Hardware-in-the-loop Simulation for Pump Control

Confidential until June 2020

by

Ørjan Bø

May 19, 2015

Supervisors:
Anders Meisfjordskar
Ilya Tyapin

This Master's Thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or conclusions that are drawn.

University of Agder, 2015
Faculty of Technology and Science
Department of Engineering

Abstract

National Oilwell Varco use pump controller cards from Bosch Rexroth in order to control variable displacement pumps used for crane operations. This thesis investigates the use of Bosch Rexroth pump controller card as part of a Hardware-In-the-Loop simulation system. Expected result is a Hardware-In-the-Loop system rig to be used as a platform for educational training, experimenting and testing in-house. The Hardware-In-the-Loop system is realized by the use of a CompactRIO real-time controller and VT-VPCD pump controller card. The system is configured and modelled in LabVIEW. Mathematical models for the hydraulic components in the control system are derived using an analytical approach. A dynamic mathematical simulation model of a hydraulic leakage circuit is set up. State variables in the dynamic simulation model are used to create model feedback for the closed loop control system.

The Hardware-In-the-Loop system shown to be applicable for pump tuning operations with a dynamic mathematical simulation model. An execution rate at 2 ms resulted in a central processing unit load at ~76% for the CompactRIO. Lowering the execution rate beyond 2 ms resulted in central processing unit overload and non-deterministic operation.

An external real-time target was implemented to the system in order to increase the computational power. An execution rate at 1 ms resulted in an average of 2% central processing unit load for each core, on a quad core processor, with the same dynamic simulation model executed on the CompactRIO. Increased computational power causes more complex models to be executed real-time.

The Hardware-In-the-Loop system with the CompactRIO is a portable solution applicable for training and educational operations. Complex simulation models causes the CompactRIOs central processing unit to overload. An external real-time target increases the computational power and opens the possibility to execute complex real-time simulation models at a higher rate compared to the CompactRIO.

Preface

This dissertation is made as a completion of the Master of Science education in Mechatronics at the Faculty of Technology and Science, at University of Agder. The master thesis is carried out in collaboration with National Oilwell Varcos (NOVs) Crane & Winch Hydraulics department in Kristiansand, Norway.

I would like to thank supervisor Anders Meisfjordskar, from NOV, for valuable ideas and comments throughout the work period. Thanks to associate Professor Ilya Tyapin for feedback and comments regarding the completion of this dissertation.

Finally, I would like to thank my loved one, family and friends for being helpful and supportive during my time at University of Agder.

Grimstad
May 19, 2015

Ørjan Bø

List of Figures

List of Tables

Abbreviations

cDAQ	= Compact Data Acquisition
CPU	= Central Processing Unit
cRIO	= CompactRIO
DLL	= Dynamic Link Library
EPROM	= Erasable Programmable Read-Only Memory
FIFO	= First In First Out
FTP	= File Transfer Protocol
FPGA	= Field Programmable Gate Array
HIL	= Hardware-In-the-Loop
GCC	= GNU Compiler Collection
GUI	= Graphical User Interface
LabVIEW	= Laboratory Virtual Instrument Engineering Workbench
LVDT	= Linear Variable Differential Transformer
MBD	= Model Based Design
NI	= National Instruments
NOV	= National Oilwell Varco
NOVN	= National Oilwell Varco Norway
PCI	= Peripheral Component Interconnect
PPQ	= Pressure, Power and Flow
PROFIBUS	= Process Field Bus
PWM	= Pulse Width Modulation
PXI	= PCI eXtensions for Instrumentation
RIO	= Reconfigurable Input/Output
RT	= Real-Time
RTOS	= Real-Time Operating System
SVE	= Shared Variable Engine
TCP/IP	= Transmission Control Protocol/Internet Protocol
USB	= Universal Serial Bus
VI	= Virtual Instrument

Contents

Chapter 1

Introduction

The crane and winch hydraulic department at National Oilwell Varco Norway (NOVN) uses pump controller cards from Bosch Rexroth for all their axial piston pumps. NOVN wants to develop a Hardware-in-the-loop (HIL) simulation system for testing and experimenting with the pump controller card. The problem, purpose and previous work done for the project is described in the following sections.

1.1 Motivation

Today, HIL systems are widely used when developing, testing and experimenting with complex systems. If one or more components in the simulation model are removed from the model and the physical components are used as a hardware in the simulation loop, one gets the exact physical response from the components. NOVN uses pump controller cards from Bosch Rexroth and wishes to create a HIL system with a pump card as a hardware in the loop. The system is to be used for educational purposes, pre-tuning and for experiments where test scenarios for known challenges related to the pump systems are simulated.

1.2 Problem Statement

NOV uses variable displacement pumps to control flow, pressure and power in their hydraulically actuated systems. The control is handled by a pump controller card (Bosch Rexroth VT VPCD-series) that receives the reference value and process value on flow, pressure or power from a PLC and returns a control signal to a directional control valve in order to control the pumps swashplate. The control is tuned by configuring the pump card settings. This is normally done on site. NOV wants a HIL system where training and pre-adjustments can be done in-house.

Six subsidiary goals are stated together with NOV. These are:

1. Build a test rig with power supply, compactRIO (cRIO) and Euro-card holder with a pump controller card.
2. Wire up the electrical interface between the pump controller card and cRIO.
3. Create mathematical models in terms of transfer function models for the hydraulic control system with proportional valve and axial piston controlled swash plate. Determine dynamic properties from experimental data or data sheets and implement the dynamics in LabVIEW.
4. Set up a dynamic simulation model of a hydraulic leakage circuit so that pressures for A- and B side of the pump are generated.
5. Create a graphical user interface (GUI) for users and operators of the HIL system. The dynamic simulation model should have the possibility to be configured from the GUI.
6. **If time allows:** Implement a SimulationX model to generate typical fault scenarios related to pump control.

1.3 Key Assumptions and Limitations

The components and equipment needed to complete the project is provided by NOV and University of Agder. A cRIO from University of Agder is available throughout the project period. All remaining required equipment is provided by NOV.

1.4 Problem Solution

Test Rig
The components and equipment used in the HIL system is to be mounted on a plate with the possibility to be fitted inside a suitcase, or similar, to get a portable solution. 230VAC, USB connection and TCP/IP communication to a host PC with LabVIEW software is required to operate HIL system.

Electrical Interface Between Pump Card and PC
The relevant signals are wired from the pump card to the I/O modules connected to a cRIO. All analogue signals are $\pm 10\,V$ or 0-10 V. Scaling and signal processing is executed on the cRIO.

Transfer Functions for Hydraulic Control System
The hydraulic pump control, thereunder proportional valve and axial piston cylinder, should be modelled as transfer functions in order to be able to simulate simple load scenarios. The transfer functions for the proportional valve and axial piston cylinder are modelled based on frequency response and step response information from data sheets.

Dynamic Simulation of Test Circuit
NOV uses a standard test circuit for calibrating valve, axial piston cylinder and sensors used in the hydraulic control circuit. By setting up a mathematical model describing this test circuit it is possible to generate pressures for A- and B-side of the pump. NOV wants to control parameters related to the hydraulic circuit from a GUI.

