

**Dynamic Control of PMSM via Profibus Communication
Between NI-CRio and Vacon Frequency Converter**

Master's Thesis

Hammad Khan

July 16th, 2012

Supervisor:
Avo Reinap

Examiner:
At LTH: Mats Alaküla
At Physics Department: Carl Erik Magnusson
At Sibbhultsverken: Anders Göransson



Department of Measurement Technology and Industrial Engineering
Lund University

Abstract

High cost of fuel and an increased awareness to safeguard the environment has renewed the interest in Electric and Hybrid Vehicles.

To be able to have a dynamic control of the electric machine in a hybrid vehicle, a fast 1 -10ms communication speed is required between the controller and the frequency converter.

Thorough knowledge of the specific electric machine is required to be able to directly control the transistors in frequency converter via a digital link to have control over the electric machine and associated programming.

Other option is to buy a commercially available frequency converter with its own governing software and interact with the software to have the required dynamic control of the electric machine.

Second approach was selected for this thesis/project work and LabView was interfaced with the Vacon software to have a fast, dynamic control of a 3 phase permanent magnet synchronous machine PMSM.

After having established the communication via Profibus protocol, the current and speed control modes were evaluated. The speed control mode did not work as was expected due to a slow communication rate between LabView and Vacon. Where the communication rate was expected to be around 10 ms, the maximum rate observed was 180 ms, rendering the dynamic control ineffective. The communication speed was later further reduced to a maximum of 150ms.

Vacon has provided a new software and suggestions to increase the speed of communication to the desirable rate. This thesis/project work has provided the ground work to build upon the dynamic communication between LabView FPGA and Vacon.

Acknowledgements

I would like to acknowledge the help and support of the following personnel during the course of my thesis and project work at IEA, LTH.

Gatachew Darge has been very helpful in his support on the hardware front; from discussion of the design to physically implementing it. All the relay cards, amplifiers, communication cables for Profibus, installing the power supply for the buffer circuits, connectors at CRio for system interface was done by him.

Francisco Marquez is well versed in Vacon programming. He explained to me how the 'Look up' tables work, besides various aspects of the electric machine. All the programming on the Vacon side was done by him.

Pelle Steen, a representative from NI, had numerous visits to lab 6 during the course of several months to help me with various questions I had concerning implementation of the LabView program and initial communication trials via CAN and Profibus.

Contents

1	CHAPTER1 INTRODUCTION.....	9
1.1	Motivation.....	9
1.2	Background & Objectives.....	9
1.3	Outline of thesis.....	10
1.4	Contributions.....	10
2	CHAPTER 2 LABVIEW TO VACON COMMUNICATION.....	11
2.1	Compact Rio Environment.....	11
2.2	Determination of Input Output (IO) Modules.....	11
2.3	LabView Programing	13
2.4	LabVew to Vacon Interface.....	13
2.4.1	CAN Communication.....	13
2.4.2	Profibus Communication.....	14
2.5	Operating LabView.....	24
2.5.1	PC Program.....	24
2.5.2	RT program.....	25
2.6	Testing of Program in Lab 6.....	25
2.7	Vacon Software.....	25
2.7.1	Application.....	25
2.7.2	NC Load.....	26

2.7.3 NC Deff.....	26
2.7.4 NC 1131.....	26
2.7.5 NC Drive.....	26
3 CHAPTER 3 TEST SYSTEM.....	27
3.1 System Description.....	27
3.1.1 Functionality.....	28
3.1.2 Power Supply in Lab 6 (IEA).....	29
3.1.3 Power Supply at Sibbhultsverken.....	29
4 CHAPTER 4 CONTROL, MONITORING AND DATA STORAGE.....	30
4.1 System Control from LabView.....	30
4.1.1 Electric Machine Control form LabView.....	30
4.2 Monitoring Parameters.....	31
4.3 Data Storage.....	32
4.3.1 FIFO Implementation in FPGA.....	32
4.3.2 PC Program and Data Saving.....	32

5	CHAPTER 5 SAFETY SYSTEM.....	33
5.1	Safety in LabView Program.....	33
5.1.1	Electric Machine Control.....	33
5.1.2	Clutch Control – Diesel Engine to Transmission.....	33
5.1.3	Clutch Control – ExSAM to Transmission.....	33
5.1.4	Engine Enable and Throttle Control.....	33
5.1.5	Flywheel, Exhaust Brake and Engine Enable.....	34
5.1.6	STOP Command in RT Program.....	34
5.1.7	Reset Command in RT program.....	35
5.2	NC Drive parallel execution.....	36
5.3	Emergency Stop.....	36

6	CHAPTER 6 SYSTEM COMMISSIONING.....	37
6.1	Interfacing the CRio with System at Sibbhultsverken.....	37
6.2	First Experimental Run of the ExSAM w/o load.....	37
6.3	Second Run of the ExSAM with load.....	37
6.4	Observed Limitations.....	37
6.5	Search for Causes.....	38
6.6	Softening of PID Controller	38
6.7	Vacon Frequency Converter Malfunction.....	38
6.8	Fast Profibus.....	38
7	CHAPTER 7 DATA ANALYSIS AND RESULTS.....	40
7.1	Matlab for Data Analysis.....	40
7.2	Results.....	40
7.2.1	Delay Time in “Slow” Profibus Mode.....	40
7.2.2	Delay Time in “Fast” Profibus Mode.....	41
7.2.3	Delay Time in Bypass Mode – Profibus.....	42
8	CHAPTER 8 CONCLUSION AND FUTURE WORK	44
8.1	Thesis Summary.....	44
8.2	Discussion.....	44
8.3	Future Work.....	47

NOTES.....48

APPENDIX A.....49

REFERENCES.....54

Chapter 1 Introduction

1.1 Motivation

To be able to control a PMSM on a range of RPM, a frequency converter is required, which is usually built or acquired. The control mechanism is usually a direct digital link via FPGA to the IGBT. This method while very effective is limited in a sense that one cannot just buy the required power converter, which complies with the applicable EMC and other safety regulations, especially when the power requirements are high.

The company Vacon manufactures various types of converters for a variety of applications. However, these converters are generally used in an environment where static rather than dynamic control is required.

The core idea of this thesis is to establish a fast, <10ms, communication with the Vacon converter to have a dynamic control of a 160kW PMSM for hybrid vehicle application.

1.2 Background & Objectives

LTH and Sibbhultsverken, together with Volvo are developing a new type of electric drive for heavy hybrid vehicles. This new design employs a 160kW electric machine, which is mounted sideways with the transmission, called ExSAM (External Starter Alternator Motor).

Existing motor configuration in a heavy hybrid vehicle is done with an electric machine is installed in series with the engine and gear box, i.e. electric machine is sandwiched between the diesel engine and the transmission. Since the engine is mounted in the width of the vehicle, the configuration has to be narrow enough to fit, which results in smaller width and large diameter of the electric machine and corresponding fly wheel, resulting in higher inertia.

The configuration in *figure 1* below uses the motor attached to the gearbox sideways, thus resulting in saving of space in the width of the vehicle, more freedom in designing the electric machine and lower inertia due to the relatively smaller size of the flywheel.

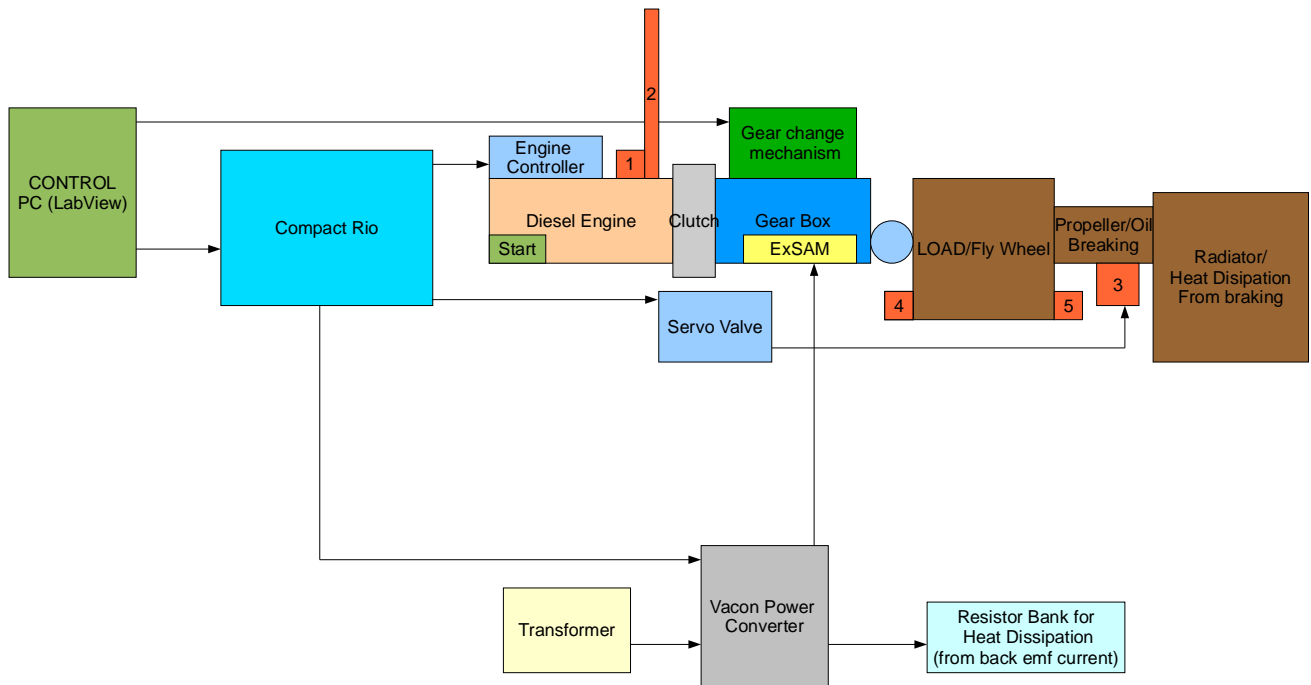


Figure 1 – System Block Diagram

A three phase AC 'Permanent Magnet Synchronous Machine' [PMSM] was designed for this purpose.

The motor design is done and the newly built motor has been installed in the test rig according to *figure 1*.

1.3 Outline of thesis

This Master's thesis is initiated to set-up the software environment for demonstrating the ExSAM in the test rig.

The objectives were divided in following three main categories:

1. Establish a fast communication between cRio LabView and Vacon to be able to have dynamic control of PMSM.
2. Control and monitoring of the Diesel Internal Combustion Engine [ICE] and the simulated load.
3. Communication and control over the sequential gear box (Section 8.3)

1.4 Contributions

Communication link between two separate systems; CRio and Vacon has been established.

Utilizing Profibus, LabView is able to communicate to Vacon Software to command and monitor the electric machine.

Chapter 2 LabView to Vacon Communication

2.1 Compact Rio Environment

National Instrument's system; Compact-Rio (CRio) with the software 'LabView' has been in use at IEA for some time now. The FPGA level program is fast enough to switch on and off transistors in a frequency converter with direct digital communication to be able to have dynamic control of an AC Machine.

Independently, the Vacon system, with its own controller, inverter and RS 232 communication link, for static control of various types of AC machines has also been used at IEA.

The main task of this thesis project is to establish a fast <10ms communication link between LabView and Vacon using the protocol 'Profibus'. This would allow the use of LabView environment to have a dynamic control of the AC Machine, utilizing Vacon's inverter and programming.

CRio is the hardware, as shown in *Figure-2* below, while the 'LabView' is the software environment, where programs can be written for specific application. There are three different levels and speeds of the programs, which are interlinked to make one coherent operating and control environment. These levels of program are described below:

1. Field Programmable Gate Array [FPGA] program: Fastest of the three programs, operates in microsecond time frame and is used to govern the operation on the electric machine.
2. Real Time [RT] program: This program is the user interface and the link between the RT and FPGA. Runs on a millisecond scale.
3. Personal Computer [PC] program: This is a link between RT program and is generally used for data recording and displaying.

2.2 Determination of Input Output (IO) Modules

Task specific modules from NI are selected to be used in the CRio; e.g. Digital Input and Output, Analog Output, Temperature Sensing, Communication, etc.

The output from these modules, for digital or analog, cannot be directly connected to the actuators due to low current levels available at the output. Tailor made buffer circuits, adapted for the output of the specific NI module and the voltage and current levels requirement at their output to drive the load circuits/actuators were designed, built and tested.

Following is a summary of the final selection of the modules for the project;

1. cRio PB – Profibus Module
2. NI 9217 – Qty 02 – Temperature Sensing
3. NI 9263 – Analog Output – 0-10VDC
4. NI 9853 – Two independent ports CAN communication
5. NI 9401 – High speed 8 channel Digital Input/Output 5V TTL
6. NI 9402 – High speed 4 channel Digital Input/Output LV TTL

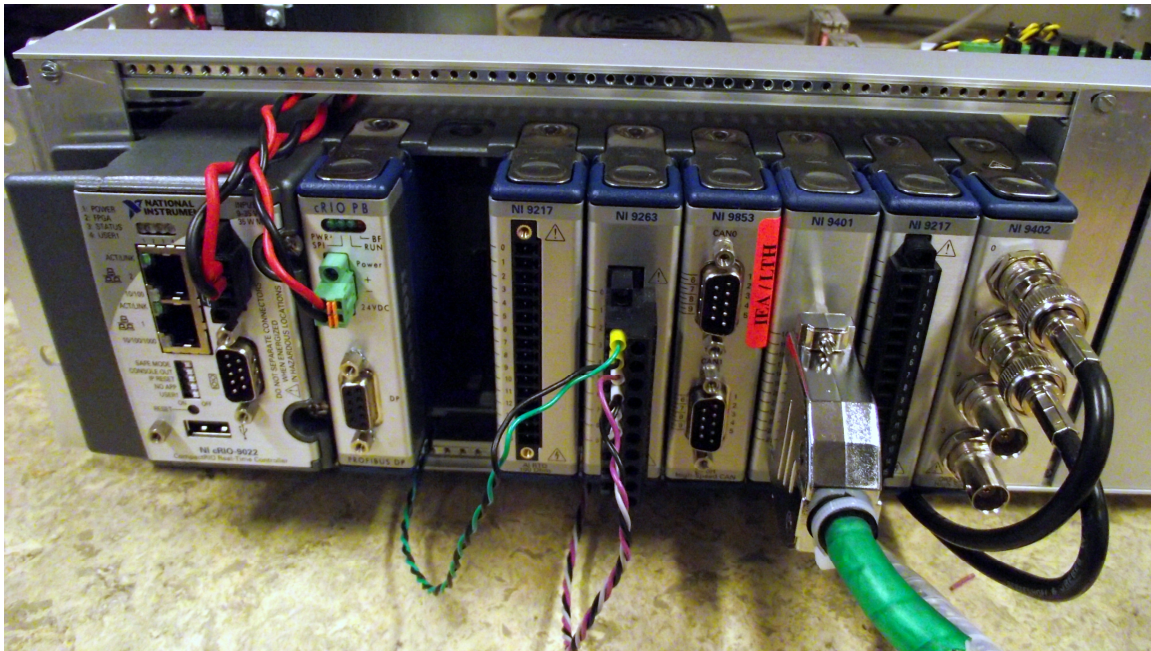


Figure 2 – Compact Rio with modules installed

2.3 LabView Programming

Unlike FORTRAN or C; which are sequential programming, LabView is a parallel programming environment. The final LabView program, has more than ten different program modules, written over a period of several months and then incorporated together to work as one unit. Following is a list of programs comprising of the final version.

