

Manipulator / Robot Arm / Cobot Development Workflow



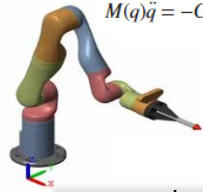
System Definition

System Composer™



- System requirements specification development/refinement/management
- System architecture modeling
- Architecture analysis and trade studies
- Interface definition
- System behavior verification

Platform Design



$$M(q)\ddot{q} = -C(q, \dot{q})\dot{q} - G(q) - J(q)^T F_{Ext} + \tau$$

- Multibody modeling
- Contact/Friction
- Rigid body trees
- Robot dynamics
- CAD/URDF import
- Robot library

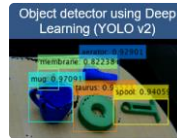
Robot Models



- Supervisory logic
- Motor controls
- Impedance control
- Torque / Force control
- PID / MPC
- Optimization

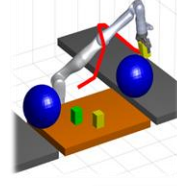
Controls

AI / Autonomy Design



- Sensor data processing / filtering
- Computer vision / Lidar processing
- Object detection
- Segmentation / Classification
- Object pose estimation
- Grasp point computation

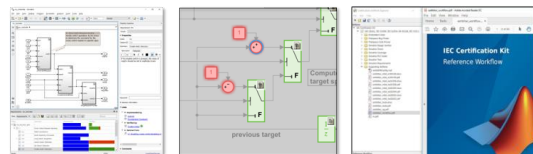
Perception



- Path/Trajectory planning
- Trajectory generation / optimization
- Obstacle/Collision avoidance
- Forward/Inverse kinematics/Dynamics
- Jacobian; Joint/Task space
- Reinforcement learning

Motion Planning

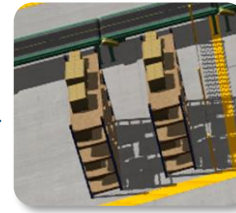
Verification & Validation



- Requirements traceability
- Model standards
- Test coverage analysis
- Design Error Detection (Run-time errors, dead logic)
- Static code analysis
- Report generation / Certification artifacts

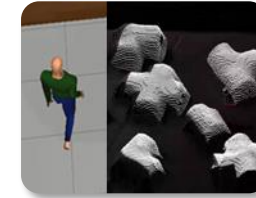
Co-sim

3D Scenario Simulation



- Operating environment
- Lighting conditions
- Variable layouts
- Scene asset creation
- External 3D simulator (Gazebo, Unreal Engine®, Unity, etc.)
- Connecting through ROS/ROS2

Scenes



- Objects and work pieces
- Randomizing object positions
- Scenario actors (e.g. humans, other robots)

Objects/Actors

Test

Connect to Robot



ROS 2



- Test and validate algorithms with hardware
- Control robot hardware
- Acquire real sensor data
- Universal Robots, Kinova support (API available through support packages)
- Connecting through ROS/ROS2 for other industrial manipulators

Build

Implement to Hardware



Raspberry Pi™



ASIC, FPGA



PLC



NVIDIA® Jetson®



ROS Nodes



Speedgoat

- Automatic Code Generation (C, C++, Verilog, VHDL, ST, CUDA, etc.)
- Automatic ROS/ROS2 C++ node generation
- Processor-in-the-Loop (PIL), Hardware-in-the-Loop (HiL) testing
- Implementation for integrated hardware testing

Mobile Robot / Autonomous Mobility Development Workflow



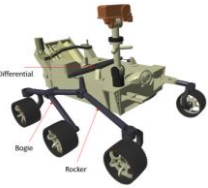
System Definition

System Composer™



- System requirements specification development/refinement/management
- System architecture modeling
- Architecture analysis and trade studies
- Interface definition
- System behavior verification

Platform Design



- Robot dynamics
- Multibody modeling
- Wheel contact / friction
- CAD/URDF import
- Kinematic models
- Sensor modeling

Robot Models



- Supervisory logic
- Motor controls
- PID controls
- Model Predictive Controls
- Optimization

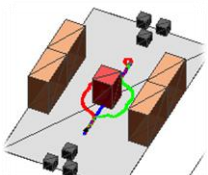
Controls

AI / Autonomy Design



- Sensor fusion
- Computer vision / Lidar processing
- Object/Obstacle detection/tracking
- Segmentation / Classification
- SLAM (Simultaneous Localization And Mapping)