Create a Graphical User Interface
The GUI can be created in LabVIEW. Detailed monitoring of the control and hydraulic system is done from Bosch Rexroths program Bodac. The interface in LabVIEW must include configuration parameters for the dynamic simulation.

Simulate Typical Fault Scenarios with SimulationX Model
When the HIL system with accurate simulation models for the pump, proportional valve and feedback sensors is working, advanced dynamic simulation models exported from SimulationX can be implemented. This to simulate more advanced load cases with control options from the GUI at the host PC.

1.5 Report Outline

Chapter 2 contains a theoretical background for the components used in the HIL system. The working principles of the hydraulic control circuit and electronic control card is explained.

Chapter 3 presents the solutions of the problems in section 1.2. First, various HIL applications are considered on the basis of costs and system performances. The appropriate HIL system is chosen and the process of constructing a rig and implementing all the features required to run a HIL simulation is described. Model estimation for various pumps and model implementation of hydraulic circuits are done to simulate a physical plant. Last, the different models are verified and tested against real-life log values.

Chapter 4 is where the various solutions related to the HIL system are discussed. The HIL system performances are considered in terms of central processing unit (CPU) load and execution rate for the RT target. The HIL system can be applicable for more than pump card simulation. New application areas and improvements of the HIL system are also presented in the chapter.

Chapter 5 contains the conclusion. The problems stated in section 1.2 are presented by its respective solutions.

Chapter 2

Hardware-In-the-Loop System Overview

A hardware-in-the-loop system combines real hardware components with software-based simulation. In this chapter the principles behind HIL simulation are explained. General theoretical background for HIL components are presented before more specific theory related to the actual HIL system is presented.

HIL simulation is widely used for developing and testing of complex systems. By representing the system components and dynamics with a mathematical model, the plant can be simulated on a real-time controller. HIL simulation is a cost-effective and efficient way to develop systems for a wide range of engineering problems. One of the main advantages of a HIL simulation is that design and testing of the control system can be done parallel with building of the actual plant. In other words, HIL simulation is more efficient in terms of costs, duration, safety and feasibility [1]. HIL simulation is not just used when developing and building new plants. In addition, it can be used for testing of various components of a plant in order to optimize the process for each component by performing rapid testing through HIL simulation. In the following sections the theoretical background for the components and software used for HIL simulation are described.

2.1 Real-Time Simulation

In the last 20 years computers have become increasingly powerful and affordable. This, in turn, has led to the emergence of highly sophisticated simulation software applications that not only enable high-fidelity simulation of dynamic systems and related controls, but also automatic code generation for implementation in industrial controllers [2].

In real-time simulation, events occur on a natural time scale. Such real-time behaviour has become important for a board spectrum of applications embedded system design, control of dynamic processes, streaming data acquisition and operator training. As a consequence, it is critically important to preserve the characteristics of the system dynamics [3]. The advances in computational power yields for increased real-time simulation capability in terms of simulation model complexity and lower execution rate. Real-time HIL simulation together with Model-Based Design (MBD) opens endless opportunities for engineers in terms of developing simulation models in parallel with the actual making of the product. MBD is a process that enables fast and cost-effective development of dynamic systems, including control systems, signal processing and communications systems. MBD has become commercially available through simulation software like MATLAB Simulink® and Modelica based software like ITIsim SimulationX.

2.2 CompactRIO

A CompactRIO is a rugged real-time controller for industrial applications. The cRIOs hardware architecture includes I/O modules, a reconfigurable field programmable gate array (FPGA) chassis, and an embedded controller. Communication between the cRIO and a host computer can be done through Ethernet, RS232 serial and universal serial bus (USB) communication as the most common options. Other communication platforms, like process field bus (PROFIBUS), are also available.

The FPGA module is a high-performance data processing unit used for I/O processing from the I/O modules connected to the cRIO. The FPGA target digital architecture consists of three main components: an array of logic blocks, called Configurable Logic Blocks, I/O pads and programmable interconnects. The FPGA module, a high-performance Virtex chip from Xilinx, is connected to the real-time target through an internal peripheral component interconnect (PCI) bus. Figure 2.1 shows the hardware architecture of a typical FPGA module [4].

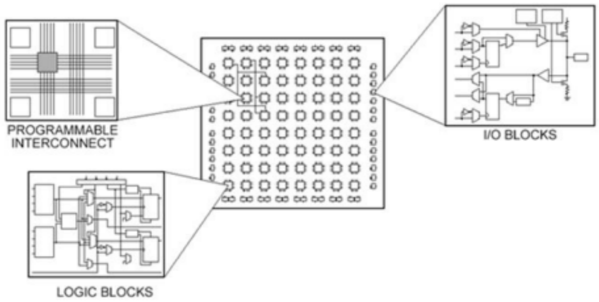


Figure 2.1: FPGA overview with available components for programming [5]

In theory, the FPGA can solve any problem which is computable. FPGAs advantages lies in that they are sometimes significantly faster for some applications due to their parallel nature and optimality in terms of the number of gates used for a certain process. Their disadvantages are related to programming time, i.e time consuming when performing updates or debugging the software.

The cRIO Real-Time Controller includes a processor along with the reconfigurable FPGA target. The controller opens the possibility for real-time deterministic processing, data logging, control and network communication [6]. An overview of the CompactRIOs archtecture is shown in Figure 2.2.

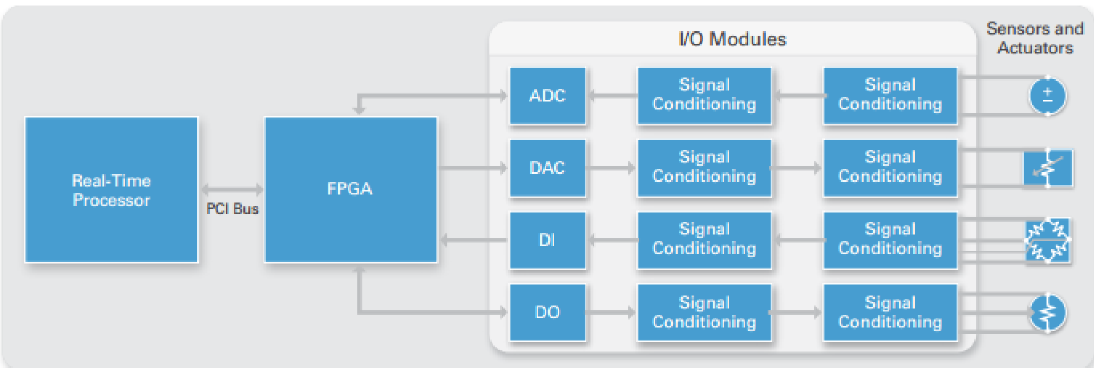


Figure 2.2: CompactRIO architecture [7]

The advantage of using a real-time controller is that high priority tasks are executed deterministically on or before time, seen that the CPU on the real-time target is not overloaded. In comparison to general purpose operating systems, like Windows, the user will experience high jitter and the user can not be sure if the application is deterministic, hence reduced reliability for critical tasks. Determinism describes how the system responds to external events or performs operations within a given time period. Figure 2.3 shows an example of a task executing outside of the desired execution time, causing jitter, and one task execution within desired execution time.