1. Electric Machine Control – Current and Speed
2. Display of EM parameters
3. Display of Graphs; EM speed, current etc.
4. Analog control over the retarder brake mechanism.
5. Digital control of the clutch – Diesel Engine to Gear Box
6. Digital control of the clutch - ExSAM to Gear Box
7. Digital control of Diesel Engine and throttle.
8. Pulse detection and translating it into RPM for encoders on ICE and GB.
9. Temperature sensing.
10. Safety programs, linking the various inputs and outputs to have a smooth and fluid control over the EM, IEngine and the interfaces to remain within the operating parameters of the system.

2.4 LabView to Vacon Interface

A protocol specific module at the CRio and card at the Vacon Control Unit (CU) is required to be able to establish a communication link between the two.

2.4.1 CAN Communication

CAN protocol was studied and programs were written which established communication between the two independent ports of NI module 9853. A specialized cable with passive termination to suppress signals reflections was built for the purpose.

CAN communication uses either a cluster OR 6 U32, data types for communication. After the successful trial of establishing communication, another CAN module, NI 9862 was used, which belonged to the project.

Although this was also a CAN module, it uses a different type of technique for operation, which requires establishing a local database/memory space, which can be used by the device.

Communication from this device to Vacon control unit was attempted, which did not work. NI help was requested and one of their representative visited the lab and it was determined that cRio_LabView and vacon uses two different kinds of CAN protocol; CAN and CAN Open.

To make the CAN at NI end compatible with CAN Open at Vacon side, there exists a library of CAN open, which could be utilized by NI. However, it turned out that this particular library was not compatible with the version of cRio that we had.

Soon afterwards, it was decided to move to Profibus communication, which is 12 times as fast as CAN and was recommended by the Vacon as the Fieldbus of choice for faster, dynamic control that our project demanded.

2.4.2 Profibus Communication

NI module CRio PB, installed in the CRio was used to communicate between LabView and Vacon.

Before being able to establish the communication between the LabView and Vacon, the Profibus protocol was studied [1-8]

Profibus Module Configuration – CRio

Manual in Ref [3], was used to get started with the cRio Profibus. This manual had instruction on how to use Measurement and Automation Explorer (MAX). First configuration of the CRio system was done using MAX. Figures 3 and 4 depicts the MAX configuration windows.

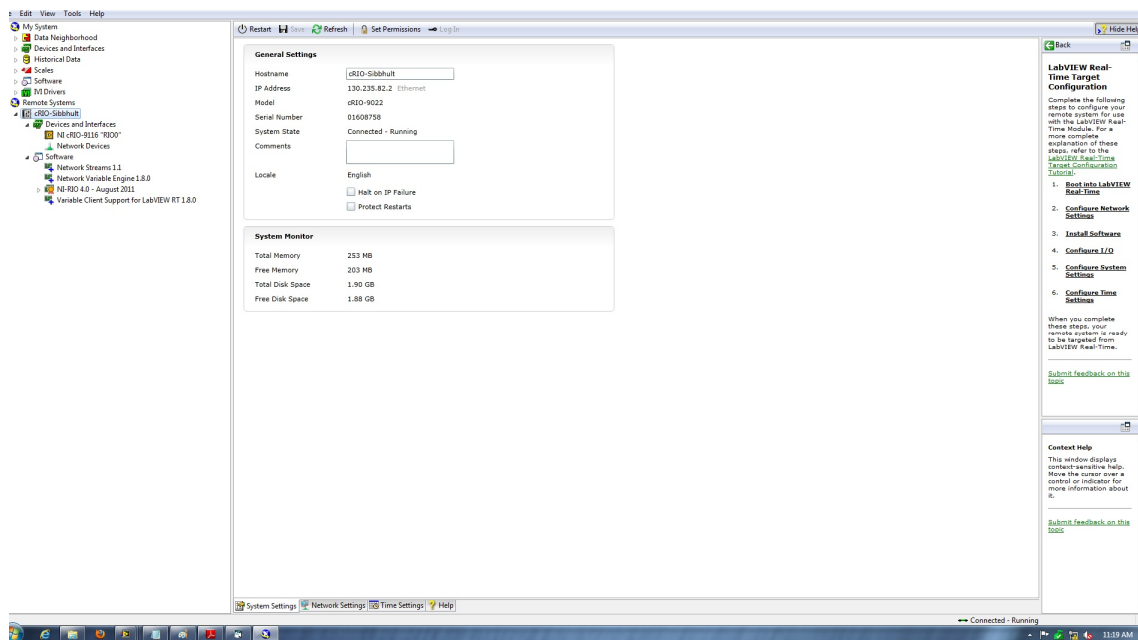


Figure 3 – MAX – General Settings

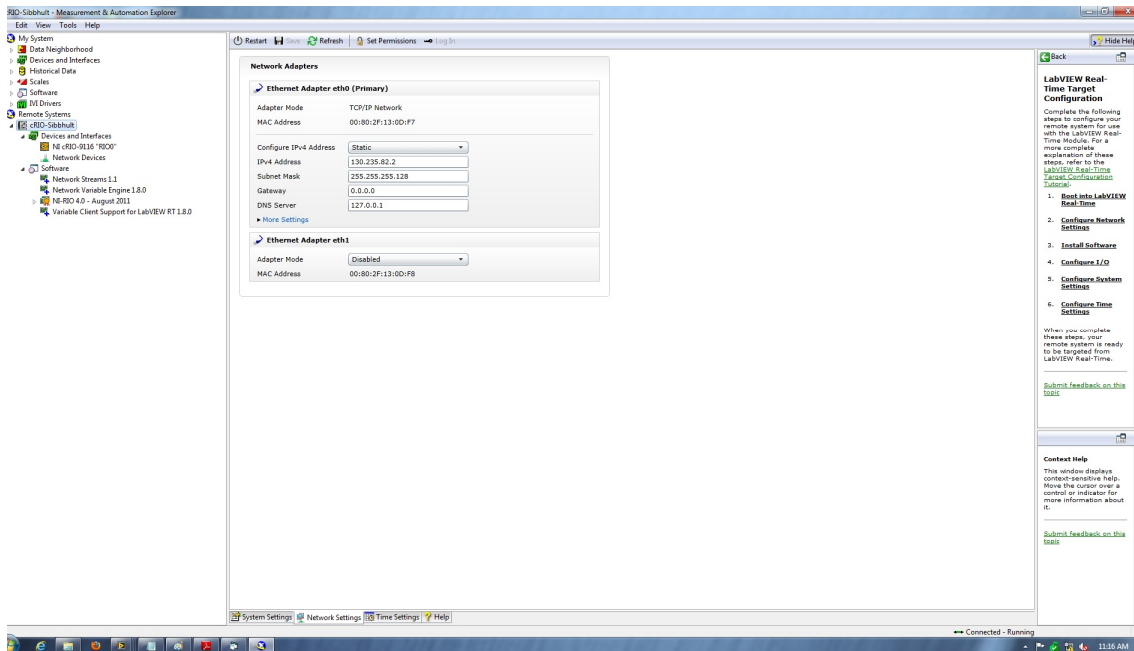


Figure 4 – MAX – Network Settings

The same manual, also introduces another software tool; “Comsoft Configurator II” and to use the 'Profibus Slave Module' specific “GSD” file to configure the NI Profibus module. The GSD file enables 'Master' Profibus module, in this case NI module, to recognise and communicate with the other nodes in Profibus communication, which in this project was limited to just one; Vacon Control Unit.

After having installed the GSD file, there are two more configuration parameters, at NI module end, that have to be selected, before communication could be established:

1. PPO type selection, and
2. Slave address

After taking all the steps as described above, the Comsoft Configurator II should be executed by checking the 'connect' symbol as shown in figure 5 below.

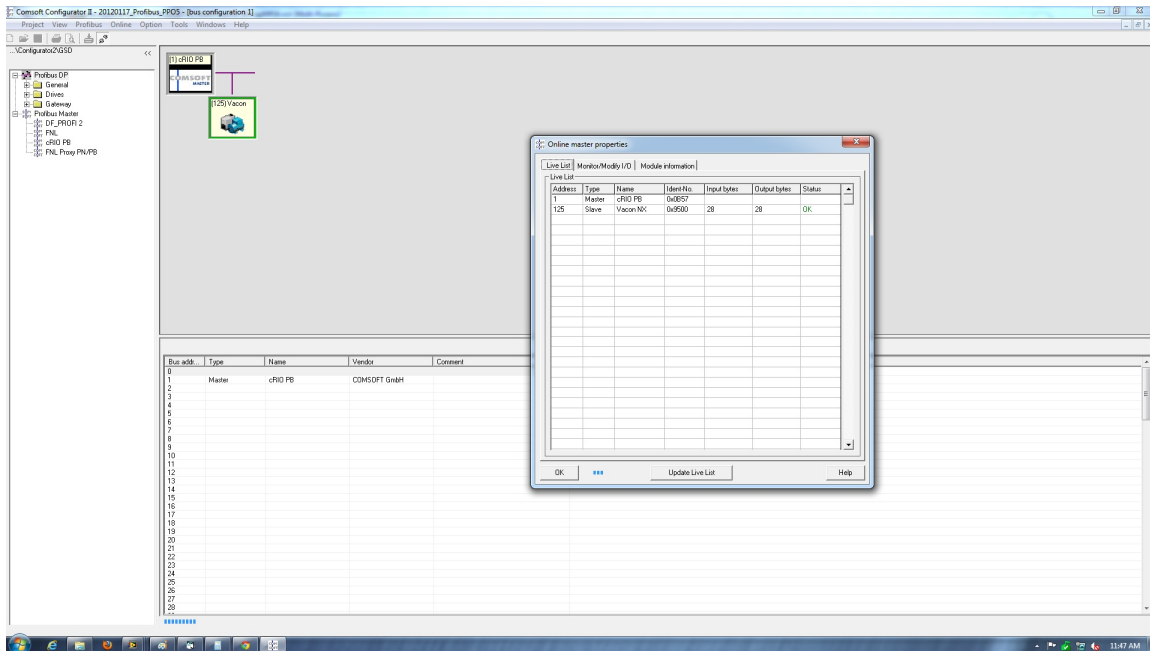


Figure 5 – Comsoft Configurator II

Then the online properties will display if the connection between LabView Master Module and Vacon Slave Module is 'OK', 'Fault' or 'Diag' [Diagnostic].

If the configuration is done correctly, the status should be 'OK' [Note1]

Profibus Module Configuration – Vacon

At the Vacon side, software 'NC Drive' was used to synchronise the PPO type and slave address for the Profibus Card 'OPT C3' [9].

Ref [6] was most useful to configure the Profibus at the Vacon's end.

1. Making and connecting the Profibus cable to the OPT-C3 option board
2. Fundamental knowledge of the Profibus protocol, using the Hex 'Control Word' and reading 'Status Word'.
3. Understanding of the Parameter/Process Data Object [PPO] and Process Data [PD].

Vacon Profibus Board OPT C3

On the Vacon side, a specialized card for Profibus Communication; Profibus OPT-C3 option board is required. This referenced card was acquired from Vacon and installed in the CU in accordance with document Ref [6]. Profibus cable was also manufactured to link the CU with NI module cRio.

Initially the program was used to send simple start-stop commands and some values, changing the parameters of e.g. frequency.

Vacon has its own program that runs on the CU, governing all the aspects of the running and stopping of the electric machine. Please see section 2.7 on Vacon Software.

To be able to run the electric machine, the corresponding application file was first downloaded to the inverter, which describes the operating parameters of the machine, e.g. Min and Max allowable Voltage, Nominal Voltage, Max allowable operating frequency, type of machine etc.

Downloading the 'Application File' in Vacon CU

The required software was installed with the help of USB to RS 232 connection from PC to the Vacon control Unit, using NC Load program. Then NC Drive was used to operate the electric machine.

ID Control and PD Data

After some initial programs to control and monitor the Vacon via LabView, it was determined that there are two ways to control and monitor the desired data:

1. First method is to use 'ID' control, where using id's in Hexadecimal, values can be assigned to and read from the Vacon Control Unit.
2. Second method for control and monitoring is to use the PD1 through PD8 to monitor eight different parameters, without having to address them individually.

The second method, monitoring via PD is faster compared to the first method.

In the early versions of the LabView programs written for the control/monitor of the EM, id control was used. Later versions used PD control for fast communication.

Even in PD control mode, id control is used at start-up to turn the frequency converter on, assign parameters to PD and at the end, same id control is used to turn off the frequency converter and reset the PD parameters to default values [Note 2].