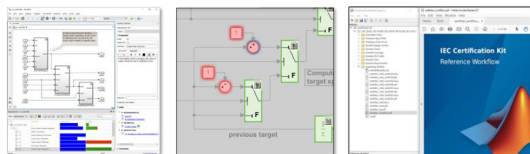
Perception



- Path planning / optimization
- Obstacle/Collision avoidance
- Dynamic replanning
- Path benchmarking
- Reinforcement learning

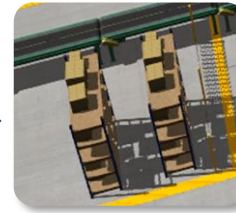
Motion Planning

Verification & Validation



- Requirements traceability
- Model standards
- Test coverage analysis
- Design Error Detection (Run-time errors, dead logic)
- Static code analysis
- Report generation / Certification artifacts

3D Scenario Simulation



- Operating environment
- Lighting conditions
- Variable layouts
- Scene asset creation
- External 3D simulator (Gazebo, Unreal Engine®, Unity, etc.)
- Connecting through ROS/ROS2



- Objects and work pieces
- Randomizing object positions
- Scenario actors (e.g. humans, other robots)

Scenes

Objects/Actors

Co-sim

Connect to Robot



ROS 2



- Test and validate algorithms with hardware
- Control robot hardware
- Acquire real sensor data
- Connecting through ROS/ROS2 for ROS-enabled mobile robots

Test

Implement to Hardware



Raspberry Pi™



ASIC, FPGA



PLC



NVIDIA® Jetson®



ROS Nodes

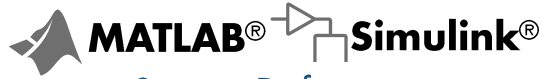


Speedgoat

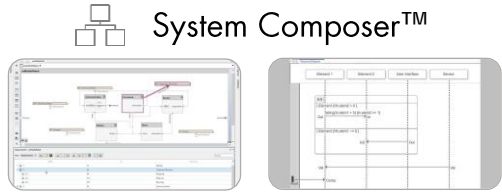
- Automatic Code Generation (C, C++, Verilog, VHDL, ST, CUDA, etc.)
- Automatic ROS/ROS2 C++ node generation
- Processor-in-the-Loop (PIL), Hardware-in-the-Loop (HIL) testing
- Implementation for integrated hardware testing

Build

UAV / Drone / Autonomous Flight Development Workflow



System Definition



- System requirements specification development/refinement/management
- System architecture modeling
- Architecture analysis and trade studies
- Interface definition
- System behavior verification

Platform Design



- UAV dynamics
- Multibody modeling
- Motor/Propeller modeling
- CAD/URDF import
- Kinematic models
- Sensor modeling

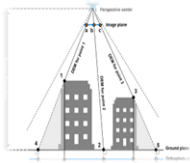
Robot Models



- Supervisory logic
- Flight modes
- Motor controls
- PID controls
- Model Predictive Controls
- Optimization

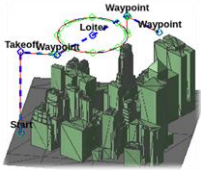
Controls

AI / Autonomy Design



- Sensor fusion
- Computer vision / Lidar processing
- Object/Obstacle detection/tracking
- Segmentation / Classification
- SLAM (Simultaneous Localization And Mapping)

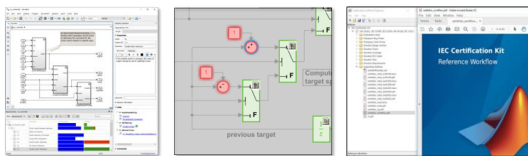
Perception



- Path planning / optimization
- Obstacle/Collision avoidance
- Dynamic replanning
- Path benchmarking
- Reinforcement learning

Motion Planning

Verification & Validation



- Requirements traceability
- Model standards
- Test coverage analysis
- Design Error Detection (Run-time errors, dead logic)
- Static code analysis
- Report generation / Certification artifacts

Co-sim

3D Scenario Simulation



- Operating environment
- Lighting conditions
- Variable layouts
- Scene asset creation
- External 3D simulator (Gazebo, Unreal Engine®, Unity, etc.)
- Connecting through ROS/ROS2

Scenes



- Environmental objects, obstacles
- Defining obstacle trajectories
- Scenario actors (e.g., humans, other UAV)