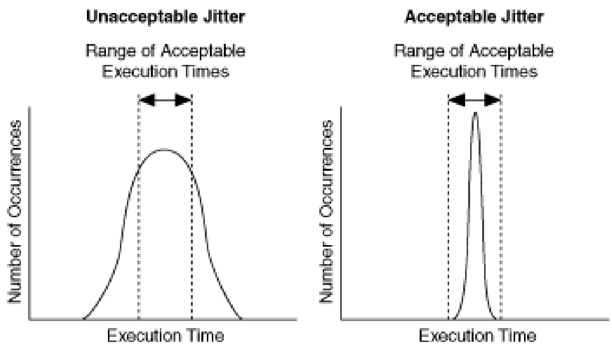


Figure 2.3: Illustration of acceptable and unacceptable jitter [8]

2.3 LabVIEW Programming Interface

LabVIEW, which is short for Laboratory Virtual Instrument Engineering Workbench, is a graphical programming environment used by engineers and scientists to develop control systems using graphical icons and wires that resemble a flowchart. The LabVIEW platform is scalable across multiple targets and OSs, and, in the case of compactRIO, LabVIEW can be used to access and integrate all of the components of the LabVIEW reconfigurable I/O architecture [7]. LabVIEW is used for a wide variety of engineering applications, like data acquisition, instrument control and industrial automation.

In a HIL system a LabVIEW project file gathers the different virtual instruments (VIs) under a project tree which includes all targets connected to the host PC through an interface, usually TCP/IP or USB interface. Figure 2.4 shows an example of a LabVIEW project with a cRIO connected via TCP/IP.

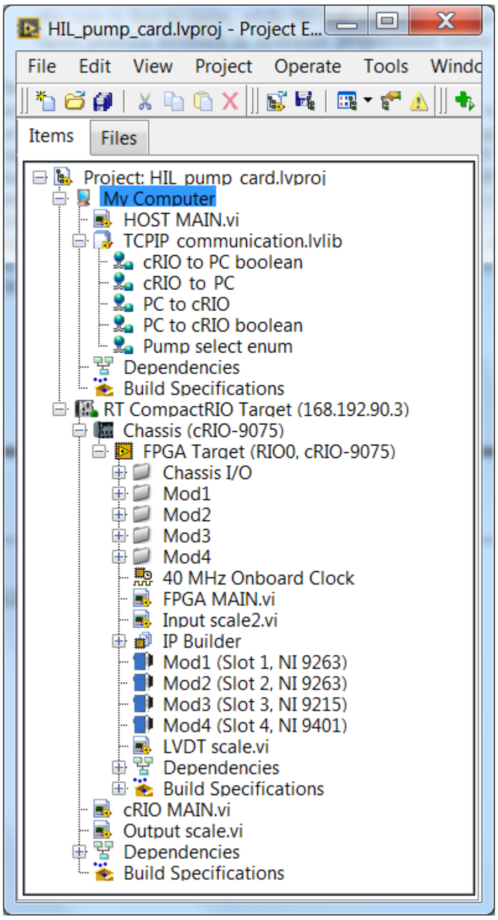


Figure 2.4: Example of LabVIEW project tree with a compactRIO with four I/O modules

LabVIEW is a module based program, with a wide variety of module packages that can be installed, depending on area of application. For HIL application the modules listed below are typically required create a HIL system.

- LabVIEW Real-Time Module
- LabVIEW Control Design & Simulation Module
- LabVIEW FPGA Module
- Shared Variable Engine

Priority-Based Execution
The LabVIEW Real-Time Module handles priorities and execution systems when creating and scheduling threads. Operation within a thread execute sequentially, while each thread executes independently. The scheduler arranges threads and execution systems in a run queue based on the priority for each thread. Figure 2.5 visualizes the different priorities which can be assigned to execution systems on a RT controller.

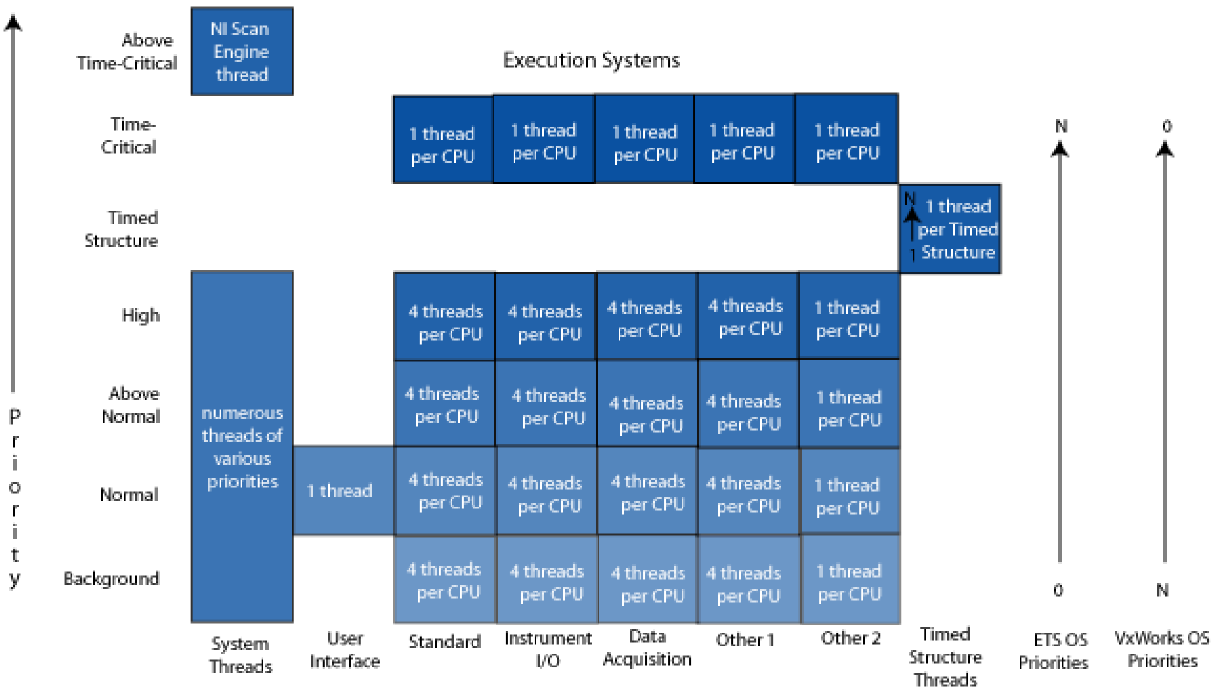


Figure 2.5: Execution of LabVIEW VIs based on priority [9]

In order to optimize the execution on the RT target LabVIEW creates up to four threads at each priority level. This rule extends to each core in a multi-core system. Because the number of threads per execution system is limited, balancing the execution system assignments can be important. If too many VIs are assigned to the same execution system, the VIs must share threads, limiting the potential parallelism of the VIs. In order to prioritise VIs and execution loops different priorities are assigned dependant on whether the task is deterministic or non-deterministic. Examples of deterministic and non-deterministic task are shown in Table 2.1.