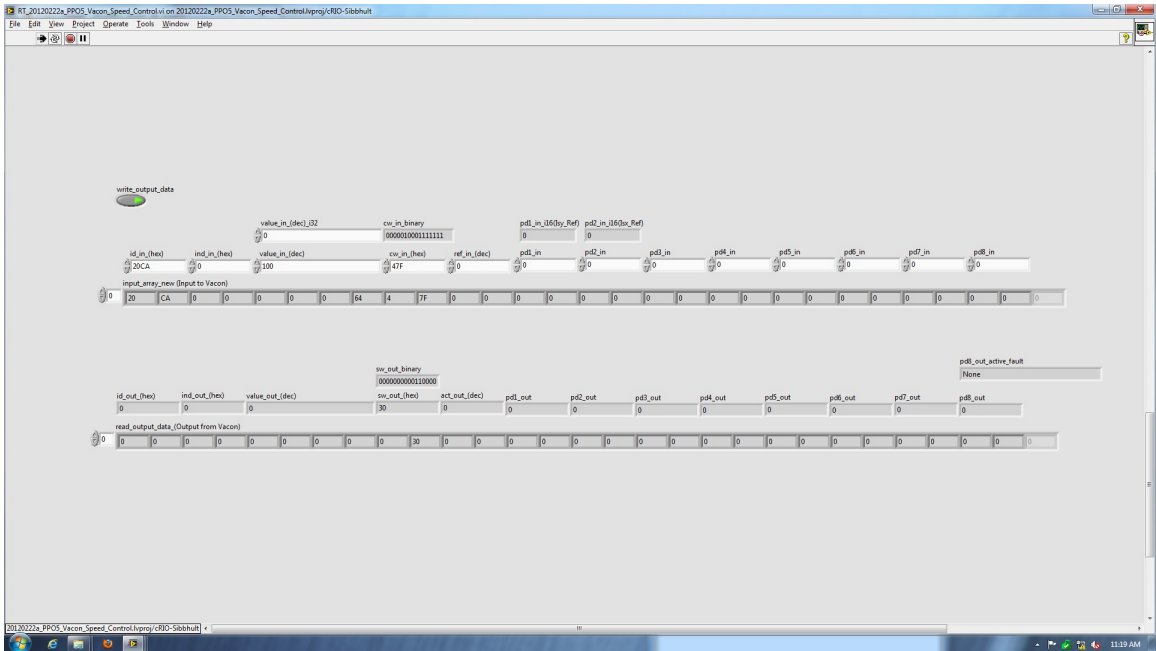


Figure 6 – ID Control

Control Word and Status Word

At first, examples from Ref [3] were followed to establish rudimentary control and monitoring of the parameters. Later, small programs were written to study and have control over the 'Control Word' and understanding of the 'Status Word'.

'Control Word' in Hexadecimal format is a command entered by Master to Slave, which is translated in binary according to 'Table 7-1. Control word bit descriptions' in Ref [3].

'Status Word' is also a Hexadecimal format response from slave to master, translated in binary according to 'Table 7-3. Status word bit descriptions' in Ref [3].

At first the above referenced tables were used to translate the control and status words as communicated back and forth between LabVIEW and Vacon to decipher the contained information.

Later, a program was written, to actually see the high and low bits, indicated by the glowing LEDs. This module, although not used afterwards, is still functional on the RT program, under the tab of 'control word' and 'status word'.

The 'Control Word' and 'Status Word' in themselves are not enough for a complete communication, other parameters such as 'ID', 'IND', 'VALUE', 'PD' and 'ACT' makes the complete communication as described in following section; PPO types. Figures 7 and 8 below shows the reading of Control and Status Word,

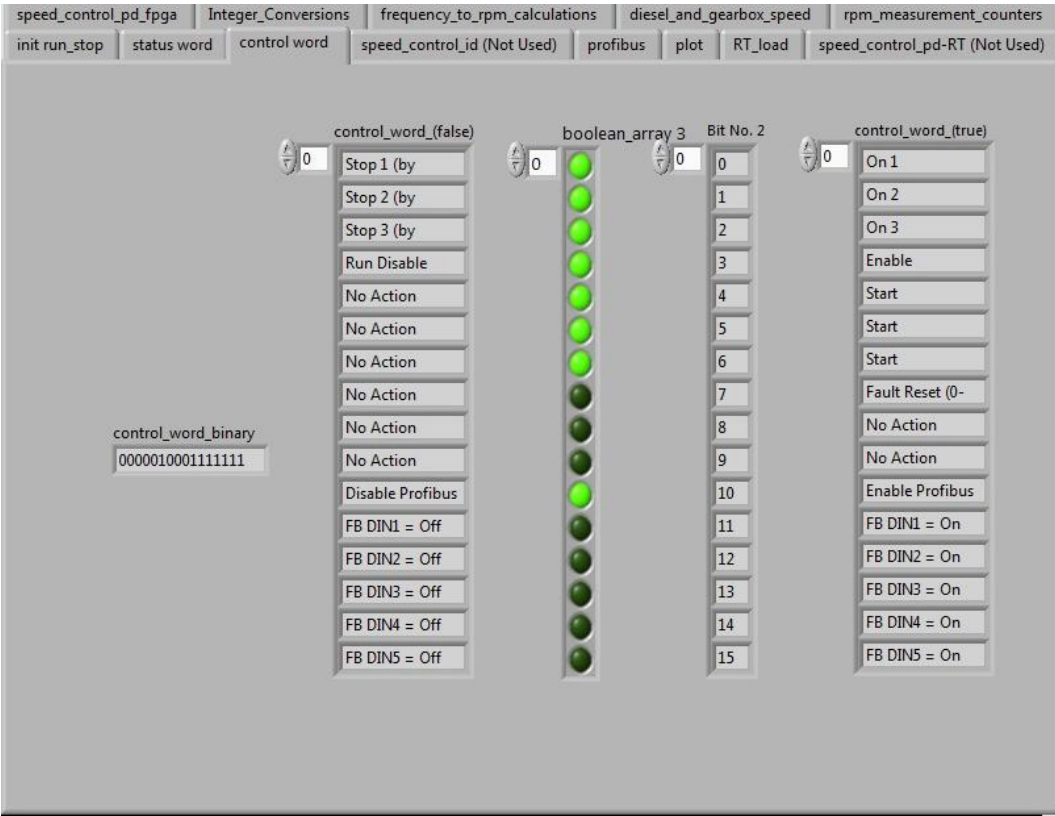


Figure 7 – Control Word

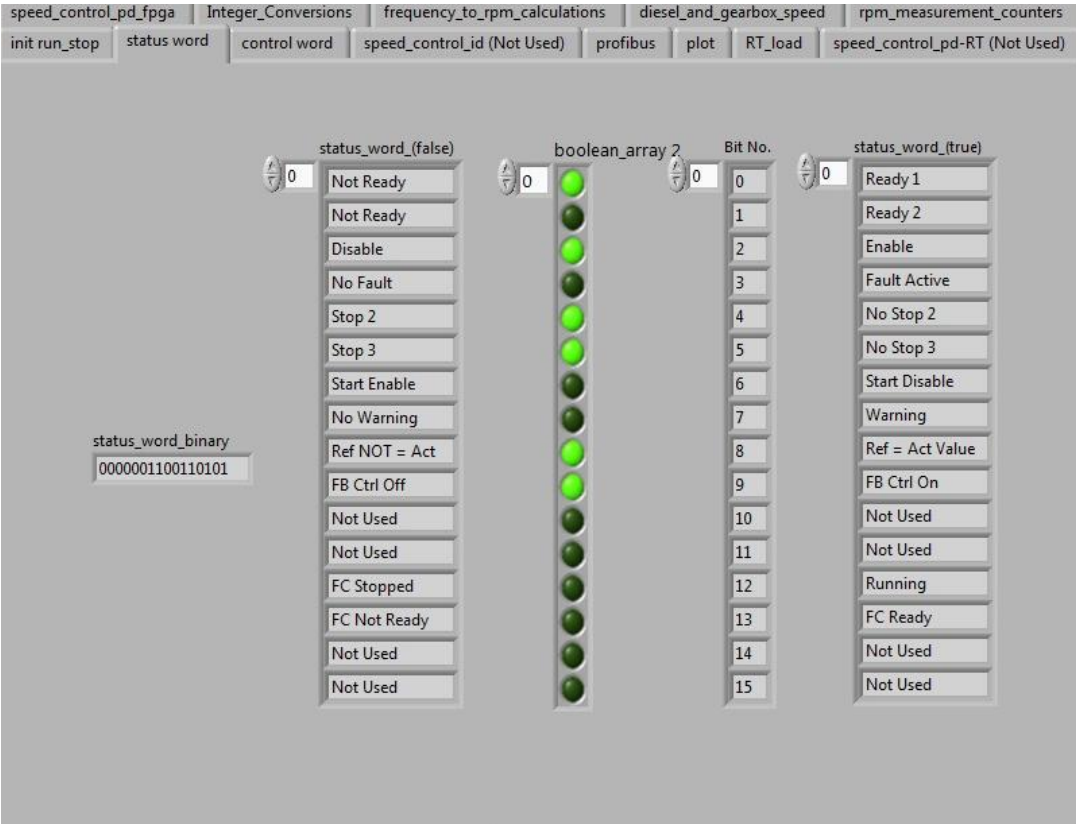


Figure 8 – Status Word

PPO Types

Vacon manual, Ref [6], section 7.3 describes the PPO (Parameter/Process Data Object) types in Profibus, shown in figure 8.

7.3 PPO types

PPOs (Parameter/Process Data Object) are the communication objects in PROFIBUS DP.

The PPOs in Vacon NX:

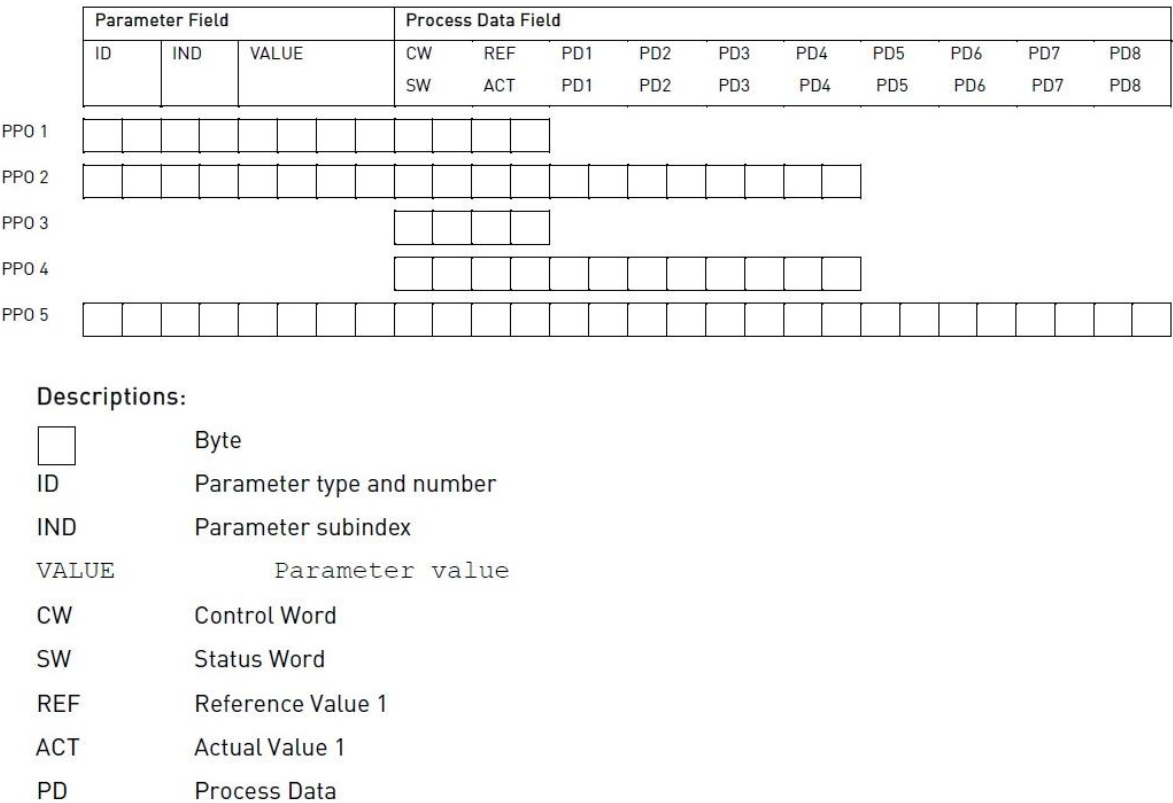


Figure 9 – PPOs (Parameter/Process Data Objects)

The generic example that was created, following the example in the manual, was using eight bytes of data, which does not correspond to any specific PPO type.

PPO 5 was selected for the project, considering the number of parameters/process data required for the task.

The example program, written earlier was modified to accommodate 28 bytes, instead of 8 that it was designed for. The program running by itself, could not be verified to be working unless the LabView side; NI Module cRIO PB is to be linked with Vacon Control Unit.

Data Types Used in LabView

LabView can use several types of data representation;

1. Unsigned – U8, U16, U32, U64
2. Integer – I8, I16, I32, I64
3. Floating point
4. Double precision
5. Fixed point
6. Decimal
7. Binary
8. Hexadecimal, etc.

All of the above mentioned data types, depending upon the specific application and ease of man machine interface, were used in this project. At most places, data types transformation was done to be able to communicate between various modules of the LabView program and between LabView and Vacon.

It is absolutely imperative that this data type transformation be done in such a way so as not to lose part of the information, make it meaningless or worse make the inverse of intended message; e.g. direction of motor rotation etc.

A review of the various data types and their transformation was done and a program module was written and incorporated into the RT program, under the tab of 'integer conversions'.

Profibus card – OPT C3 (Section 2.4.2.2.1) at Vacon control unit, expects bytes of data. Most of the parameters are in the form of one byte and for this reason, data type of 'U8' was used to represent the byte [6].

