Model Based Development of a Multi-Axle Harvesting Machine

Vincent Theunynck | Karel Viaene
Design Engineers Model Based Control
VINTECC
28/06/2016
Key Takeaways

1. Software development without hardware availability through model-based design

2. Models allow for hardware independent application software

3. Leverage this workflow with rapid prototyping techniques for new feature development
We provide

- Model-based Software Solutions
  - From concept phase to production release
- Model-based Project Support
  - Know-how and expertise in sensors and control
- Training/Consultancy

Contact details

Contact: Vincent Theunynck / Karel Viaene
Office: Leenstraat 15, Izegem, BELGIUM
Tel: +32 (0) 495 216 227
Email: info@vintecc.com
**TRENDS**

**Our Competencies**

- **Faster Time-to-market**
  - ROI (+More resources for innovation)
  - Early mover advantage

- **Modeling and Identification**
  - Black/white box modeling
  - Design verification

**System Safety / Quality**

- IEC 61131-3 compliant code
- Requirements tracing & verification

**Complexity / Performance**

- Closer interaction of multiple eng disciplines
- More automation = more sensors & software

**Software Control Systems**

- Control System development
- Auto Code Generation

**Rapid Prototyping**

- dSPACE, Speedgoat
- Industrial controllers: Beckhoff, B&R, …

**Smart Sensors**

- Vision
- Soft Sensors
Use Case

Model Based Development of a Multi-Axle harvesting machine
Presentation Overview

1. Problem Introduction

2. System Overview

3. Application Software Development

4. Moving Forward
1. **Problem Introduction**

YOUTUBE VIDEO: https://www.youtube.com/watch?v=iN4LHLpJwvM
MBD OF A MULTI-AXLE HARVESTING MACHINE

Specifications

- 100 ton max
- CAT 780 Hp
- Sugar Beet Collector / Fertiliser Spreading
- Several 100k EUR
Objectives

- Design wiring scheme
- Select all electronics
- Develop full machine software

Challenges

- Starting from a clean sheet of paper
- No hardware available during the development process

Resources

- 6 man-months
Model-Based Design allows parallel development of all three domains, and therefore reduces development time, and allows early testing of your machine,
2. SYSTEM OVERVIEW
**SYSTEM OVERVIEW: SOFTWARE SUBSYSTEMS**

**Collector:**
- Collects and cleans the sugar beet
- Height (pressure) control:
  - Manual
  - Automatic based on skid plates
- Speed control:
  - 5 speed CL controlled discs

**Engine:**
- CAT C18
- Acceleration control:
  - Pedal accelerator
  - Speed Lock Mode
  - Speed Limit Mode

**Elevator belt:**
- Transports beats into the bin
- CL speed controlled

**Spirals:**
- Moves beats to the back
- CL speed controlled

**Axis alignment:**
- Realign axes:
  - Manual
  - Auto alignment

**Chain belt:**
- Unloads the bin
- CL speed controlled

**Bin:**
- Frame lift, Full bin lift (for truck unload)
- Filling detection

**Spreaders:**
- Spreads residue
- Manual control:
  - Height, length, skidder

**Driveline:**
- Forced based handle with selectable acceleration mode
- Auto functions:
  - Autospeed
  - Autoreverse
SYSTEM OVERVIEW: ELECTRONIC DESIGN

Control modules
- DBCM1:
  - IFM CR0232 Infineon 32bit Tricore
  - 32 inputs, 48 outputs
- DBCM 2:
  - IFM CR2532 Freescale powerPC
  - 32 Inputs, 32 outputs

Engine ECU:
- Caterpillar C18 780bhp

Multi-function handle:
- Base:
  - Sauer Danfoss JS6000
  - CAN J1939
- Functions:
  - Proprietary design w/ up to 80 cmds
  - CAN J1939

Display:
- Maximatecc Pilot XS 10” Touchscreen
- Linux ARM processor
  - Front end (graphics): Qt
  - Back end (CAN): Codesys 3.5

Back control module
- DBCM 3:
  - IFM CR0232 Infineon 32bit Tricore
  - 32 inputs, 48 outputs
3. APPLICATION SOFTWARE DEVELOPMENT
APP SOFTWARE DEVELOPMENT: DISPLAY

Graphics
- User cmds
- Status info, Diagnosis, etc

Interface
- CAN Settings protocol
- CAN Sensor/Cmd info

Data Engine Server

Low level software
- EEPROM
- ADC/DAC
- CAN drivers

Microcontroller

OS (Linux)

Integration

Build & Deploy
APP SOFTWARE DEVELOPMENT: CONTROLLERS

Component development in Matlab/Simulink

- Collector
- Driveline
- ....