Table 2.1: Deterministic and non-deterministic tasks

Deterministic tasks	Non-deterministic tasks
Closed-loop control	File I/O
Decision-making logic	Network or serial communication
Safety logic	Tasks involving large memory allocations
FPGA or RIO Scan Interface	Calls to non-deterministic libraries or drives

The threads in a run queue can have two different states; blocked or running. Running mean that the thread is either in the running queue or is currently executing. Blocked means that the thread can not execute until some event occurs, such as the completion of an I/O operation or a timing function. When a thread is executing it runs until one of the following events arises [9]:

- Thread becomes blocked by a Wait VI within a While Loop or the built-in timing mechanism of a Timed Loop.
- Thread is interrupted by a higher priority thread.
- Thread finishes executing.

When setting loop priorities the priority should be set based on how important it is that the task fulfil a certain timing guarantee. The VI priority is set under the "Execution" tab for VI properties. Normally, the priorities for the tasks are set by using timed loops for deterministic tasks. Timed loops have additional features for executing code deterministically, compared to while loops. Seen in relation to Figure 2.5, a timed loop executes above high priority and below time-critical priority.

Variables
Communicating between loops, VIs and network targets require different variable settings. There are several variable types to consider while setting up the real-time application. These are; local variables, global variables and shared variables. Shared variables can be configured as both single-process and network-published variables.

If local or global variables are used on a RT target these variables must be used carefully. Race conditions can occur when two or more pieces of code execute in parallel and have access to a shared piece of memory. A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly. If each piece of code is independent, there is no way to distinguish the order LabVIEW uses to access the shared resource. Overusing local and global variables, such as using them to avoid long wires across the block diagram or using them instead of data flow, slows performance [10].

Single-process shared variables is to be used for tasks locally on a target. For deterministic operations, the shared variables can be configured with a real-time first-in first-out (RT FIFO) option. Enabling RT FIFO is recommended for streaming data between any two processes on a real-time operating system (RTOS) because they always preallocate memory and have a maximum buffer size. RT FIFOs are recommended for transferring commands and messages to or from a time-critical loop.

Using network-published shared variables is a method for sharing tags across a network. Network-published shared variables are hosted by the Shared Variable Engine (SVE). SVE is a software component that hosts data published over Ethernet. The engine can run on real-time targets or Windows PCs. To use network variables, the SVE must be running on at least one of the systems on the network. If time-critical tasks runs over Ethernet, the network-published shared variables can be configured for RT FIFO, making the communication process deterministic. When RT FIFO is enabled, LabVIEW automatically runs a background loop to copy the network data into RT FIFO. This prevents jitter from occurring within the time-critical loop while performing network communication, but it does not mean that the network communication itself is deterministic [7].

2.4 Electro-Hydraulic Pump Control

Electronic pump controller cards are used for pressure, power and flow (PPQ) control for variable displacement pumps. In larger hydraulic systems pump cards can be used in a master-slave configuration for easily expansion of the hydraulic pump stack configuration.

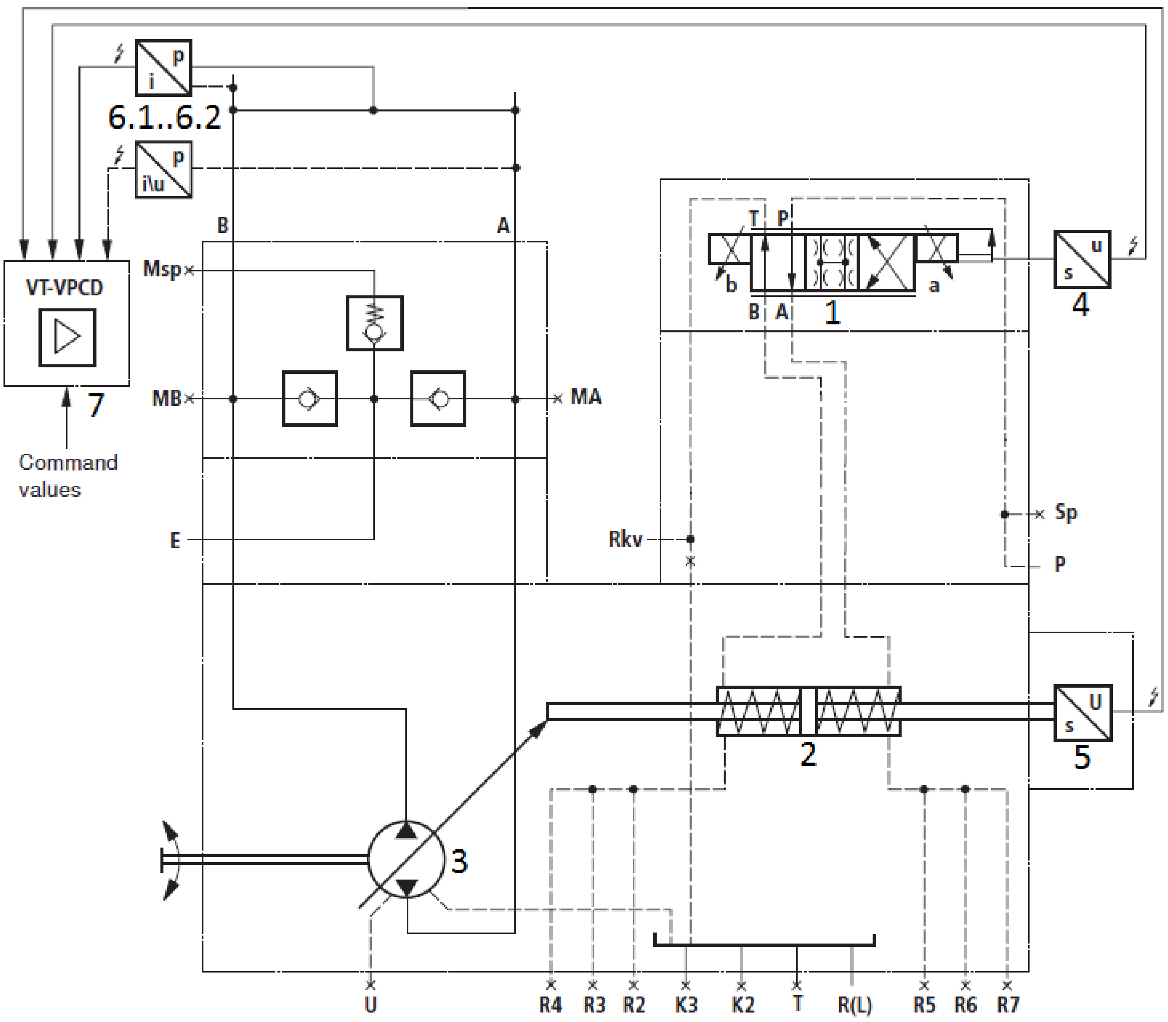
Bosch Rexroth is one of the world leading manufacturers in hydraulic equipment and control electronics for hydraulic components. The VT-VPCD pump controller card is used for electro-hydraulic control of the swivel angle and pressure as well as for limiting the power of variable displacement axial piston pumps. VT-VPCD has features like:

- Pressure controller with subordinate swivel angle controller
- Mooring-capability
- Master-slave-capability
- Leakage compensation
- Diagnostics
- Optional field bus system: Profibus DPV0 (for cyclic exchange of data and diagnosis)

Bosch Rexroth has developed a computer program called Bodac for configuration of their pump controller cards. Analogue I/O, valve settings, sensor settings, tuning parameters and more can be configured in Bodac. Hence, a wide variety of configurations can be chosen to customize the electro-hydraulic control system for the application. When the electro-hydraulic control system is set up Bodac is used to optimize the system by means of PPQ control and response to changes in control inputs.