At places where more than one byte was needed to be combined or split, as in the case of parameter 'VALUE', it was done in the program by adding and splitting the bytes.

Furthermore, while the 'Process Data', e.g. 'ID', 'CW' uses Hexadecimal commands, it is easier for the operator to use data type of integer to enter a value and see the feedback from Vacon in integer form, with the accompanying sign (positive or negative).

For this purpose data type of integer was used for the user to enter and read various desired values, while the program would take these values and convert them into 'U8' for LabView Vacon communication.

In the earlier version of the program, two main data types; 'Hexadecimal' and 'Integer 16 (I16)' were utilized.

Later versions, including the final version of the LabView program, have automated Hexadecimal commands for start, stop and applying the 'ID' command in the initialization and exit phase to configure the 'Process Data Field'.

All the user inputs;

1. Isy* in current control mode.
2. Speed reference in 'speed control' mode.
3. Field weakening, Isx* current

are in integer form with plus and minus sign.

Similarly, the feedback from Vacon is also converted into integer form with sign to be displayed on the main RT interface screen.

Scaling of Numbers – LabView to Vacon

In NC Drive, the 'Application file' defines the motor nominal current (mnc) and motor nominal voltage (mnv), among numerous other items, under 'Parameters'.

Same application, under 'Keypad Control' allows input values for 'isx*' and 'isy*' in % of nominal current, e.g. 300% of nominal current input is allowed under application: FastPB\APFIF16_Multipurpose_IEA\apfiff16.vcn [12].

The % range is also defined there for each isx* and isy* as [-300% to 0] and [-300% to 300%] respectively. 'mnc = 40 Amps' and 'mnv = 212 Volts' for the Haldex machine.

Example (isy*) : Entering 3 in the 'value column' for isy* means --> 3% of the 'mnc' --> $3\% \times (40/100\%) = 1.2$ Amps. To make an equation/translation from % to Amps :

$(RI) \times (mnv)/100 = \text{Amps}$, where RI: Referenced Input.

For example, if the input from user is required to be in Amps, this value should be converted into the RI to make it understandable by the Vacon Control Unit.

$RI = (\text{Amps} \times 100)/mnc$ --> $RI = (1.2 \times 100)/40 = 3$, which is the 3% input in above example.

There is a factor of 10 converting from 'Vacon % Values' to 'LabView % Values' →

Value of 3 in above example would be shown as 30 in LabView. --> To be able to take a number for Amps and converting it into Vacon number would be as follows:

$RI/10 = (\text{Amps} \times 100)/\text{mnc} \rightarrow RI = (\text{Amps} \times 1000)/\text{mnc}$. This conversion is what

LabView 'Flat Sequence'; 'speed_control_pd_fpga' is implementing.

The value 'isy*_input[A]' is the value entered by the user as Amps, which is then multiplied by 1000 and divided by 'mnc' to acquire a 'RI', which is 'current_ctrl_isy*' and this is fed into the fpga via Read/Write Control.

Abbreviations 'mnc', 'mnv', 'RI', and 'Amps' are not used in Vacon program, rather they were adopted to demonstrate the example and to implement the scaling in LabView.

Scaling of Numbers – Vacon to LabView

LabView 'Flat Sequence'; 'read and write profibus data Crio ↔ Vacon, implements the back scaling for the 'isx*', 'isx', 'isy*', 'isy', 'usx*' and 'usy*' in sequence steps 5 – 8.

Electric Machine's nominal current and voltage is displayed in the LabView RT program under the title of 'Motor Nominal Current & Voltage'. These values must be synchronized with the values as listed under 'Basic Parameters' in the Vacon's 'Application File' for the specific electric machine. As an example, running this program for ExSAM, which among other differences, have $\text{mnv} = 400$ and $\text{mnc} = 100$ will obviously result in incorrect readings.

To change these values in LabView to accommodate a different motor, scaling on both ends; LabView to Vacon and Vacon to LabView must be changed accordingly.

2.5 Operating LabView

First the PC program is launched and afterwards the RT program. The underlying layer of the LabView program, which is FPGA is summoned by the PC and RT and does not need to be executed independently.

2.5.1 PC Program

This PC program is written by PhD student Yury Loayza and is incorporated in the LabView Project with slight modifications. The main purpose of this program is to store the desired data, generated during the electric machine operation, for further analysis using for example Matlab program.

This program is also used to select, display and merge signals on a single graph to visualize various time varying signals.

2.7.2 NC Load

It is a PC program used to download either system software or application to the inverter. [9].

2.7.3 NC Deff

It is a PC program to manage the 'variables' database of the application [9].

2.7.4 NC 1131

This allows editing of the source code, which can then be compiled and downloaded to the inverter via NC Load [9].

2.7.5 NC Drive

This is like a PC program, which is the user's operation interface. NC Drive has the option menu to, beside other things, keep the control to NC Drive, give it to 'Control Pad' OR to give the control to 'FieldBus', which in this case is Profibus [9].

Chapter 3 Test System

3.1 System Description

The complete system, as depicted in figure 1, comprises of following discrete units;

1. 160 kW PMSM
2. Vacon frequency converter
3. Vacon Control Unit, with its task specific cards in their specific slots
4. Vacon Control Pad
5. CRio with Modules
6. Buffer circuits, relay cards, amplifier, relays and connectors
7. Diesel Engine
8. Transmission
9. Sequential Gear Box
10. Flywheel
11. Retarder
12. Servo valve for the retarder brake
13. Encoder for Diesel engine RPM
14. Encoder for Transmission - Volvo speed sensor
15. Throttle control for Diesel engine

Figures 11 and 12 below shows the complete system;

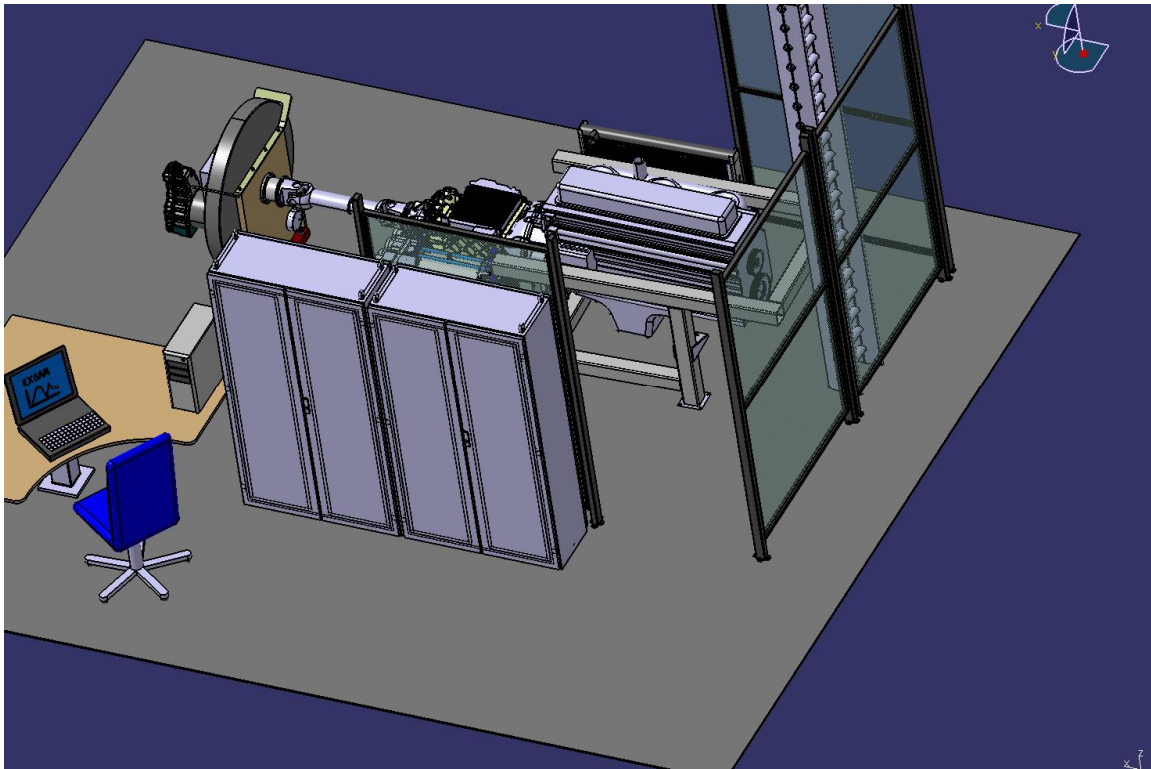


Figure 11 – Test System, view 1

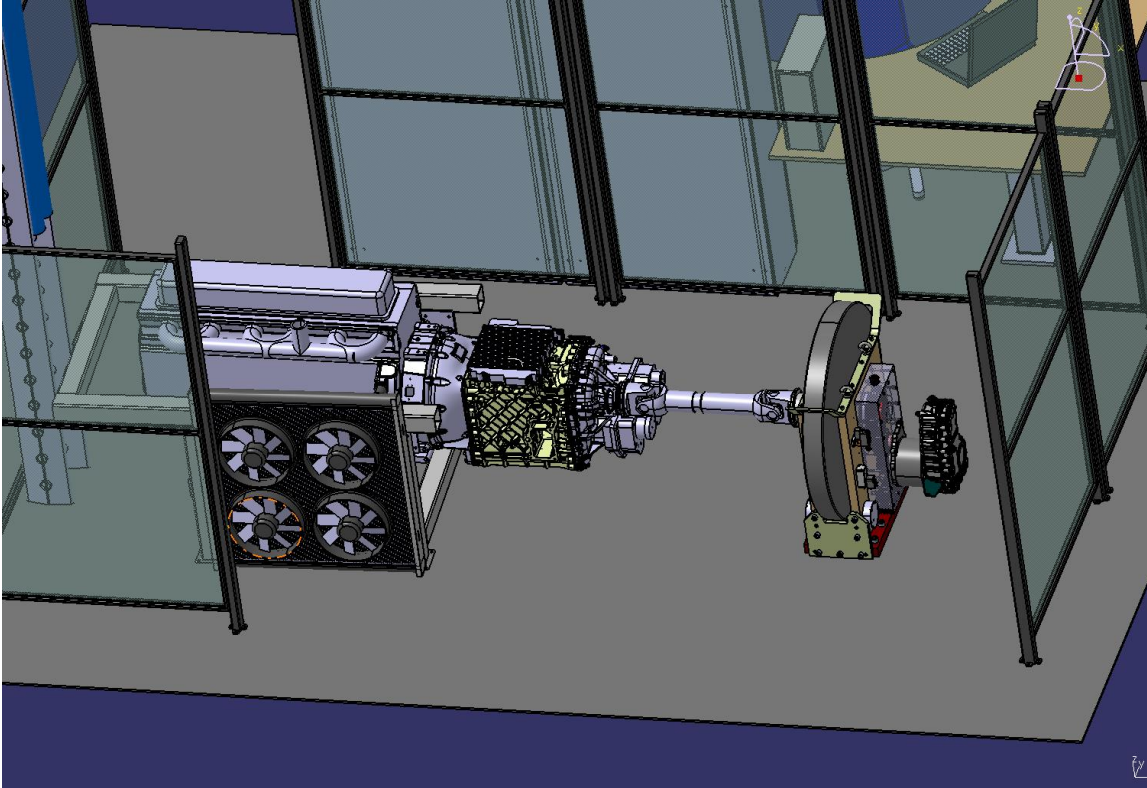


Figure 12 – Test System, view 2

3.1.1 Functionality

1. The core aspect of the program was to control the PMSM via fast communication to be able to have a dynamic control of the machine.
2. Control and monitoring of the Diesel Engine.
3. Control of the simulated load.
4. Control and monitoring of the coupling mechanisms;
 - 4.1 ExSAM coupling to the transmission
 - 4.2 Diesel engine coupling to the transmission

3.1.2 Power Supply in Lab6 (IEA)

During the development phase at IEA, lab 6, Haldex machine and the DC link voltage of 200 Volts was used from the dedicated output sockets available in the lab.

3.1.3 Power Supply at Sibbhultsverken

Vacon frequency converter 'NXP' series was used at Sibbhultsverken. This particular converter is able to provide 520 Amps and 380 -500 VAC, which is more than what is needed for the ExSAM.

Chapter 4 Control, Monitoring and Data Storage

4.1 System Control from LabView

LabView via CRio was connected to all the functions of the test rig, which are summarized below

1. Control of the frequency converter:
 - a) Electric machine current control
 - b) Electric machine speed control.
2. Monitoring and measurement of I_{sx}^* , I_{sy}^* , I_{sx} , I_{sy} , U_{sx}^* , U_{sy}^* , DC link Voltage and electric machine rpm and frequency.
3. ExSAM and Transmission coupling control and monitoring.
4. Clutch engagement and disengagement between ICE and GB control and monitoring.
5. Analog control for the retarder brake on the flywheel via servo control valve, clutch control and possible oil pumps control
6. Eight (8) channel digital output for :
 - a) Engine enable signal,
 - b) Exhaust brake,
 - c) Flywheel brake
 - d) Exsam coupling/decoupling
 - e) 4 channels for the throttle control
7. Recording of desired data in files for further analysis.

4.1.1 Electric Machine Control from LabView

Two modes of controls; current and speed were programmed and used to control the ExAM. For any of these controls to work, control over the Vacon frequency converter was a prerequisite.

Current Control Mode

After having established the communication via Profibus protocol, the current control worked well, while the RPM was kept low since initially no field weakening current ' i_{sx} ' was applied. For this mode, the PID controller was bypassed with direct input of I_{sy}^* and I_{sx}^* if required.

Field weakening control by using I_{sx} was utilized. The program was designed in a way that the field weakening control could be used in either current/torque control or speed control mode.

Speed Control Mode through PID Controller

A PID controller was at first used in the RT program to check the functionality of the speed control. Electric machine RPM was used as an input/reference to the PID controller, while I_{sy} was manipulated by the PID controller output to achieve and maintain the motor speed.

The program was extensively tested in the lab on Haldex machine to get a predictable response. Later, the same LabView program was used at Sibbhult on ExSAM on March 20th 2012 (More on section 6.2).