Application layer (Codesys v2.3 IDE)

Low level software

- EEPROM
- ADC/DAC
- CAN drivers
- PWM control

Microcontroller

OS

Req & Spec doc

Funcationality

Automatic Code Generation

Integration

Build & Deploy

(DBCM1, DBCM2, DBCM3)
**APP SOFTWARE DEVELOPMENT: DRIVELINE COMPONENT**

- **PLANT Models**
  - Model-in-the-Loop testing

- **CONTROL Models**
  - Software-in-the-Loop testing

- **Functionality**
  - Automatic Code Generation with Simulink PLC Coder

- **PLC Coder**
  - Integrate into embedded hardware
**APP SOFTWARE DEVELOPMENT: DRIVELINE PLANT MODEL**

- Built from component datasheets
- Tuning parameters to account for inaccuracies
**APP SOFTWARE DEVELOPMENT: DRIVELINE COMPONENT**

**PLANT Models**
- Model-in-the-Loop testing

**CONTROL Models**
- Automatic Code Generation with Simulink PLC Coder

---

**Functionality**
- PLC Coder

**Integrate into embedded hardware**
APP SOFTWARE DEVELOPMENT: PROCESSOR-IN-THE-LOOP TESTING

- Process in-the-Loop testing
- PLC Code
  - Integrate into embedded hardware
  - CAN
  - Integration
  - Parameter Tuning
  - Monitoring & Tuning

Field Deployment & Tuning
- Fully validated
- Field deployment
APP SOFTWARE DEVELOPMENT: PROCESSOR-IN-THE-LOOP TESTING

- CAN J1939
- DBCM1
- DBCM2
- DBCM3
- CAN interface
- VNT

PIL Test
- All I/O on TX CAN (temporarily)
- PC is simulating Plant Models
- ‘Soft’ real-time

- Pilot XS Display
- Joysticks

Vector .dbc file

APP SOFTWARE DEVELOPMENT: PROCESSOR-IN-THE-LOOP TESTING

- CAN J1939
- DBCM1
- DBCM2
- DBCM3
- CAN interface
- VNT

PIL Test
- All I/O on TX CAN (temporarily)
- PC is simulating Plant Models
- ‘Soft’ real-time

- Pilot XS Display
- Joysticks

Vector .dbc file
APP SOFTWARE DEVELOPMENT: FIELD DEPLOYMENT

- Processor-in-the-Loop testing
- Field Deployment & Tuning
- Fully validated

PLC Code
Integrate into embedded hardware
CAN
Integration
Parameter Tuning

Field deployment
Monitoring & Tuning
Monitoring & Tuning

CoDeSys

MathWorks
SUMMARY KEY TAKEAWAYS

1. Software development without hardware availability through model-based design:
   - 90% of the design verified before final field deployment
   - Development time shortened by months

2. Models allow for hardware independent application software

3. Leverage this workflow with rapid prototyping techniques for new feature development
4. MOVING FORWARD
Controlled Loading:
- Uses a sugar beat collector
- Uses a simple hatch to unload

Controlled Unloading:
- Uses a fertiliser collector
- Uses vertical rotors and horizontal discs

Sugar Beat Collector control

Sugar Beat Bin unloading

Fertiliser Collector control

Fertiliser Bin unloading

[Shared Functionality]

Driveline

Axes Alignment
FERTILISER CONTROL DEVELOPMENT

Objectives

- Develop functionality so that fertiliser operation is supported
- Starting from our initial collector/unloading Simulink control models

Challenges

- Hard to make a plant model from fertiliser spreading behaviour…
- In-field control system development is required

Solution

- Use a flexible environment to update/tune/re-iterate our control system (RAPID PROTOTYPING)
How can we develop the fertiliser controls without hampering field operation?

Needs to be upgraded for fertiliser control
Rapid Prototyping

- Fastly iterating alternative/improved algorithms
- Using existing wiring / sensors / actuators / …
- ... or extend with additional I/O if required
RAPID PROTOTYPING

Development Environment

- Simulink External Mode
- Beckhoff TwinCat ADS server
- Exported Block Diagram

Simulink Coder

BECKHOFF

In-field functionality check

Sensors

Actuators

CAN
INTEGRATION OF NEW FUNCTIONALITY

Development Environment

Component 1
Control Model

CAN

PLC

PLC Encoder

Sensors

Actuators

Common Functionality

Req & Spec doc
Summary Key Takeaways

1. Software development without hardware availability through model-based design
   - 90% of the design verified before final field deployment
   - Development time shortened by months

2. Models allow for hardware independent application software
   - PLC, embedded PC or custom controller as final target
   - Rapid Pro system could have been any other Matlab/Simulink compatible system

3. Leverage this workflow with rapid prototyping techniques for new feature development
   - Models are the single source of truth and are reused throughout the development life cycle
Thank you for your attention!

Contact: Vincent Theunynck / Karel Viaene
Tel: +32 (0) 495 216 227
Email: info@vintecc.com