Functional Description
An electrically controlled proportional valve controls the swivel angle, pressure and limits the power of the A4VS[1] variable displacement pump. This valve determines the position of swashplate through an actuating piston for swashplate control. When the pump is not rotating, the actuating systems are pressureless and the enable is not activated. The swashplate is held by the spring centring feature in the "zero" swivel angle position. The position of the swashplate is established by inductive position transducer, the actual pressure value by a pressure transducer. Both actual values are fed to the VT-VPCD control electronics and interlinked by the software. Figure 2.6 shows the hydraulic control circuit for closed loop applications along with sensor feedback signals [11].

[1] Bosch Rexroth delivers A4VS pumps in sizes from 40 → 1000 ccm



1	Proportional valve
2	Axial piston cylinder
3	A4VSG pump
4	Position transducer valve
5	Position transducer cylinder
6.1..6.2	Pressure sensors for A- and B-side
7	VT-VPCD pump controller card

Figure 2.6: Hydraulic control circuit for swivel angle control in closed loop

The actual power value is the product of the actual pressure value and the actual swivel angle value. The controller software ensures with the help of a minimum value generator that the active controller is assigned the relevant working point. In the steady-state condition, i.e. the swivel angle command value is equal to the actual swivel angle value, the power command value is equal to the actual power value or the pressure command value is equal to the actual pressure value, the control spool of the valve is in the central position. If higher-level controllers demand, for example, an increase in the swivel angle, which corresponds to an increase in flow, the valve spool must be shifted from the central position until the swivel angle has reached the required value. The sectional drawing in Figure 2.7 shows the A4VS variable displacement pump with HS4 control. Data sheet for the digital control electronic card for HS4 control is located in Appendix A. HS4 control consists of the following components [11]:

- Axial piston pump A4VS-HS4 with built-in proportional valve type 4WRE6-2X/822, including position transducer for sensing the swivel angle and valve travel
- Pressure transducer HM17 recommended for acquiring the system pressure
- Control electronics VT-VPCD for realizing all of the electrical functions required for the HS4 control

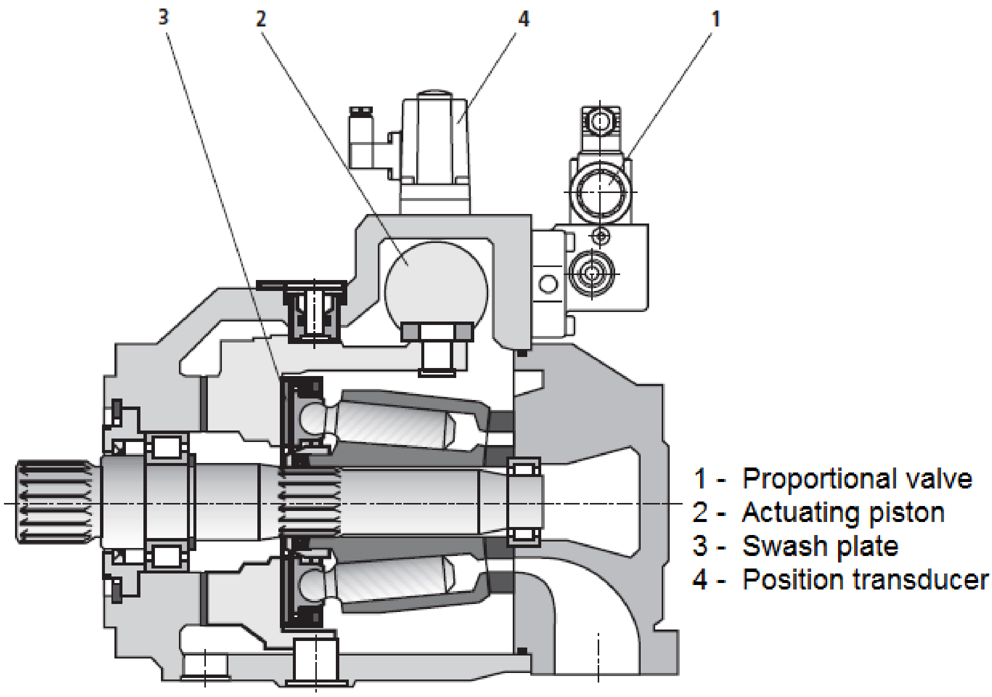


Figure 2.7: 2D visualization of variable displacement A4VS pump with HS4 control [11]

If the proportional valve is de-energized and the pump rotates clockwise and actuating pressure is present, the pump swivels to swivel angle $\alpha = 0$ for open loop variants (A4VGO) and $\alpha =$ -100 % for closed loop variants (A4VSG).

Electronic Controller Card
The VT-VPCD electronic pump cards are built up by a series of both analogue and digital in- and outputs, voltage- and current drivers, microcontrollers and more. Figure 2.8 shows the block schematics for VT-VPCD-1-1X/V0/1-0-1 configured for HS4 control with swivel angle transducer type AWX F004 D01.

The central unit is a microcontroller, which controls the entire sequence and realizes the functions of the controller. Configuration data, command values and parameters are saved in a non-volatile flash memory. Four binary-coded, digital inputs are used for calling up parameter sets from the memory, in which a maximum of 16 sets can be saved. A call-up activates a command value for the swivel angle, pressure and power limitation as well as ramp times for swivel angle and pressure build-up.

