A toolchain for Rapid Control Prototyping using Rexroth PLCs and open source software

MODPROD Workshop 2014
Linköping University

Nils Menager
Lars Mikelsons

04.02.2014
Outline

Motivation & Introduction
• Introduction to Rapid Control Prototyping
• Rapid Control Prototyping in the automotive industry
• Requirements and specifications

Implementation
• Structure of the toolchain
• Modelica and the C++ code generation
• Compilation of the C++ code using the VxWorks compiler
• Designing the function block as interface in IndraWorks

Application
• Application on real-life industrial system using HiL

Summary & Outlook
• Summary of the results
• Outlook on further investigations
Motivation & Introduction

Introduction to Rapid Control Prototyping

- First step during development of a new system is to build up a simulation model

- For the validation of the simulation model, a **controller model** is necessary
Introduction to Rapid Control Prototyping

- During the commissioning process, the control algorithm is usually developed and implemented completely new.

- Rapid Control Prototyping (RCP) is a method to easily and quickly develop and test control strategies on hardware targets.
Introduction to Rapid Control Prototyping

- Rapid Control Prototyping includes
  - the (dynamical) description of the system
  - model-based controller design inside a simulation environment
  - code generation and execution on a hardware target
  - testing of the controller structure inside the simulation environment (HiL) and on the real system

- Until now, Rapid Control Prototyping and model based controller development mainly used in the automotive industry
Motivation & Introduction

Rapid Control Prototyping in the automotive industry

- One common toolchain in the automotive industry uses Matlab/Simulink

- Matlab/Simulink has 4 main disadvantages for usage in automation industry
  - with every new release of Matlab/Simulink all models have to be tested again due to possible changes on the code generation
  - no possibilities to modify the code generation module
  - Matlab/Simulink not suitable for modeling of system (two different tools necessary for controller and system modeling)
  - standard customer has often no possibility to buy Matlab/Simulink due to high costs
Motivation & Introduction

Requirements and specifications for the toolchain

Specifications for the Rapid Control Prototyping toolchain:
- based on open source software due to costs
- Modelica models as source
- based on in-house developed code generation to avoid dependencies
- applicable to standard Rexroth controllers without modifications
- automated, easy-to-use toolchain for commissioning engineers

Source: knowhowforum.com
Implementation

Motivation & Introduction
- Introduction to Rapid Control Prototyping
- Rapid Control Prototyping in the automotive industry
- Requirements and specifications

Implementation
- Structure of the toolchain
- Modelica and the C++ code generation
- Compilation of the C++ code using the VxWorks compiler
- Designing the function block as interface in IndraWorks

Application
- Application on real-life industrial system using HiL

Summary & Outlook
- Summary of the results
- Outlook on further investigations
Implementation

Structure of the toolchain

Modelica model -> .mo file -> Code generation -> VxWorks Compiler -> IndraWorks program -> MLC

Simulation runtime

C++ code
Modelica and the C++ code generation

- Starting point of the toolchain is a Modelica model
- Commercial (e.g. Dymola) as well as open source (e.g. OpenModelica, JModelica) compilers available
- Here, the open source OpenModelica compiler is used
- C++ code generation module inside the OpenModelica compiler developed at Bosch Rexroth
- To execute the code, a simulation runtime is necessary
- Here, the Rexroth in-house C++ simulation runtime is used
Compilation of the C++ code using the VxWorks compiler

- Real-time operating system on Rexroth controller is VxWorks
- Compilation of the C++ code (model and simulation runtime) for VxWorks
- Usage of Windriver VxWorks compiler
- Result is a set of libraries
- Transfer of the libraries onto the Rexroth controller flash-disk partition using an FTP client
Designing the function block as interface in IndraWorks

- Function block inside IndraWorks as interface between the PLC program and the C++ code
- Function block has inputs and outputs as well as internal variables
- Workflow inside IndraWorks remains unchanged
- Only difference: Implementation part of the function block is empty (external linking of the implementation)
- Each (different) function block can have its own C++ implementation
- Combination of IEC function blocks and C++ function blocks possible
- All task settings or diagnosis methods can be used unrestrictedly
Application

Motivation & Introduction
- Introduction to Rapid Control Prototyping
- Rapid Control Prototyping in the automotive industry
- Requirements and specifications

Implementation
- Structure of the toolchain
- Modelica and the C++ code generation
- Compilation of the C++ code using the VxWorks compiler
- Designing the function block as interface in IndraWorks

Application
- Application on real-life industrial system using HiL

Summary & Outlook
- Summary of the results
- Outlook on further investigations
Application on real-life industrial system using HiL

Virtual system (Simster)

Controller

Real system

Validation of the controller in a Hardware-In-The-Loop setup → Video

Application on real-life industrial system using HiL
Summary & Outlook

Motivation & Introduction
- Introduction to Rapid Control Prototyping
- Rapid Control Prototyping in the automotive industry
- Requirements and specifications

Implementation
- Structure of the toolchain
- Modelica and the C++ code generation
- Compilation of the C++ code using the VxWorks compiler
- Designing the function block as interface in IndraWorks

Application
- Application on real-life industrial system using HiL

Summary & Outlook
- Summary of the results
- Outlook on further investigations
Summary & Outlook

Summary of the results

- Requirements for a toolchain for rapid control prototyping in the automation industry
- Common toolchain in the automotive industry using Matlab/Simulink not applicable to problems in automation industry
- Setup of an open source toolchain based on Modelica
- Validation of the controller in a HiL-setup
Outlook on further investigations

- Validation of the controller on a real system
- Realization of a real-time HiL-simulation
- Application of alternative control strategies like Model Predictive Control
A toolchain for Rapid Control Prototyping using Rexroth controllers and open source software

MODPROD Workshop 2014
Linköping University

Nils Menager
Lars Mikelsons

04.02.2014