For the system control as listed under section 4.1, separate programs were written as described under section 2.3.

4.2 Monitoring Parameters

Following is a list of parameters, which were monitored by LabView;

1. Currents – i_{sx}^* and i_{sy}^* - Referenced
2. Currents – i_{sx}^* and i_{sy}^* - Acknowledged (Vacon)
3. Currents – i_{sx} and i_{sy} – Measured (Vacon)
4. Voltages – u_{sx}^* and u_{sy}^* - Referenced
5. Voltage – DC Link
6. RPM – Electric Machine
7. RPM – Diesel Engine
8. RPM – Transmission
9. ExSAM and Transmission Coupling
10. Diesel Engine and Transmission Coupling
11. Temperature Monitoring of the Electric Machine

4.3 Data Storage

Data storage program was incorporated into the project at various levels of programs as described in the following three sections [10].

4.3.1 FIFO Implementation in FPGA

First In First Out (FIFO) program was implemented in the FPGA to acquire the raw data (Unscaled) from the FPGA.

4.3.2 PC Program and Data Saving

A PC program was also modified and incorporated into the project to select, display and store the desired FIFO data on the PC for further analysis in MatLab.

Chapter 5 Safety System

5.1 Safety in LabView Program

Safety features in LabView program modules were written and incorporated in the second iteration of program writing, after having established the communication link between the two systems in first version of programs.

5.1.1. Electric Machine Control

Initial LabView programs did not have any safety features and it was possible to suddenly stop a fast running machine without considering the back emf. This was fixed in later versions of the program, where now once all the inputs are in place, it is not possible to force stop the machine. Even with the 'STOP' button pressed, depending on the RPM of the machine, the program will gradually reduce the RPM to zero in a controlled manner.

RPM input limitation, in accordance with the DC Link voltage still needs to be incorporated. This is necessary, since Vacon has its own isx* injection at a point where a RPM request exceeds the DC link voltage capacity.

5.1.2 Clutch Control - Diesel Engine to Transmission

This clutch requires 10 seconds to fully engage or disengage, which the LabView program catered with an option, as was required, to be able to select engage while it is in the middle of disengage cycle and vice versa.

5.1.3 Clutch Control – ExSAM to Transmission

This clutch control was more critical, since it required both Diesel Engine and Transmission RPM to be close to zero before it could either engage or disengage.

Diesel Engine and Transmission, both have digital encoders; one directly coupled and one proximity sensor. Output from these two encoders was used in the program module controlling the ExSAM and Transmission coupling.

5.1.4 Engine Enable and Throttle Control

Engine enable command can only be initiated if the 'Exhaust Brake' and 'Flywheel Brake' signal is not present. Throttle could be operated once the engine enable command is present.

5.1.5 Flywheel, Exhaust Brake and Engine Enable

Once the 'Engine Enable' command is withdrawn or 'Exhaust Brake' and/or 'Flywheel Brake' signal is detected, the throttle will go to zero and engine will be disabled. This state will persist, even after the 'Exhaust Brake' and/or 'Flywheel Brake' signals are taken off.

5.1.6 STOP Command in RT Program

There is only one stop switch in LabView program, not counting the 'Forced STOP', which is independent of any LabView Program environment.

This stop function is so linked with the safety program, that once pressed, it will not immediately shut down the system.

The program will check the following parameters;

1. EM speed, regardless if operating in current control or speed control mode.
2. Diesel Engine speed.
3. Gear Box speed.
4. ExSAM and GB coupling.

At the press of the 'STOP' switch, the program takes the following steps;

1. Assigns zero current (including isx^*), or speed reference to EM, depending upon the mode of operation.
2. Disables the Diesel Engine and puts throttle to zero.
3. Disengages Exhaust and Flywheel brake, not the retarder brake.
4. Monitors the RPM from EM, ICE and GB.
5. If in 'speed control' mode, the program continues to use the output from the 'PID' controller to control 'Isy' current to gradually take the RPM of the electric machine down.
6. Only after the RPM of the EM, ICE and GB are reduced to almost zero, does the program allow the ExSAM and GB be decoupled (if they were coupled).
7. Initializes the current and speed references previously assigned to EM.
8. Initializes all the clutch and ICE controls.
9. PD values are assigned the default values, instead of the one being used by the LabView program. [NOTE 2,3]
10. Initializes various program counters and displays.
11. Turns off the frequency converter/inverter.
12. Stops the FPGA program execution.
13. Stops the execution of the RT program.

The RT program does not have any input for the 'Diesel engine and Transmission Coupling' position determination.

At the launch of the RT program, to ensure if the clutch is in engaged or disengaged position, either control should be activated and position physically checked, by trying to turn the wheel.

However, once the position has been determined and either control has been used at least once, the clutch position indicators; 'engaged_ind' and 'disengaged_ind' LEDs can be safely used to ascertain the clutch position as long as the 'STOP' command has not been used.

5.1.7 Reset Command in RT Program

Just like Stop command, there is only one Reset command, denoted by 'reset_init' reset initialization. However, unlike the Stop command, which can be used at any time, the reset command is only operational once 'Step 1' and 'Step 2' are taken and the electric machine is either in ready mode or in 'speed or current control' mode. Reset command ready status, indicated by 'ready_reset' LED indicator only lights up, once Step 1 and 2 are taken.

Once initiated, the reset command takes the following steps;

1. Puts the throttle to zero.
2. ready_reset indicator goes off.
3. Assigns zero current (including isx^*), or speed reference to EM, depending upon the mode of operation.
4. Monitors the RPM from EM, ICE and GB.
5. If in 'speed control' mode, the program continues to use the output from the 'PID' controller to control 'Isy' current to gradually take the RPM of the electric machine down.
6. Only after the RPM of the EM, ICE and GB are reduced to almost zero, does the program allow the ExSAM and GB be decoupled (if they were coupled).
7. Initializes the current and speed references previously assigned to EM.
8. PD values are assigned the default values, instead of the one being used by the LabView program. [NOTE 2, 3]
9. Initializes various program counters and displays.
10. Turns off the frequency converter/inverter.
11. Takes the program at the start of 'Step 1', indicated by 'ready_init' led indicator.

5.2 NC Drive parallel execution

Vacon program; NC Drive is executed even before the LabView program is launched. This ensures a safety measure that in case if the LabView program is not responding, control could be moved to NC Drive to assign zero references.

5.3 Emergency Stop

An emergency stop button was installed by Sibbhultsverken on the frequency converter cabinet, easily accessible to the operator, in case of a requirement of emergency shutdown.

This switch, once activated, disconnects the frequency converter supply voltage to the electric machine while also disabling the buffer circuits, in essence disconnection the CRio from the system.

Chapter 6 System Commissioning

6.1 Interfacing the CRio with System at Sibbhultsverken

First interface of CRio with the system at Sibbhultsverken was time consuming due to several factors;

1. Commissioning of the Vacon frequency converter [11].
2. Limited availability of all the required equipment and tools.
3. Requirement of connectors at the CRio end to interface with the various components of the system.

6.2 First Experimental Run of the ExSAM w/o load

First trial of the ExSAM, with current control worked well on March 20th 2012. The ExSAM was run without a load.

Speed control, which at the time was located in the RT program, was also tried. Although the current controller worked fine, the speed controller was not very smooth in operation. There was certain amount of fluctuation around the 'Set RPM', which was more than expected.

Afterwards, once back at IEA, the program was modified to transfer the PID controller from RT to FPGA, besides other changes. Since RT works at millisecond and FPGA at microsecond scale, it was believed that this change will speed up the communication rate to have the dynamic control of the PMSM.

6.3 Second Run of the ExSAM with Load

On May 8th and 9th 2012, the new version of the program was tested at Sibbhultsverken. While the current controller worked fine, the speed controller had large fluctuations around the 'Set Speed'. ExSAM gear coupling could be heard knocking. In short, the speed controller did not work the way it was supposed to.

The speed control mode did not work as was expected due to a slow communication rate between LabView and Vacon. Where the communication rate was expected to be around 10ms, the rate observed was 120-180 ms, rendering the dynamic control ineffective.

6.4 Observed Limitations

The PC program was observed to determine delay in communication between LabView and Vacon. It was observed that although the refresh rate or sampling frequency is high enough; approx 5-6 millisecond for displaying motor RPM etc., the acknowledgement of reference current values (I_{sy}^* , I_{sx}^*) from Vacon after having received them from LabView is taking 150-160 millisecond. This time frame for acknowledging and updating the current values (I_{sy}^* and I_{sx}^*) should have been in 5-6 millisecond range as well.

6.5 Search for Causes

In essence what was happening that the PID controller in the FPGA was working fast enough to be able to control the PMSM, however the feedback from the Vacon on the actual values of I_{sy} and I_{sx} was taking so long that each time these values were made available at PID controller, they were either quite high or very low from the 'Set/Reference Value'. The PID, in an attempt to bring these value close to referenced value would apply the full allowed current in the opposite direction, which will make the system jerk and at the next sampling period, the cycle will be repeated.

6.6 Softening of PID Controller

The proportional and integral gains of the PID controller were reduced to decrease the amount of current input against any given reference speed. This reduced the current ramp gradient towards the set point and allowed the 'Static Speed Control' at the cost of sacrificing the 'Dynamic Speed Control'.

This was merely a temporary solution to run the system in speed control mode, while the reason for communication delay is further studied and discussed with Vacon.

6.7 Vacon Frequency Converter Malfunction

On the day of May 10th 2012, while running the ExSAM, the Vacon frequency converter made a loud noise before shutting down. On later inspection, it was observed that there was some internal burning of the components. By design the frequency converter, along with all the high voltage equipment is inside the metal enclosure, and that contained the affects of that incident.

6.8 Fast Profibus

Vacon provided feedback with a manual, Ref [14] and an application software; "Advanced Application".

Following figures shows the delay time for "Normal" and "Fast" Profibus.

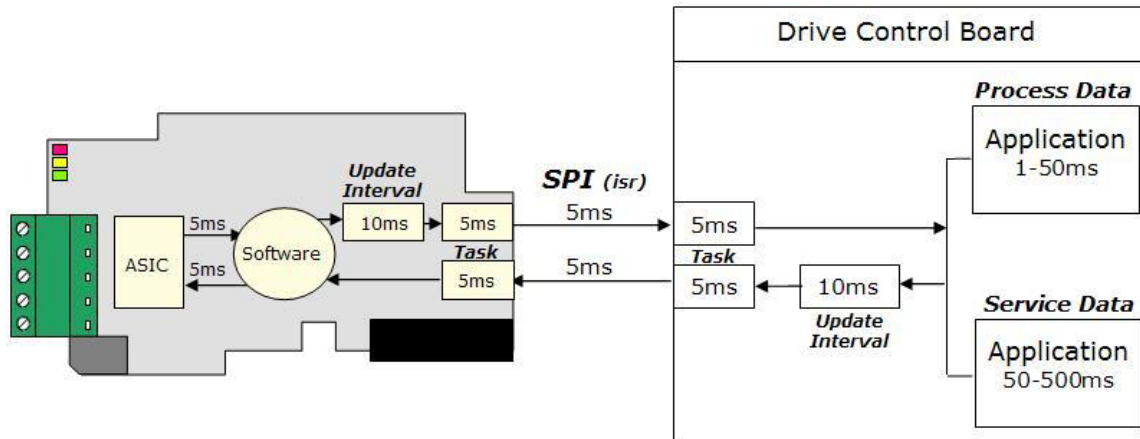


Figure 13 – Delay Time for “Normal” Profibus

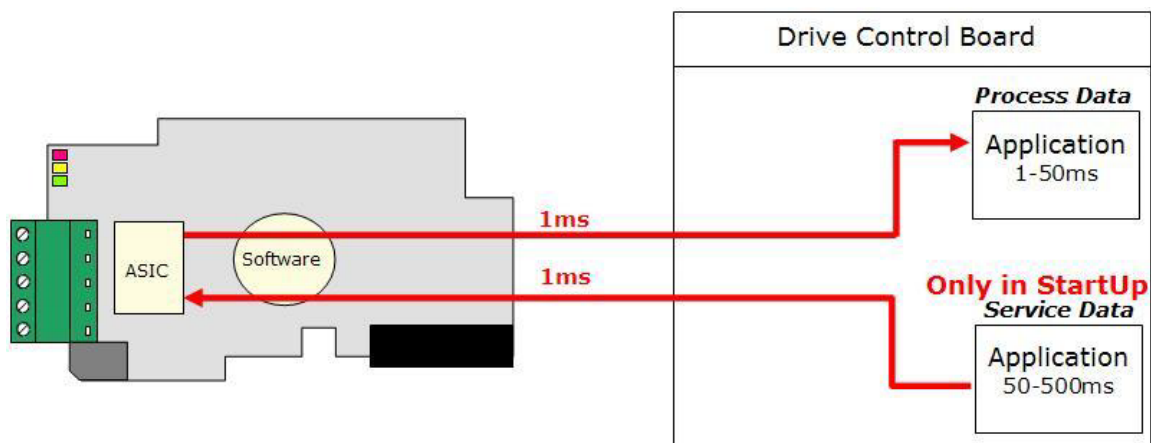


Figure 14 – Delay Time for “Fast” Profibus

In the 'fast mode', ID control does not work. This means that assigning/configuring the PD in values at the start of the program for isx^* and isy^* and then initializing the PD in values at the end of the program is no longer possible, since it uses 'ID Commands'.

NC Drive was used to assign the PD in data values to specify isx^* and isy^* . Unlike LabView, which does not have memory, NC Drive records these configurations. Drawback, initially was that with PD in configured, NC Drive could not be used to assign or initialize the isx^* and isy^* . This issue however, was fixed in the next revision of 'Application File' which was loaded in the Vacon CU.

Chapter 7 Data Analysis and Results

7.1 MatLab for Data Analysis

MatLab was used [10] for data analysis.

