Entwicklung und Testen von Robotischen Anwendungen mit MATLAB® und Simulink®

Maximilian Apfelbeck, MathWorks

MATLAB EXPO 2015
DEUTSCHLAND
Robot Teleoperation

Control Device  
*ROS-Node*

Turtlebot  
*ROS-Master*

Controller  
*ROS-Node*

IMU  
V, W
3 Key Take Aways

- Fast algorithm design & testing
- Team collaboration
- Connectivity to Robots
Robot Architecture

- **Sensors**
- **Actuation**
- **Intelligence / Algorithms**
- **Infrastructure**

**Components:**
- Global Planner
- Local Planner
- Motor Controller
- Camera
- LIDAR
- Joystick
- Map server
- Image processing
- Localization & Mapping
- Kinematics
- Drv
- Enc
- WiFi
- Remote machine

**Software:**
- MATLAB
- Simulink
Challenge
Develop control systems for a two-armed mobile humanoid robot with 53 degrees of freedom

Solution
Use Model-Based Design with MATLAB and Simulink to model the controllers and plant, generate code for HIL testing and real-time operation, optimize trajectories, and automate sensor calibration

Results
- Programming defects eliminated
- Complex functionality implemented in hours
- Advanced control development by students enabled

“Model-Based Design and automatic code generation enable us to cope with the complexity of Agile Justin’s 53 degrees of freedom. Without Model-Based Design it would have been impossible to build the controllers for such a complex robotic system with hard real-time performance.”

Berthold Bäuml
DLR
Model-Based Design
Continuous Verification and Validation

Requirements

System Design
- Environment
- Physical Components
- Algorithms

System-Level Specification

Component Design

Subsystem Design

Research
- Data Analysis
- Algorithm Development
- Data Modeling

Algorithm Development

Implementation
- Embedded Software
  - C, C++
- Digital Electronics
  - VHDL, Verilog
- MCU, DSP, FPGA, ASIC

Code Verification and Validation

Subsystem Implementation

Integration testing

Subsystem Integration & Test

User Acceptance Testing

Complete Integration & Test

MATLAB EXPO 2015 DEUTSCHLAND
Model-Based Design
Multi-Domain Modeling & Algorithm Development

- Algorithms
  - Programming Language
  - Data Flow (Block diagram)
  - Event Driven (State Machine)

- Physics
  - Physical Modeling

- Environment
  - Simulink 3D Animation

- Research
  - Data Analysis
  - Algorithm Development
  - Data Modeling

- Implementation
  - Integrate existing legacy code

- System-Level Specification
  - Requirements
  - System Design
  - Physical Components
  - Environment
  - Algorithms
  - Component Design
  - Data Modeling
  - Algorithm Development
  - Data Analysis

- DSP
- FPGA
- ASIC
- Embedded Software
- Digital Electronics
- C, C++
- VHDL, Verilog

- MCU

- Integrates existing legacy code
Without Robotics System Toolbox

Deliver a Robotics Application

Challenges:
• Easily access of sensor data on robots
• Multi-platform support
• Simulation of environment (not the robot itself)
• Ready-to-use algorithms

Robotics System Toolbox

Powerful but not easy
Main Functionality

1. Access ROS capabilities from MATLAB (I/O)
2. Access ROS capabilities from Simulink (I/O and C++ code generation)
3. Algorithms for autonomous wheeled robots
4. Application Examples for working with TurtleBot and Gazebo (robot simulator)
Speed-up Algorithm Design with System Toolboxes

- Video & Image Processing
  - Image Analysis
  - Video Analysis (motion tracking)
  - Image Enhancements
  - Scene Reconstruction

- Signal Processing
  - Algorithm Design
  - Time and frequency domain
  - Frame-based processing
  - Digital filters (FIR, IIR, multirate, adaptive)
Model-Based Design
Continuous Verification and Validation

System Design
- Environment
- Physical Components
- Algorithms

Component Design

Research
- Data Analysis
- Algorithm Development
- Data Modeling

Implementation
- Embedded Software
  - C, C++
- Digital Electronics
  - VHDL, Verilog
  - MCU, DSP, FPGA, ASIC

Verification and Validation

Integration testing
- Complete Integration & Test

Code Verification and Validation

Subsystem Integration & Test

Subsystem Design

User Acceptance Testing

Complete System-Level Integration & Test

Requirements

System-Level Specification

Data Modeling

Algorithm Development
Model-Based Design
Generate ROS C++ node

- C/C++, VHDL and PLC-Code Generation from one model

- Support for Fixed Point Data Format
  - Automatic scaling
  - Supported in Simulation and Code-Generation

- Easy integration of legacy C/C++-Code

- Generate a ROS C++ node from a Simulink model and deploy it on a robot
3 Key Take Aways

- Fast algorithm design & testing
- Team collaboration
- Connectivity to Robots