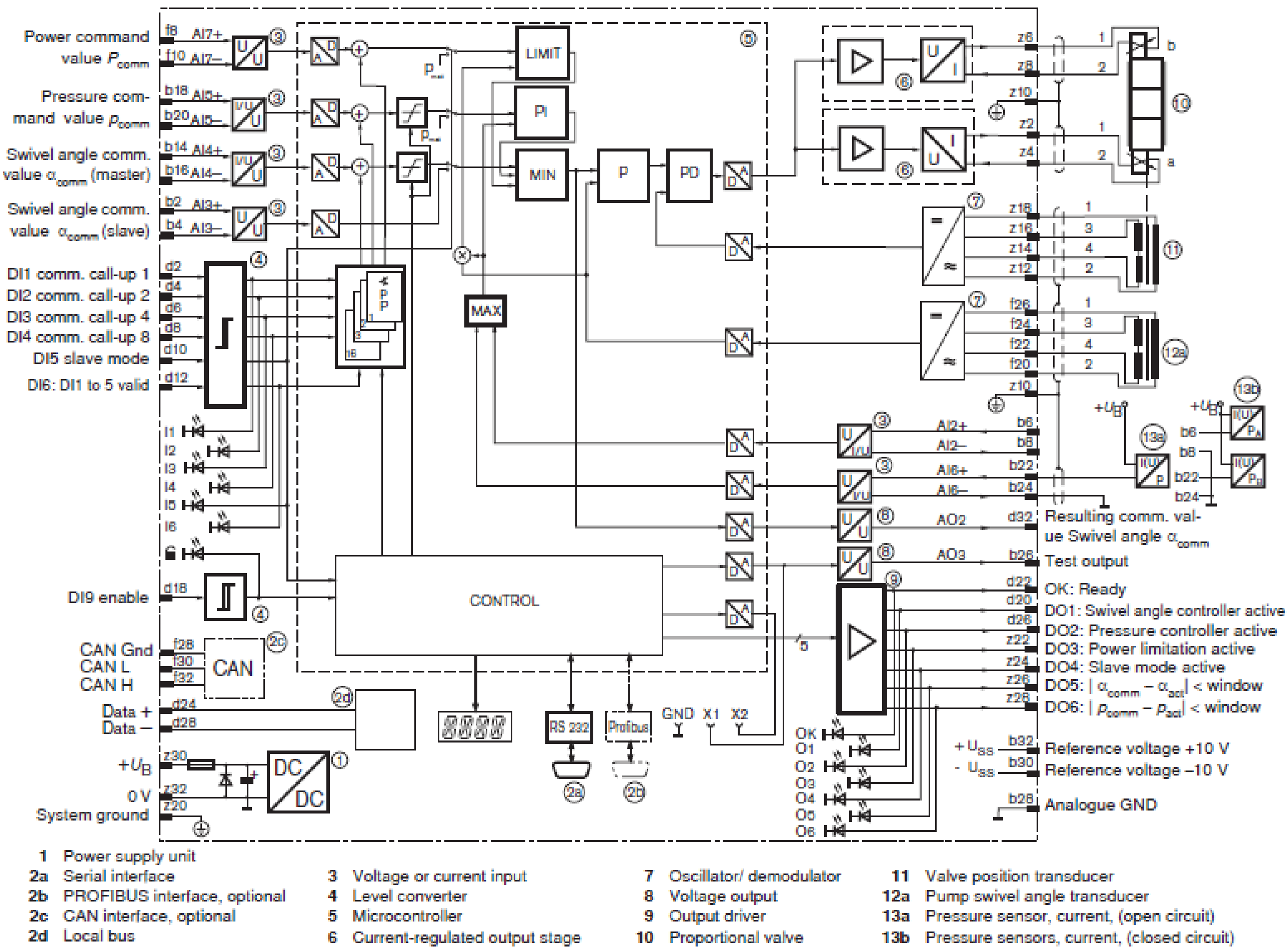


Figure 2.8: Block schematics of VT-VPCD pump controller card [11]

The analogue command values for swivel angle, pressure and maximum power are entered via the differential inputs AI7, AI5 and AI4. When a positive swivel angle command value is applied, the pump swivels in counter clockwise direction, hence direction of flow from P $\rightarrow$ B. The digital call-up command values are added to the analogue values and the sum of both commands is fed to the input of the controller through a ramp generator. The output signal from the controller depends on the difference between command values and actual values. Large deviation from command value yields for higher output signal.

In order to close the control loop the controller card has one actuator output and three or four sensor feedback signals for open or closed hydraulic circuit, respectively. The actuator is a proportional valve which is actuated by controlling the current to coil A and B. Sensor feedback from position transducers for the proportional valve and the axial piston cylinder required as input to the controller. The controller also requires pressure feedback. For open hydraulic circuits only one pressure sensor is needed, while closed hydraulic circuits yields for two sensors, one for each A- and B-side of the pump. In the closed circuit, both pressure sensors are evaluated. As soon as the control electronics is in the closed-loop pressure control mode, the higher of the two pressures determines the control characteristics [11]. The feedback sensors are described in section 2.5.

2.5 Position Transducers and Pressure Sensors

Sensors are an absolute requirement for the HS4 control system in order to determine the state of control elements and close the control loop. For the HS4 control function properly a feedback from swivel angle, valve position and working pressures is required. These feedback sensors are described below.

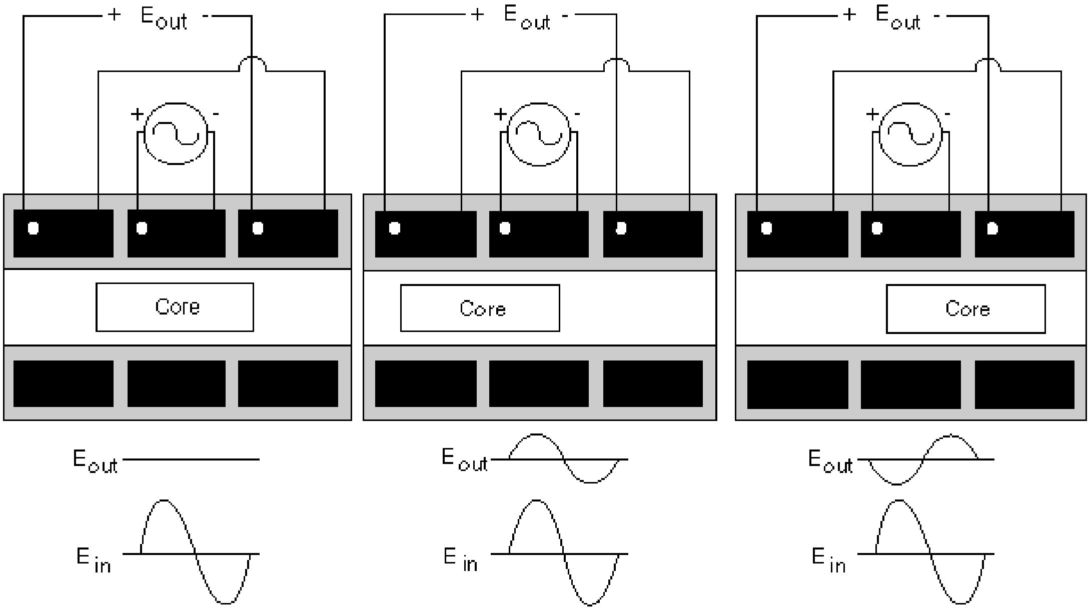
Position Transducers

In Bodac the VT-VPCD pump cards can be configured for different sensors for position feedback from the swivel angle transducer. The valve position sensor can not be configured. The valve position sensor is a linear variable differential transformer (LVDT) sensor to measure displacement for the spool position in the valve. The output signal from the LVDT sensor represents the distance an object has travelled from a reference point. The displacement measurement also indicates the direction of motion [12]. In figure 2.9 the working principle of a LVDT is shown. With no displacement, hence with the spool in center position, the induced output voltage is zero. For a positive defined displacement the amplitude of the induced voltage is linear proportional to displacement and has the same phase as the reference signal, see figure 2.9b. A negative defined displacement is 180° phase shifted from the reference signal as shown in figure 2.9c. A negative displacement has the same relative amplitude as a positive displacement. Hence, the 180° phase shift is the only difference. LVDT sensors have many advantages. Some are given below [13]:

Friction-free operation: One important feature of an LVDT is its friction-free operation. There is no mechanical contact between the LVDT's core and the coil assembly.