7.2 Results

Various ‘Modes’ of Profibus were used, after having detected the delay in communication as described under section 6.5. The three modes that were used are as follows:

1. Profibus Mode, which proved to be very “Slow”
2. “Fast” Probibus Mode
3. “Bypass” Mode in Profibus

Analysis was done in the MatLab, where the input signal/command for ‘isy’ and ‘isx’ as recorded in FPGA was compared to the acknowledging and applied/measured currents ‘isx’ and ‘isy’ in millisecond time frame.

7.2.1 Delay Time in “Slow” Profibus Mode

Following figures shows the recorded delay in “Slow” Profibus mode.

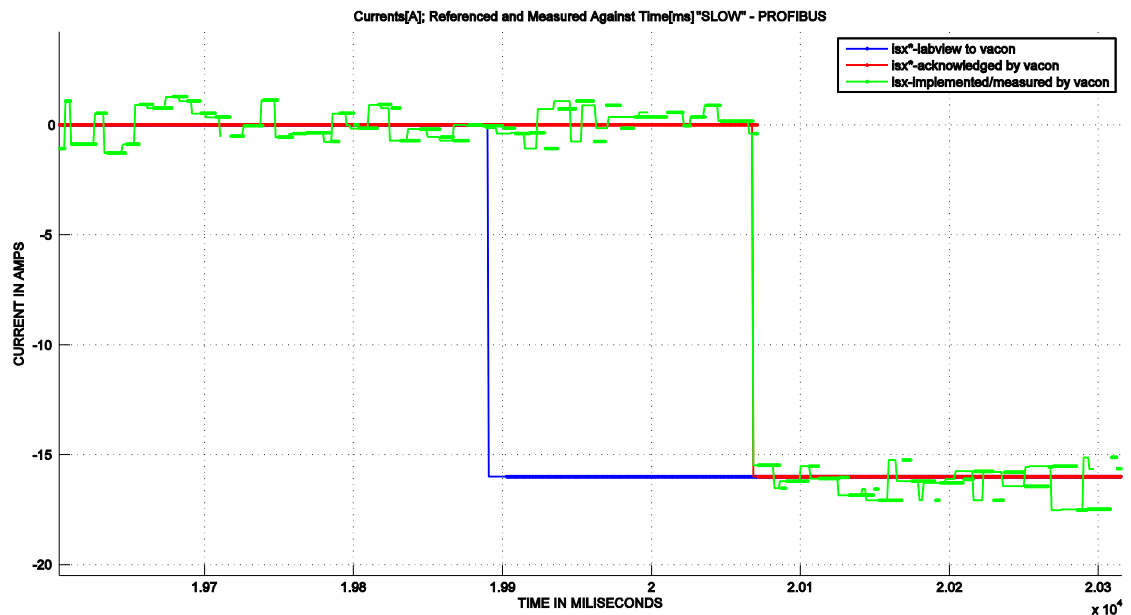


Figure 15 – Delay isx_180ms

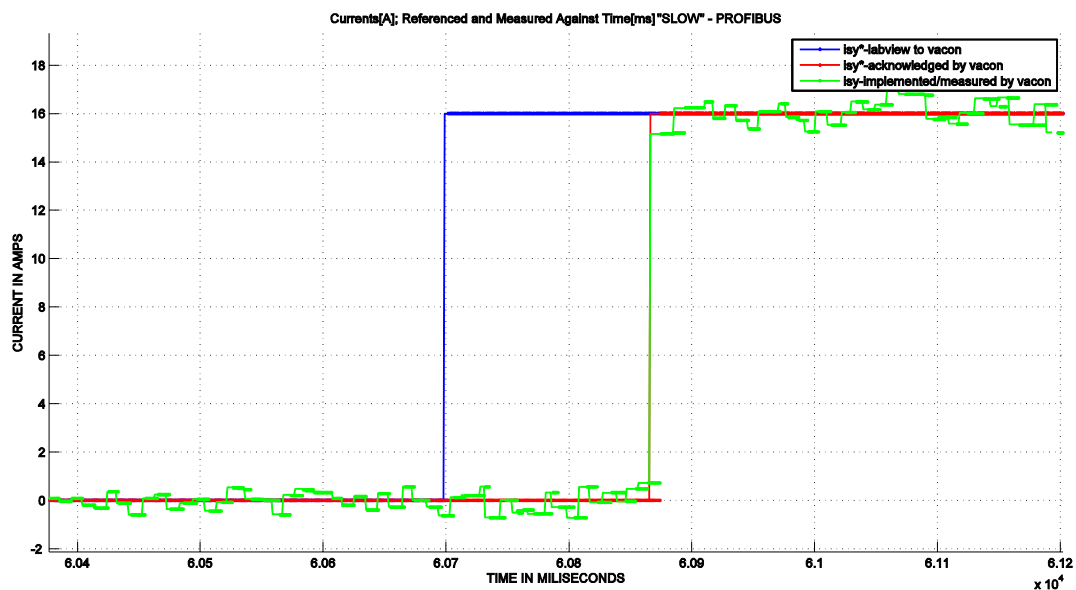


Figure 16 – Delay isy_{180ms}

7.2.2 Delay Time in “Fast” Profibus Mode

Following figures shows the recorded delay in “Fast” Profibus mode.

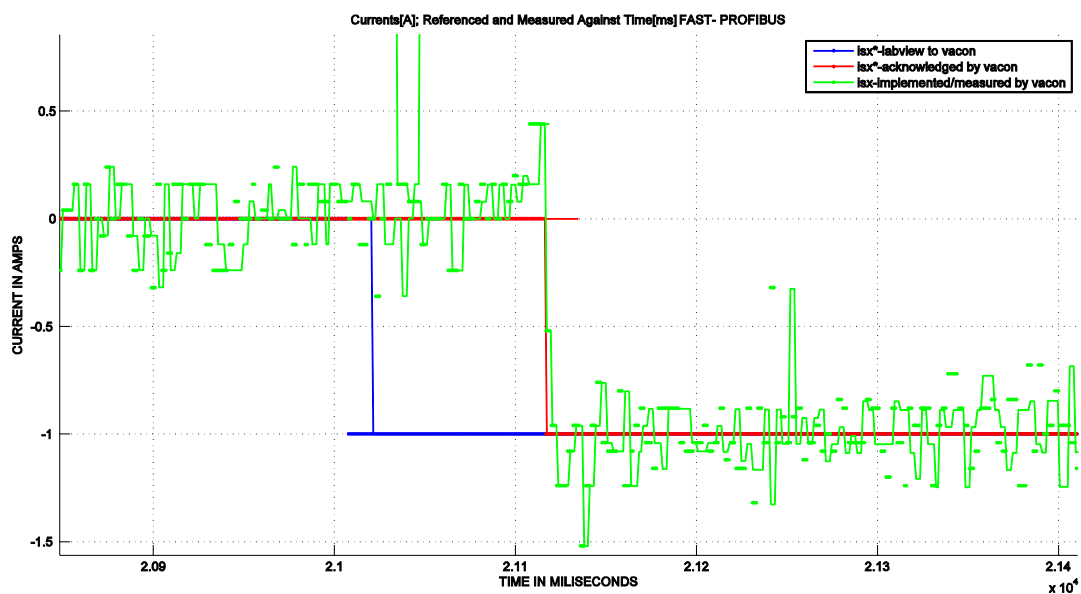


Figure 17 – Delay isx_{110ms}

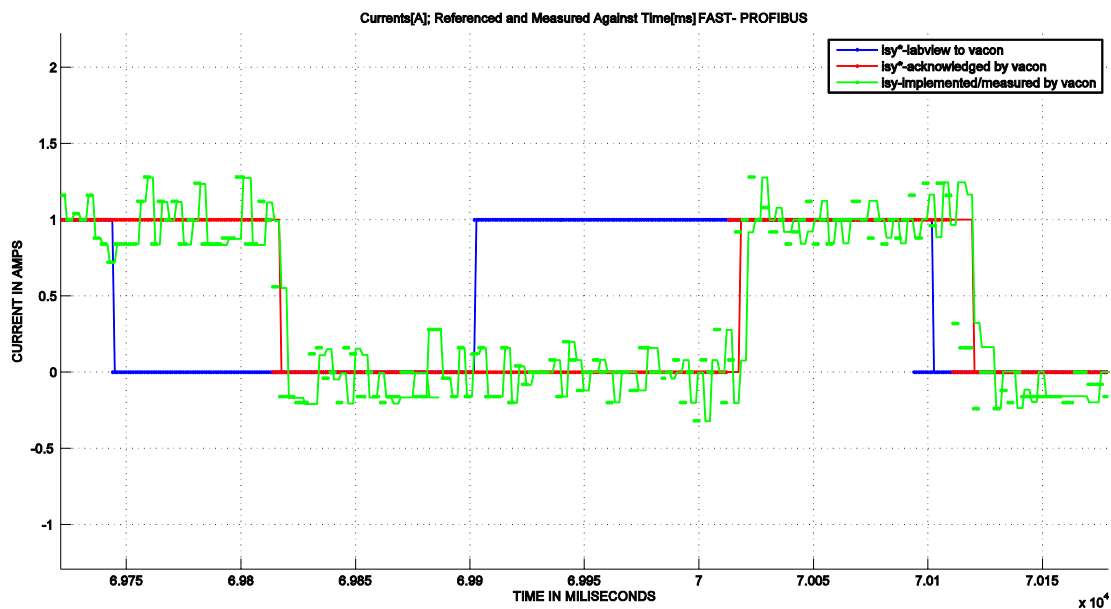


Figure 18 – Delay isy_110ms

7.2.3 Delay Time in Bypass Mode – Profibus

Following figures shows the recorded delay in “Bypass” Profibus mode.

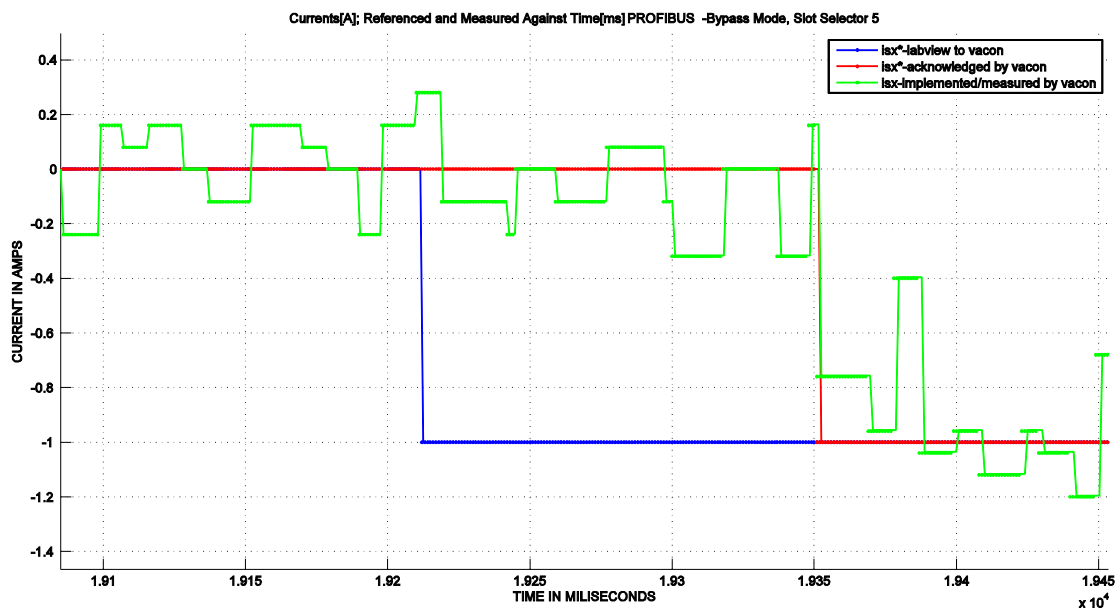


Figure 19 – Delay isx_140ms

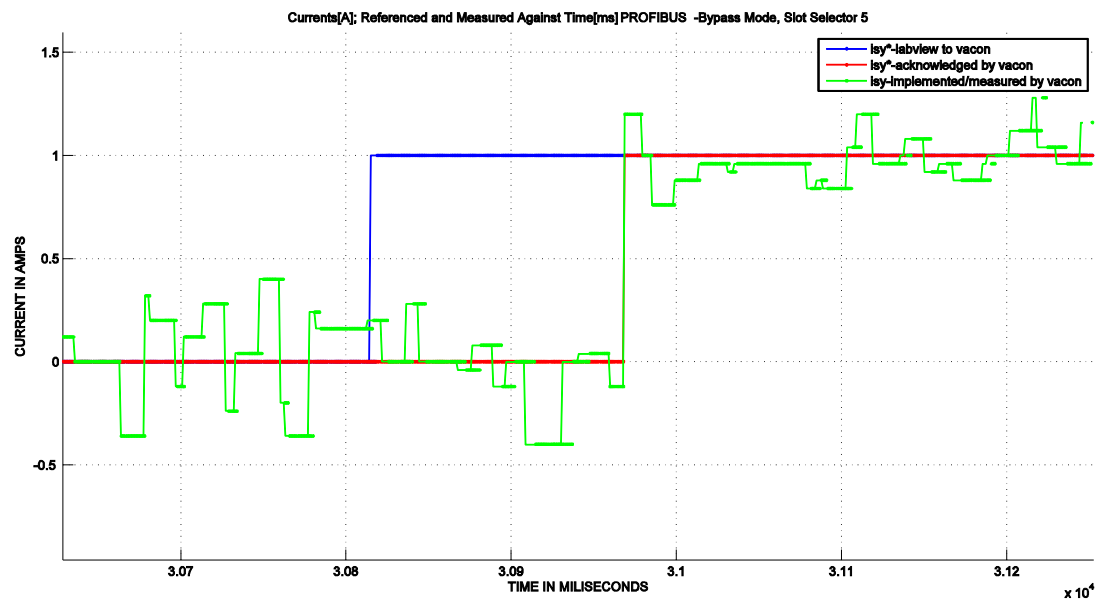


Figure 20 – Delay isy_150ms

Chapter 8 Conclusions and Future Work

8.1 Thesis Summary

Programming in LabView was done and using Profibus protocol, communication between two (2) distinct, commercially available systems; CRio and Vacon, was established to command and monitor the PMSM.

