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

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

System Safety / Quality
- IEC 61131-3 compliant code
- Requirements tracing & verification

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

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

Software Control Systems
- Control System development
- Auto Code Generation

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

Smart Sensors
- Vision
- Soft Sensors

Market Trends

Our Competencies

Our Clients
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 Beat Collector / Fertiliser Spreading
- Several 100k EUR
PROBLEM INTRODUCTION

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 beats
- 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

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

**Driveline:**
- Forced based handle with selectable acceleration mode
- Auto functions:
  - Autospeed
  - Autoreverse

**Bin:**
- Frame lift, Full bin lift (for truck unload)
- Filling detection
**Display:**
- Maximatecc Pilot XS 10” Touchscreen
- Linux ARM processor
  - Front end (graphics): Qt
  - Back end (CAN): Codesys 3.5

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

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

**Engine ECU:**
- Caterpillar C18 780bhp

**Back control modules**
- **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)

Graphics

Interface

Data Engine Server

Low level software

Microcontroller

OS (Linux)

Graphics

Interface

Data Engine Server

Low level software

Microcontroller

Integration

Build & Deploy
APP SOFTWARE DEVELOPMENT: CONTROLLERS

Component development in Matlab/Simulink

- Collector
- Driveline
- ....

ST FBs

Application layer (Codesys v2.3 IDE)

Low level software

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

Microcontroller

OS

Functionality
Automatic Code Generation
Integration
Build & Deploy

(CBCM1, DBCM2, DBCM3)
Model-in-the-Loop testing

PLANT Models

Software-in-the-Loop testing

CONTROL Models

Automatic Code Generation with Simulink PLC Coder

Functionality

PLC Coder

Integrate into embedded hardware

SOFTWARE DEVELOPMENT: DRIVE LINE COMPONENT
APP SOFTWARE DEVELOPMENT: DRIVELINE PLANT MODEL

- Built from component datasheets
- Tuning parameters to account for inaccuracies
Model-in-the-Loop testing

PLANT Models

FUNCTIONALITY

Automatic Code Generation with Simulink PLC Coder

Software-in-the-Loop testing

CONTROL Models

PLC Coder

Integrate into embedded hardware

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

Processor-in-the-Loop testing

PLC Code
Integrate into embedded hardware

Integration

Parameter Tuning

Field Deployment & Tuning

CAN

Fully validated
Field deployment

CoDeSys
Monitoring & Tuning

Monitoring & Tuning

Monitoring & Tuning
**APP SOFTWARE DEVELOPMENT: PROCESSOR-IN-THE-LOOP TESTING**

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

---

**CAN J1939**

- **Vector .dbc file**
- **DBCM1**
- **DBCM2**
- **DBCM3**
  - CAN interface
  - VNT

**Pilot XS Display**

**Joysticks**

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

- **PIL Test**
  - All I/O on TX CAN (temporarily)
  - PC is simulating Plant Models
  - ‘Soft’ real-time
**APP SOFTWARE DEVELOPMENT: FIELD DEPLOYMENT**

1. **Processor-in-the-Loop testing**
   - Integrate into embedded hardware
   - CAN
   - Integration
   - Parameter Tuning
   - Monitoring & Tuning

2. **Field Deployment & Tuning**
   - Fully validated
   - Field deployment

---

**PLC Code**

**Integration**

**Fully validated**
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

One Machine | Two Functionalities

Sugar Beat Collector control
Sugar Beat Bin unloading

Fertiliser Collector control
Fertiliser Bin unloading

[Shared Functionality]
Driveline
Axes Alignment

[Specific Functionality]
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

CAN

Sensors

Actuators

In-field functionality check

Req & Spec doc
INTEGRATION OF NEW FUNCTIONALITY

Development Environment

Component 1
Control Model

PLC

Common Functionality

Sensors

Actuators

CAN

PLC Coder

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!