Infinite resolution: Since an LVDT operates on electromagnetic principles in a friction-free structure, it can measure infinitely small changes in core position.

Separate coil and core: The coil can be isolated from the core by inserting a non-magnetic tube between the core and the bore. Then, a pressurized fluid can be contained within the tube in which the core is free to move, while the coil assembly is unpressurised. This makes the LVDT ideal for spool position feedback in hydraulic proportional valves.



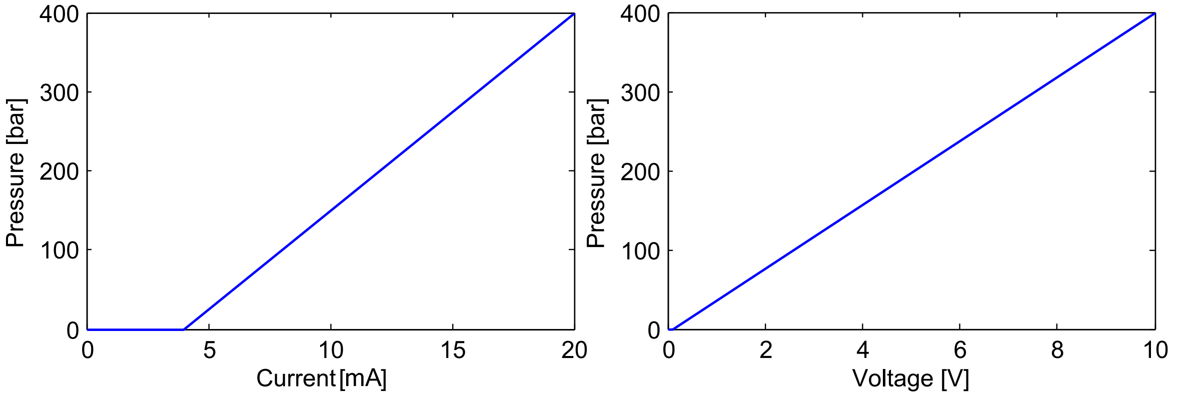
(a) No displacement (b) Positive displacement (c) Negative displacement

Figure 2.9: LVDT signals for no displacement (a), positive displacement (b) and negative displacemnt (c) [14]

Pressure Sensors

Both current and voltage sensors can be used for the pressure transmitters. If the pump card is configured for current sensors, it uses a signal in the range 4 to 20 mA. When configured for voltage sensors, the pump card uses a voltage signal in the range 0 to 10 V.

Bosch Rexroth recommends the use of pressure transducer HM17. HM17 is a pressure transducer with integrated electronics. It comes in eight pressure measuring ranges, from 50 to 600 bar. It is available with both current and voltage output. Figure 2.10a shows the linear relation between pressure and current, while Figure 2.10b shows the linear relation between pressure and voltage. Both with a pressure measuring range from 0 to 400 bar.



(a) Current sensor (b) Voltage sensor

Figure 2.10: Relation between pressure/current (a) and pressure/voltage (b) for pressure sensor at range 0 to 400 bar

2.6 Modelling Hydraulic Systems

A hydraulic simulation model has to be implemented in the HIL simulation loop. Real-time simulations yields for dynamic hydraulic simulations which allows for a much more detailed investigation of the performance of the hydraulic system compared to steady-state analysis. Steady-state analysis corresponds to solving a set of algebraic equations whereas dynamic simulation corresponds to solving a mixed set of differentials and algebraic equations [15]. Dynamic simulations have initial value problems, meaning the pressures in pressure nodes, volumes and the position and velocity of all mechanical components has to be known to perform a dynamic simulation. This introduces the differential equation for the pressure gradient, shown in Equation 2.1.

$$\dot{p} = \frac{\beta \cdot (Q - \dot{V})}{V} \tag{2.1}$$

Where;

$\dot{p}$	=	Pressure gradient	[Pa/s]
β	=	Oil stiffness	[Pa]
Q	=	Net-flow into volume	[m^3/s]
$\dot{V}$	=	Volume gradient	[m^3/s]
V	=	Volume	[m^3]

In the equation, if the pressure gradient is positive the pressure is increasing and visa versa. The pressure gradient is determined by the net-flow into the volume and/or change in volume. An other important feature when considering dynamic simulation is the oil compressibility and stiffness. The oil stiffness, β, depends on temperature, dissolved air and compressibility. An overview of the oil stiffness subjected to different pressure and temperatures is shown in Figure 2.11.

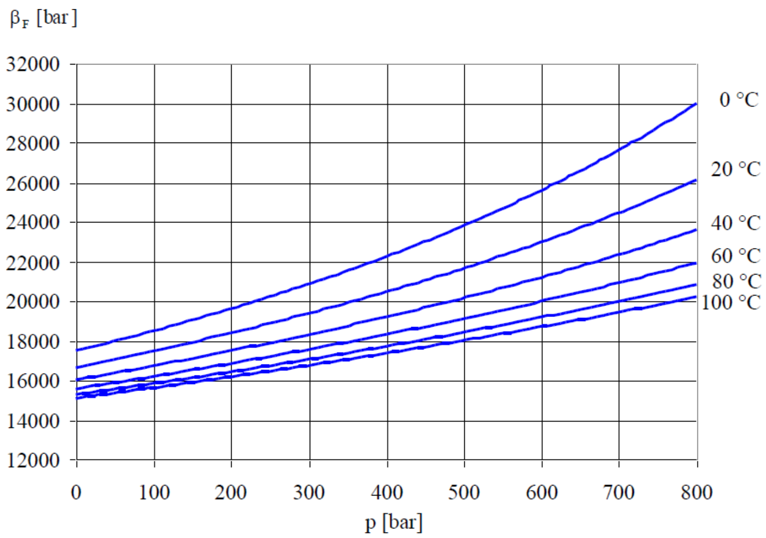


Figure 2.11: Oil stiffness subjected to different pressure and temperatures [15]

A typical temperature chosen when running a dynamic simulation is 40°C. A good estimation of the oil stiffness can be determined by adding an equation with a linear relation between zero pressure value and maximum working pressure, as shown in Equation 2.2.

$$\beta = \beta_{0\,bar} + \frac{p_{actual}}{p_{max}}(\beta_{p_{max}} - \beta_{0\,bar}) \tag{2.2}$$

Where;

$\beta_{0\,bar}$	= Oil stiffness at 0 bar	[Pa]
$\beta_{p_{max}}$	= Oil stiffness at maximum working pressure	[Pa]
p_{actual}	= Actual pressure	[Pa]
p_{max}	= Maximum working pressure	[Pa]

Setting Up a Dynamic Simulation

Setting up a dynamic simulation of a hydraulic circuit is done by following these steps [15]:

1. Identify all volumes in the system and set up the pressure build up equation for each volume, identify all accumulators and set up the volume expansion equations.
2. Identify all orifices and their dependencies on the motion of mechanical parts in valves.
3. Determine initial pressure for each volume and initial fluid volume for each accumulator.
4. Calculate pressure gradients for each volume and volume gradient for each accumulator.
5. Calculate acceleration of all movable mechanical parts in valves.
6. Update pressure in each volume and volume of each accumulator.
7. Update position and velocity of all movable mechanical parts in valves.
8. If the analysis is not yet concluded then go back to 4.

