準
- テストカバレッジ解析
- 設計エラー検出 (ランタイムエラー、デッドロジック)
- 静的コード解析
- レポート作成/証明書作成

## ハードウェアへの実装



PX4 Autopilot



Raspberry Pi™



ASIC, FPGA



NVIDIA® Jetson®



ROS Nodes



Speedgoat

- 自動コード生成 (C,C++, Verilog, VHDL, ST, CUDA, 他)
- ROS/ROS 2用C++ノードの自動生成
- プロセッサ・イン・ザ・ループ (PIL)、ハードウェア・イン・ザ・ループ (HIL) テスト
- 統合ハードウェア・テストのための実装

# 自律ロボット開発関連MathWorks製品

**MATLAB**® **Simulink**®  
システムの定義

System Composer™

Requirements Toolbox™

連成

## 3Dシミュレーション

UAV Toolbox™

RoadRunner

Automated Driving Toolbox™

RoadRunner Scene Builder

Robotics System Toolbox™

RoadRunner Scenario™

RoadRunner Asset Library

## プラットフォーム設計

Simscape™

Simscape Multibody™

Simscape Electrical™

Aerospace Blockset™

Stateflow®

Simulink Control Design™

Robust Control Toolbox™

Optimization Toolbox™

## AI・自律系設計

Computer Vision Toolbox™

Lidar Toolbox™

Deep Learning Toolbox™

Sensor Fusion and Tracking Toolbox™

Navigation Toolbox™

Robotics System Toolbox™

UAV Toolbox™

Deep Learning Toolbox™

Reinforcement Learning Toolbox™

テスト

## ロボットとの接続

ROS Toolbox

Robotics System Toolbox Support Package for Universal Robots UR Series Manipulators

Instrument Control Toolbox™

Robotics System Toolbox Support Package for KINOVA Gen3 Manipulators

Parrot® Drone Support from MATLAB

## 検証と妥当性確認

Requirements Toolbox™

Simulink Test™

Simulink Check™

Simulink Coverage™

Simulink Design Verifier™

Polyspace®

IEC Certification Kit™

DO Qualification Kit™

実装

## ハードウェアへの実装

Embedded Coder®

Simulink PLC Coder™

GPU Coder™

ROS Toolbox

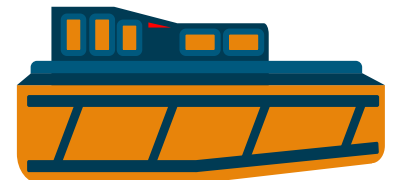
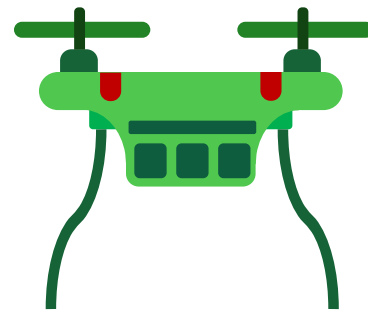
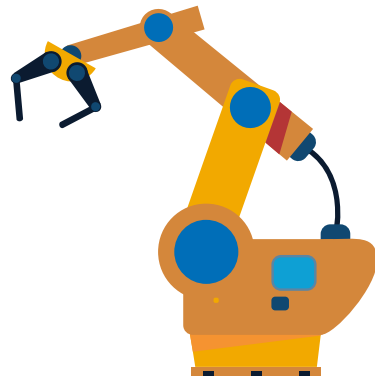
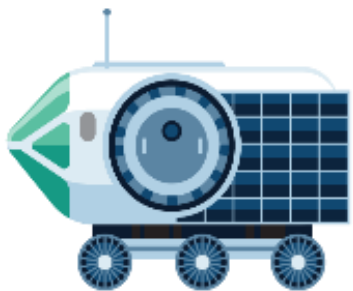
HDL Coder™

Simulink Real-Time™



## まとめ

1. MATLAB・Simulinkはロボット設計及び自律アルゴリズム開発を支援する豊富な機能・ツールを総合開発環境内で提供
2. センサーモデルを含むシステムシミュレーションを行うことにより実機試験のリスク・コスト・工数の低減可能
3. エンド・トゥ・エンドワークフローによるプロトタイピングから量産の開発・検証プロセスを効率化





Accelerating the pace of engineering and science

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