LabView program was also used to control the system, comprising of Diesel engine operation, transmission coupling to engine and ExSAM, load management [Note 4], monitoring of essential parameters of the ExSAM, and the remaining heavy hybrid truck platform test bench.

The speed control was not effective due to delay in communication speed between the two systems.

8.2 Discussion

The speed controller did not work properly due to a lack of communication speed between the two systems.

Vacon has provided some feedback for;

1. Use of a software tool available in 'NC Drive', called 'Data Logger' to monitor the delay in received command (from LabView) to acknowledgment at the Vacon's end.
2. Changing and compiling the 'Application File' again.

There is also a possibility that the 'Delay Times', as shown in chapter 7, may be showing the delay between 'Referenced Currents' and 'Monitoring value of Currents'. This means that Vacon could be implementing the 'Currents' faster and updating the 'Monitoring Value' later, being lower priority than the actual implementation.

This hypothesis will be checked by not only using the Vacon's 'Data Logger' as described above, but through current measuring device coupled to the CRio.

The "Fast" Profibus mode was tried in lab 6 to run the Haldex machine after the "Normal" Profibus did not yield the desired results, as explained in section 6.3.

However, even after repeated trials, the machine did not rotate. Assuming that it must be a resolver calibration issue [13], PhD student Francisco Marquez's help was sought. It turned out that with the "Fast" mode activated, the slots "B" and "C" in the Vacon CU could not be used [14]. There are five slots in vacon CU and resolver card was using slot "C". It is possible that with "Fast" Profibus option, the Vacon CU requires a different kind of resolver card, which might be compatible with it.

Then the “Bypass” Profibus mode was tried on the Haldex machine and it worked. However the issue of communication delays, although reduced to max of 150ms from 180ms (Normal Mode), still remains to be addressed in collaboration with Vacon.

The frequency converter has recently been fixed by Vacon, after the incident described under section 6.7. A team from IEA has re-commissioned the Vacon converter and tested the ExSAM during the days of July 2nd to 4th 2012.

The communication between the two systems; CRio and Vacon, was also tested with the frequency converter off. This was done to test if the switching of the transistors might be affecting the speed of communication by forcing the data go back and forth on the connecting Profibus cable, before a clean signal is received and acknowledged. With frequency converter powered off, the current reference of isy was not acknowledged by Vacon. However, isx was acknowledged with the same delay.

Vacon’s ‘Data Logger’ was also tested with results presented in *figure 21* below.

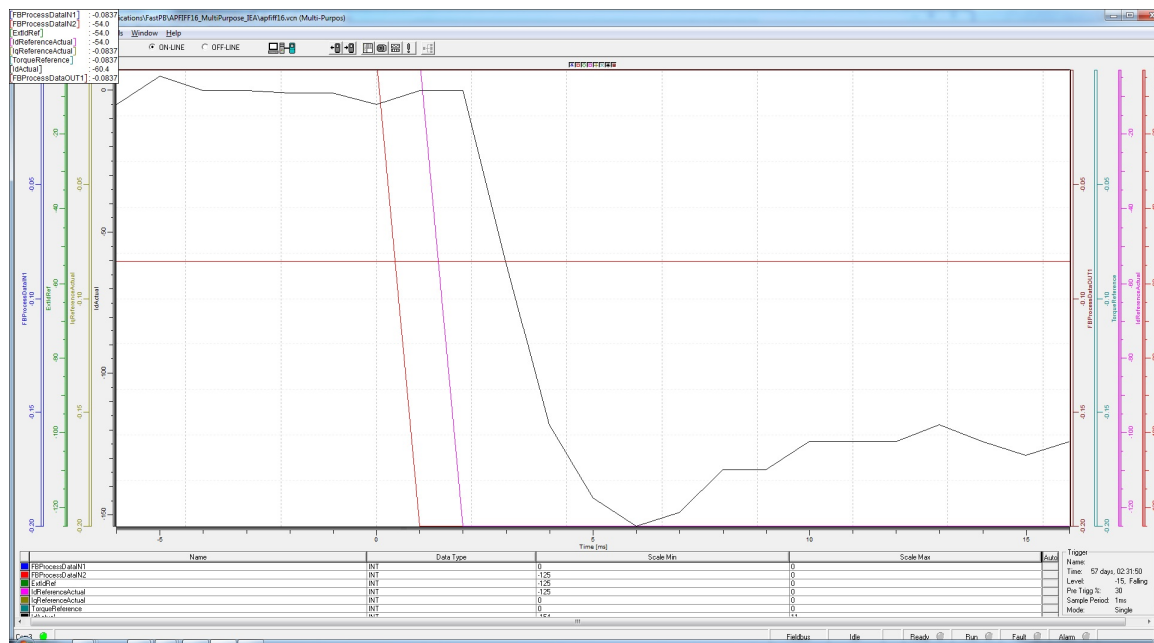


Figure 21 – Vacon’s ‘Data Logger’ Results

According to this figure, signal is received from CRio at zero millisecond, the leftmost vertical/inclined line. This signal is acknowledged in one millisecond afterwards by Vacon and implemented after one additional millisecond.

The Profibus loop in LabView is running at microsecond time scale and Vacon control unit as per figure 21 above, is running on 1-2 ms scale. However, the delay of 112 ms persists.

It could be that Vacon is counting time not from when it has received the signal, rather when it has confirmed it after some internal processing. Figure 13 on page 39 shows a maximum delay of 110 ms (Normal/Slow Profibus), not counting the processing of “Service Data”.

Following figure 22 shows the delay as recorded by LabView [Note 5]

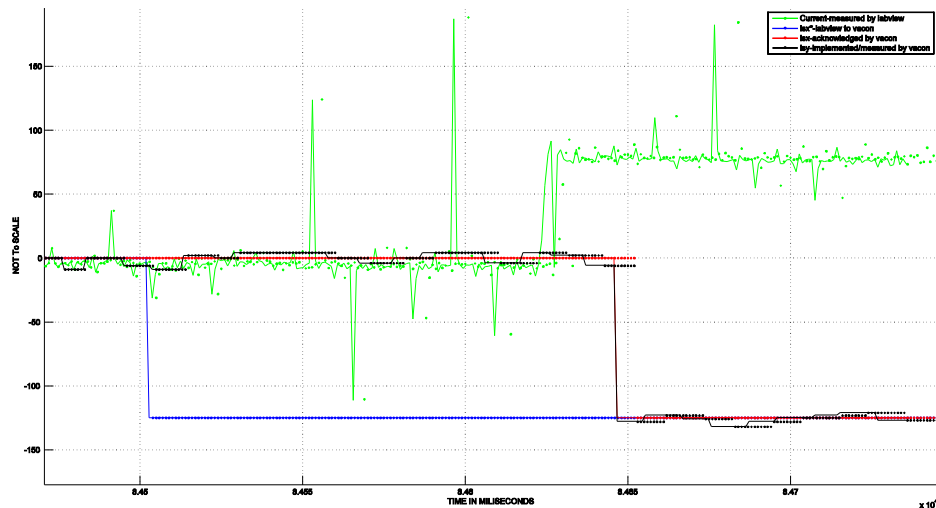


Figure 22 – Delay Results from Vacon and independent current measuring (Bypass Profibus Mode)

The delay in the signal sent by LabView and acknowledged by Vacon is 150 ms (only maximum delays are taken in all of the figures). The green line in figure 22 above is the measured current change, meaning that the current is in fact applied after 130ms and the monitoring value is updated later making a total of 145 ms [Note 6].

The experiments were repeated with some changes in cables and software [Appendix A] and according to the results the most time delay in “Normal” mode is 140 ms in acknowledgement and 110ms in implementation [Note 7]

The delay in “Bypass” mode is 152 ms in acknowledgement and 118ms in implementation.

The delay in “Fast” mode is 117ms in acknowledgement and 112 ms in implementation. Due to resolver card conflict, the machine could not be operated in “Fast” mode.

Since the PID controller takes the motor rpm as an input to manipulate the currents and the ‘implemented’ current changes the motor rpm, the delay as seen by the PID controller is the ‘Implemented Delay’ and not the delay in ‘Acknowledging/Confirming’ the current command.

8.3 Future Work

Work on the communication speed issue resolution is ongoing, with 'Data Logger' and direct current measurement through CRio continuing.

Three main areas of focus for the future work are listed below;

1. Continuing work on 'Fast' Profibus
2. Implementation of 'Look Up' tables for automated field weakening current injection.
3. LabView communication with sequential gearbox [15].

Vacon is being contacted to elaborate on the delay as discussed on page 46, paragraph 1.

NI is also being contacted to determine if there might be any delay in internal processing of the signal in the Profibus module, before it actually sends out the signal/control command.

It will be evaluated if something similar to "CAN Analyzer" exists for Profibus. Hooking up the Profibus signal to Oscilloscope along with Vacon's feedback will also be tested to synchronize the two distinct time scales and thereby ascertaining the source of the time delay. Approach with Oscilloscope may not work since if there is always a digital communication between the two systems regardless of the command signal, it will not be distinguishable on the scope.

Currently only two out of eight possible Process Data [PD] commands are being used; pd1_in for Isy* and pd2_in for Isx* [Note 8]. These are the only two values manipulated by LabView to send 'Current References Commands', whether in 'Current Control' OR 'Speed Control' mode. Vacon's NC Drive has the "Application File" describing all the parameters of the electric machine, including the maximum and minimum allowable currents for both 'isx' and 'isy'. The delay in actual implementation of the current command could be due to Vacon checking the received current command against the allowed currents that can be applied and then implementing and acknowledging it.

This hypothesis will be checked by configuring one or few of the remaining six pd_in (LabView to Vacon) commands, asking Vacon to provide information on e.g. 'Motor Speed', 'DC Link Voltage' etc.; parameters, which Vacon only has to provide information about without having to check the validity of the command against a data base.

NOTES

NOTE 1: It is important to 'Disconnect' the Comsoft connection before running the LabView program. If the LabView program is run without first disconnecting the Comsoft connection, the LabView environment will Not have any access; Controlling OR Monitoring of the Electric Machine and the associated system.

NOTE 2: This is important because otherwise NC Drive will not be able to take the control back from FieldBus and once executed, will run with the last current values I_{sy}^* and I_{sx}^* as was set in LabView without allowing the user to change or reset the values.

NOTE 3: After the modification to achieve 'Fast Profibus', section 6.8, assigning and reassigning the PD values were not performed.

NOTE 4: Although the LabView program for loading was in place and functional, it was not used since the flywheel at the time needed to be balanced.

NOTE 5: Current sensor was placed on one of the three phases to determine the change in current, corresponding to the signal/current reference going to Vacon. This reference was captured by an analog input module in Crio. Y scale on the figure is not important and applicable, only the time scale is of concern.

The delay as shown in figure 22 is for "Bypass Profibus", whereas the delay in figure 13 is for "Normal/Slow Profibus".

NOTE 6: Although figure 22 shows a delay of 145 ms and not 150 ms, the value of 150 ms is selected to report the maximum delay in "Bypass Profibus Mode". This is due to the fact that in this mode the extensive testing and data analysis resulted in 150 ms of observed delay. The measurement with the current sensing device to detect the actual implementation (as shown in figure 22) is performed recently and it has currently limited amount of data compared to the data available in "Bypass Profibus Mode", without the sensing device.

NOTE 7: The data presented in 'Appendix A' was recently acquired after installing new cables for LabView current measurement and reinstalling the 'Application File' in NC Drive. The loop timing of LabView was also reduced to make it more compatible with Vacon software. There is some difference in values from the data that was acquired earlier; however this could be due to the fact that the new data in amount is less compared to the earlier acquired data.

NOTE 8: In contrast to using only two `pd_in` (LabView to Vacon), all of the available eight `pd_out` (Vacon to LabView); `pd1_out`....`pd8_out` are being used.

APPENDIX A: PROFIBUS MODE DELAYS

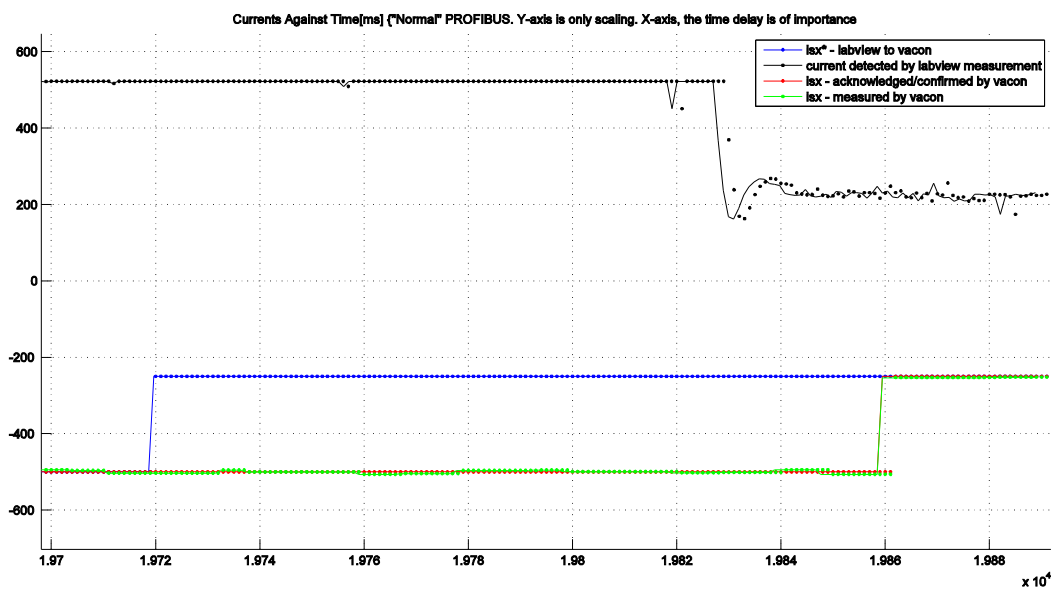


Figure 1_Appendix_A: Normal Profibus Delay

The delay here is 140ms in acknowledgement/confirmation, while it is 110ms in implementation [Note 7].