Objects/Actors

Test

Connect to Robot



- Test and validate algorithms with hardware
- Connect to and control UAV hardware
- Acquire real sensor data
- Connecting through ROS/ROS2 for ROS-enabled UAV

Build

Implement to Hardware



PX4 Autopilot



Raspberry Pi™



ASIC, FPGA



NVIDIA® Jetson®



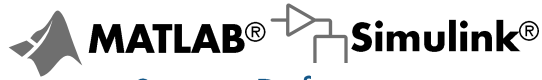
ROS Nodes



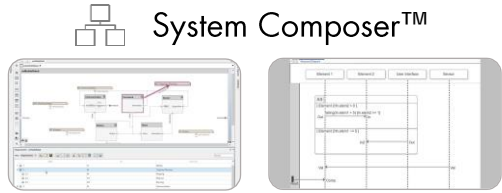
Speedgoat

- Automatic Code Generation (C, C++, Verilog, VHDL, ST, CUDA, etc.)
- Generate code and implement directly to PX4 autopilots
- Automatic ROS/ROS2 C++ node generation
- Processor-in-the-Loop (PIL), Hardware-in-the-Loop (HIL) testing
- Implementation for integrated hardware testing

Autonomous Underwater/Surface Vehicle Development Workflow

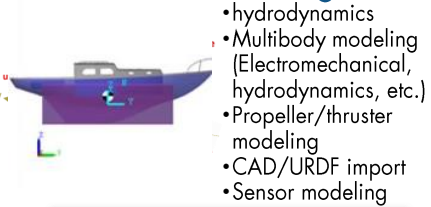


System Definition



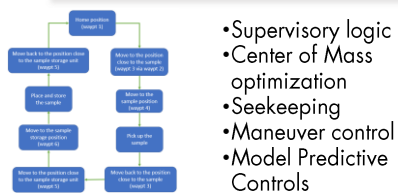
- System requirements specification development/refinement/management
- System architecture modeling
- Architecture analysis and trade studies
- Interface definition
- System behavior verification

Platform Design



- hydrodynamics
- Multibody modeling (Electromechanical, hydrodynamics, etc.)
- Propeller/thruster modeling
- CAD/URDF import
- Sensor modeling

Robot Models



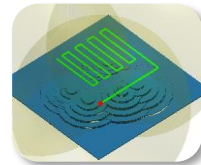
Controls

AI / Autonomy Design



- Sensor fusion with radar/sonar
- Computer vision / Lidar processing
- Object/Obstacle detection/tracking
- Segmentation / Classification
- SLAM (Simultaneous Localization And Mapping)

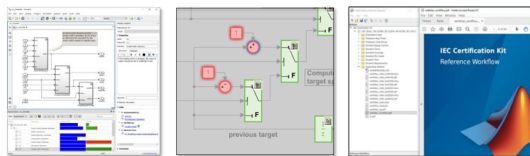
Perception



- Path planning / optimization
- Obstacle/Collision avoidance
- Dynamic replanning
- Path benchmarking
- Reinforcement learning

Motion Planning

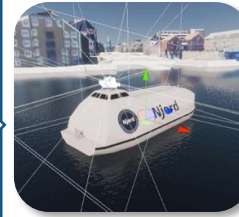
Verification & Validation



- Requirements traceability
- Model standards
- Test coverage analysis
- Design Error Detection (Run-time errors, dead logic)
- Static code analysis
- Report generation / Certification artifacts

Co-sim

3D Scenario Simulation



- Operating environment
- Lighting conditions
- Variable layouts
- Scene asset creation
- External 3D simulator (Gazebo, Unreal Engine®, Unity, etc.)
- Connecting through ROS/ROS2

Scenes



- Environmental obstacles
- Defining obstacle trajectories
- Scenario actors (e.g., humans, other AUV, ASV, etc.)

Objects/Actors

Test

Connect to Robot



- Test and validate algorithms with hardware
- Control robot hardware
- Acquire real sensor data
- Connecting through ROS/ROS2 for ROS-enabled robots

Build

Implement to Hardware



Raspberry Pi™



ASIC, FPGA



PX4 Autopilot



NVIDIA® Jetson®



ROS Nodes



Speedgoat

- Automatic Code Generation (C, C++, Verilog, VHDL, ST, CUDA, etc.)
- Generate code and implement directly to PX4 autopilots
- Automatic ROS/ROS2 C++ node generation
- Processor-in-the-Loop (PIL), Hardware-in-the-Loop (HIL) testing
- Implementation for integrated hardware testing