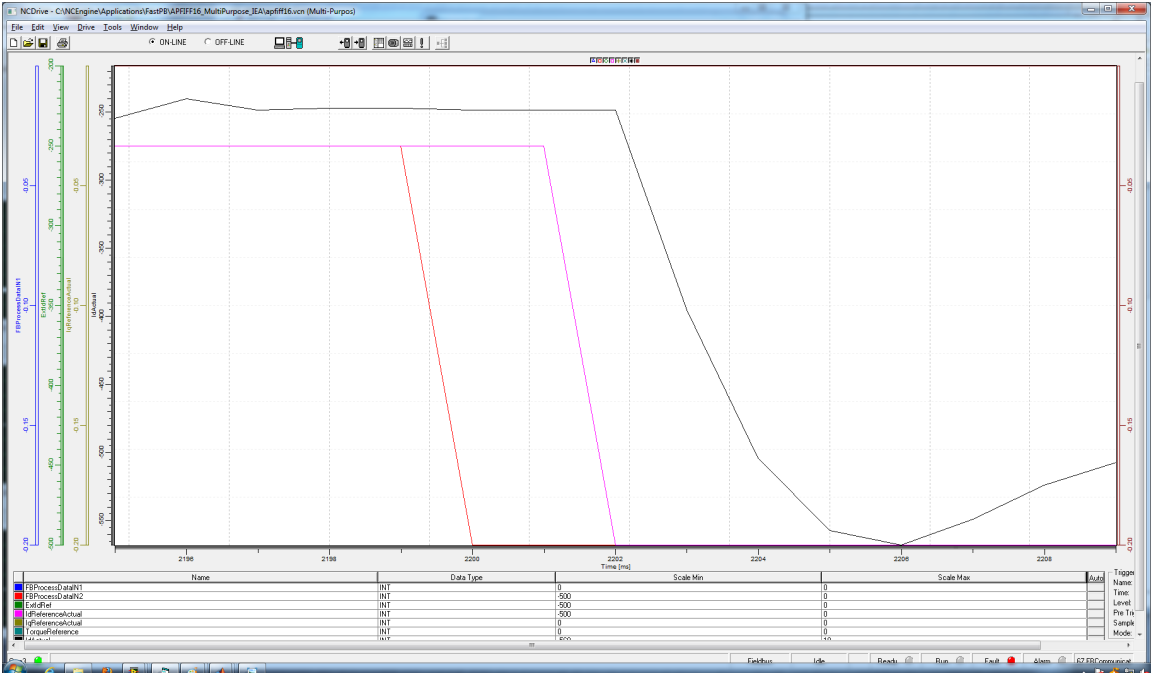


Figure 2_Appendix_A: Normal Profibus Delay from Vacon's 'Data Logger'

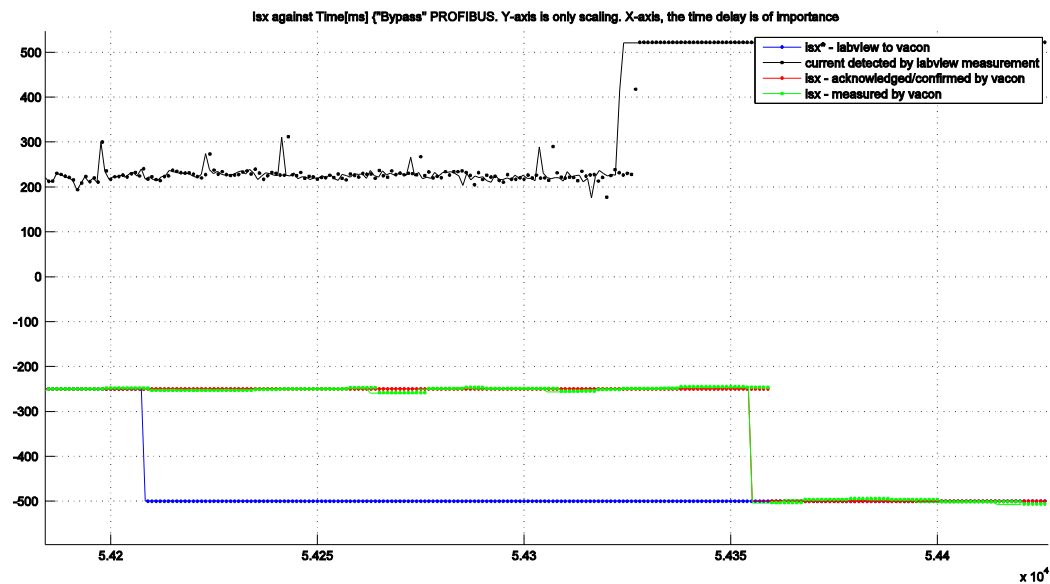


Figure 3_Appendix_A: Bypass Profibus Delay

The delay is ~ 152ms in acknowledgement/confirmation, and 118ms in implementation of the current command [Note 7].

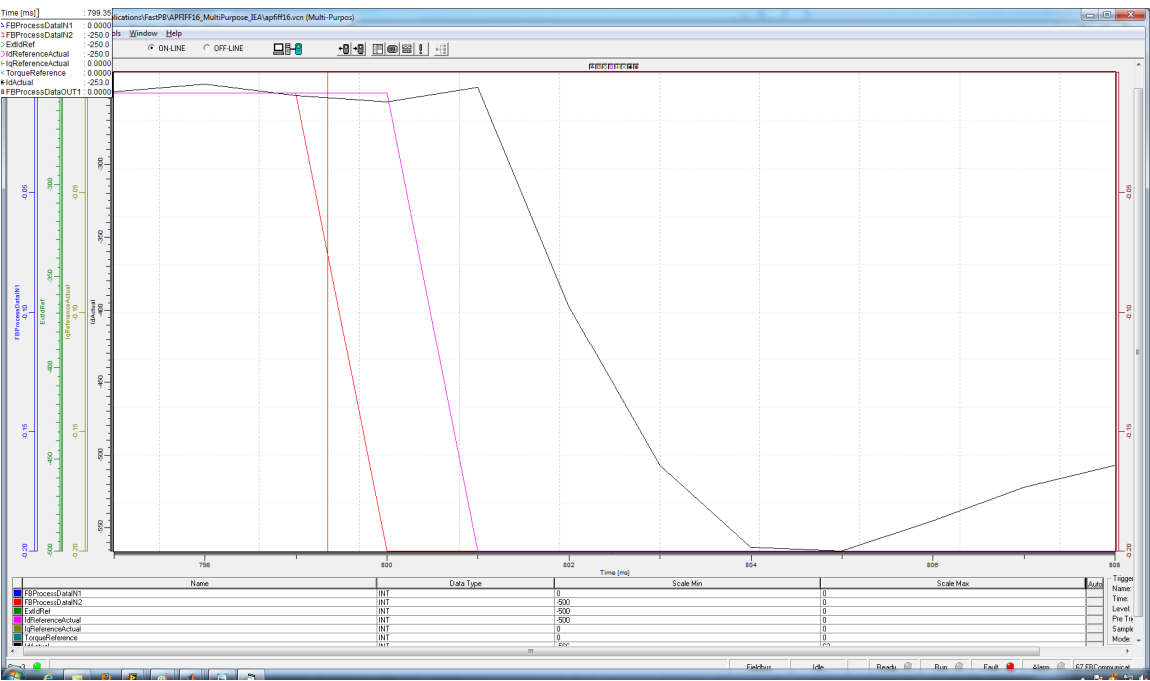


Figure 4_Appendix_A: Bypass Profibus Delay from Vacon's 'Data Logger'

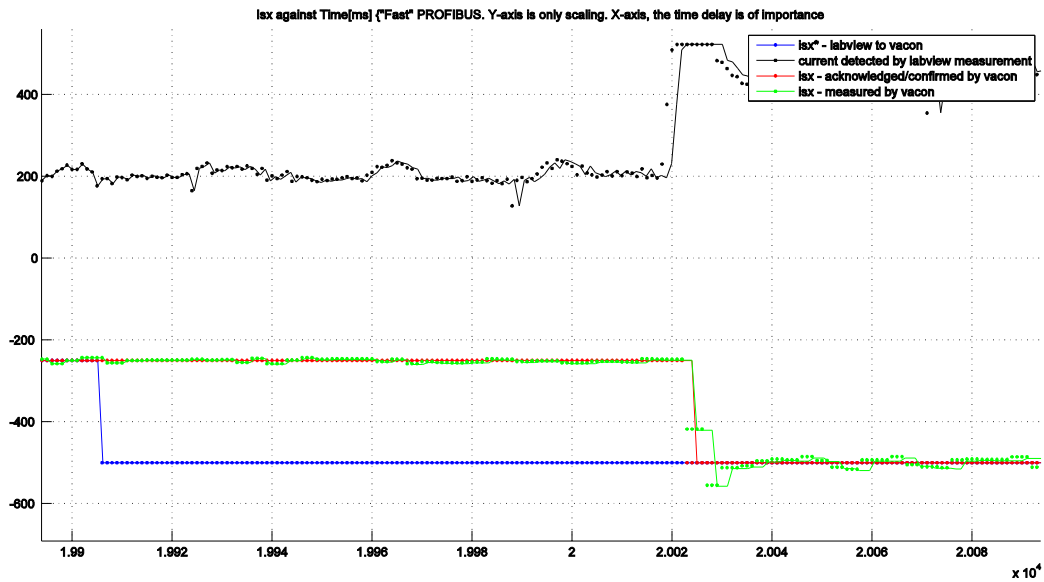


Figure 5_Appendix_A: Fast Profibus Delay

The delay is 117ms in acknowledgement/confirmation, and 112ms in implementation of the current command [Note 7].

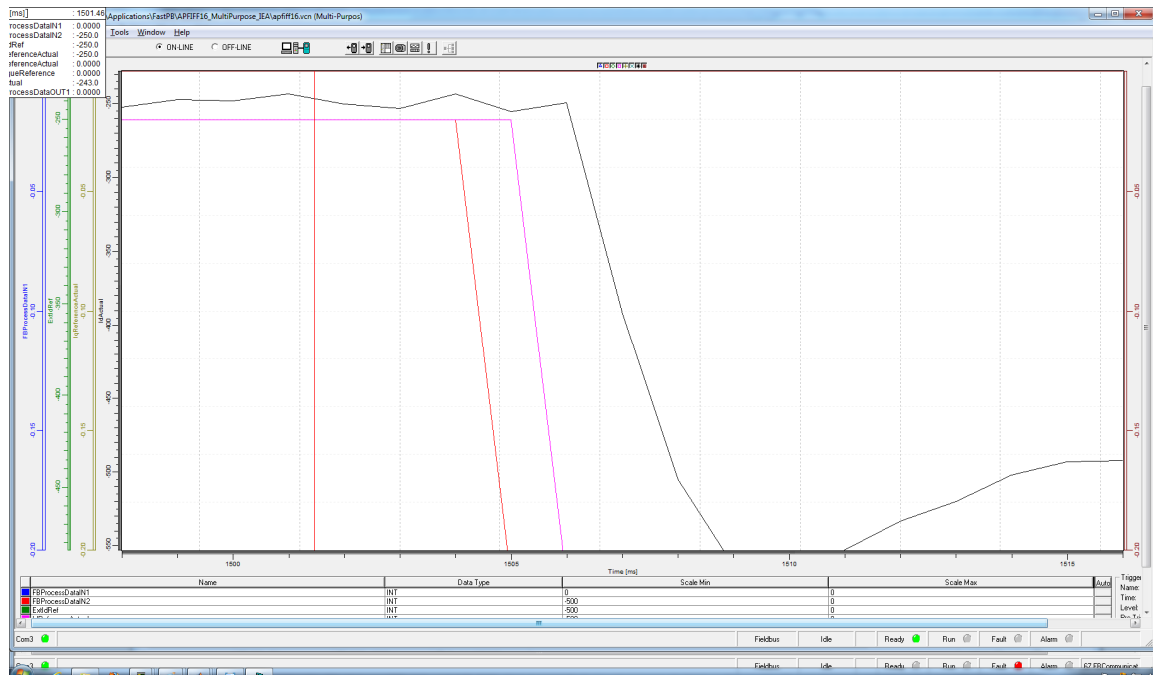


Figure 6_Appendix_A: Fast Profibus Delay from Vacon's 'Data Logger'

As per the information from Vacon's 'Data Logger', the delay in "Normal" Profibus mode is 2 to 3 milliseconds (*Figure 2_Appendix_A*) in confirmation and implementation. This delay is further reduced to 1-2 milliseconds in "Bypass" (*Figure 4_Appendix_A*) and "Fast" (*Figure 6_Appendix_A*) Profibus modes.

Vacon and NI, both are contacted for the resolution of the delay of over 100 ms as described in section 8.2. This issue will most likely be concluded after the summer break, once the concerned personnel at Vacon and NI are available.

REFERENCES

1. Vacon NX Profibus DP
2. Vacon NX Fieldbuses
3. Getting Started with the CompactRIO PROFIBUS Module
4. CompactRIO PROFIBUS DP - DP Master – Getting Started
5. CompactRIO PROFIBUS DP – Installation Instruction
6. NX Frequency Converters – Profibus DP Option Board
7. Profibus Overview by National Instruments
8. Profibus Interfaces by National Instruments
9. Programing at Vacon side and an explanation of its software/workings was done/explained by PhD student Francisco Marquez.
10. All of the data storage sequence and programming, including the MatLab code for data analysis, was done by PhD student Yury Loayza.
11. Commissioning of Vacon frequency converter at Sibbhultsverken was done by Avo Reinap, Rasmus Andersson and Anders Goransson.
12. Multi-Purpose Control Application - Vacon. *This application was modified by Francisco for Haldex machine application.*
13. OPT BC Resolver Option Board - Vacon.
14. "Fast" Profibus DP – Vacon.
15. MSc Thesis: Automated shifting of a manual sequential transmission in a hybrid vehicle by Axel Bergman and Per Byrhult, IEA Lund University.