By following these steps a dynamic simulation can be set up by examining a hydraulic circuit and knowing its initial conditions. When the hydraulic system is modelled using a commercial simulation tool, like SimulationX, the governing equations are set up and can be exported as a shared library and executed on a real-time target.

SimulationX is a high-end modelling platform for system analysis. It contains libraries for 1D mechanics, 3D multibody systems, power transmission, hydraulics, pneumatics, thermodynamics, electrics, electrical drives, magnetics and controls. This makes the program applicable for modelling of hydraulic and mechanic systems for execution as part of a HIL system. This yields for scalable simulation models, only limited by the real-time targets system performance.

Chapter 3

Solution

This chapter presents the work carried out in order to build the test rig and implement a HIL simulation. Based on set requirements various systems are benchmarked against each other to find the best solution for the HIL system. The process of configuring both LabVIEW and Bodac for HIL simulation is described, as well as validation and verification of the results.

3.1 Requirements

Requirements is set in order to determine the specifications for the HIL system. Some requirements are set by the system, i.e number of inputs and outputs needed in order to obtain a functional HIL system. Requirements set by NOV forms the final set of requirements to finalize the specifications for the HIL system.

3.1.1 Technical Requirements

Technical requirements are determined by the required inputs and outputs to and from the pump card. The pump card must function under similar conditions as in NOVs hydraulic applications, hence the same electrical interface has to be set up. The required analogue inputs and outputs listed below are needed.

Analogue inputs:

- Actual valve (LVDT model), ±10V
- Actual swivel angle, 0-10V
- Actual pressure A, 0-10V
- Actual pressure B, 0-10V
- Swivel angle command, ±10V
- Pressure command, 0-10V

Analogue outputs:

- Resulting valve command, ±10V
- LVDT reference signal, 5 kHz, ±10V

The LVDT reference signal introduces the need for high-speed waveform generation and acquisition for the feedback signal. The 5 kHz waveform reference signal sets the requirement for a sample rate based on how many samples are needed to form one period of the waveform. Typical sample rates are 100, 500 and 1000 kS/s. Equation 3.1 computes number of samples for one period of a waveform.

$$\text{Samples} = \frac{K_{sample}}{\lambda} \tag{3.1}$$

Where;
K_{sample} = number of samples per second [S/s]
λ = waveform frequency [1/s]

The pump card from Bosch Rexroth need 24 VDC supply voltage. 230 VAC/24 VDC power supply is therefore required for the HIL system to operate. From the technical requirements directly related to the pump card, the real-time controller must have at least six analogue inputs, two analogue outputs and a 24 VDC power supply.

3.1.2 Customer Requirements

NOV wants to use the HIL system for pre-tuning, experiments and for training purposes. Hence, different pump models and different hydraulic circuits must be modelled and simulated in order to create various scenarios for the pump controller card. In order to execute the simulation models in a deterministic manner, all the simulation models have to be executed on the real-time controller. Hence, the real-time target must have sufficient performance to execute these simulation models.

From equation 3.1 the number of samples for one period of a waveform could be determined. A 5 kHz frequency yields for 20, 100 and 200 samples for one period for 100, 500 and 1000 kS/s, respectively. 20 samples per period was decided to be sufficient resolution for the system, hence the analogue input and output modules should have a sample rate of at least 100 kS/s.

Since the system has to be applicable for training purposes, NOV wants a GUI with an instructor panel, where information and controls for the simulation models, pump controller status and network settings are configured. The pump card has digital outputs for various status states and digital input to set various commands, including a digital input to activate the pump card. These digital inputs and outputs determines a new requirement, a digital in- and output (DIO) module.

In order to preform a complete pump tuning sequence, NOV requires the possibility to add imperfections to the system, such that all tuning features in Bodac can be used in order to simulate a physical system.

Final Requirements are set based on the requirements set in section 3.1.1 and 3.1.2. These are:

- Minimum six analogue inputs
- Minimum two analogue outputs
- Minimum eight DIO
- 230 VAC/24 VDC power supply
- Real-time execution of pump- and hydraulic simulation model

3.2 Design Specification

This section describes the process of selecting suitable components for the HIL system based on the requirements set in section 3.1. In the stage of selecting a real-time controller for the system, factors like cost, hardware performances and possibilities for the future expansion were considered.

While considering various solutions for a real-time controller, controllers from National Instruments where considered because of easy implementation with LabVIEW and LabVIEW Real-Time module. Three suggestions, a cRIO system, a PXI system and a compactDAQ (cDAQ) with a real-time target PC was set up. The different system configurations are presented below.

CompactRIO
CompactRIO is a commonly used controller for real-time simulations, like HIL testing. General theory related to the cRIOs hardware can be read in section 2.2.

In order to meet the requirements set in section 3.1 the system in Table 3.1 is proposed. The cRIO-9075 uses the VxWorks RTOS which is incompatible with dynamic link libraries (DLLs). DLLs provide a way for programs to access external code, such as code exported from SimulationX or Simulink™.

The DLL files can be compiled into another file type (.out file) in order to execute on a RT target running VxWorks. This process is described in section 3.3.7.

Table 3.1: CompactRIO system with I/O modules

Type	**Qty**	**Specifications**	**Price Ex-VAT [NOK]**[1)]
NI cRIO-9075	1	400 MHz Real-Time controller, LX25 FPGA, 128 MB RAM, Ethernet, RS232, 4 slot chassis	10 300,-
NI 9263	2	Analogue output ±10 V, 16 bit resolution, 100 kS/s, 4ch module	6 920,-
NI 9215	1	Analogue input ±10 V, 16 bit resolution, 100 kS/s, 4ch module	4 585,-
NI 9401	1	Bidirectional digital I/O, 8ch module	2 470,-
NI 9924	1	DIN-rail screw terminal to 25 pin D-SUB module	511,-
NI SH25F-25M	1	25 pin D-SUB male to female, 1 m	251,-
Sum			25 037,-

[1)] Prices are gathered from ni.com at 05/03-2015

This cRIO system is suitable for HIL applications for simple real-time simulation models. More complex simulation models might exceed the CPUs load capacity. To increase the computational power in the HIL system an external computer running as a LabVIEW RT target can be implemented. Table 3.2 shows pros and cons for using a cRIO in the HIL system.

Table 3.2: Pros and cons for compactRIO as the real-time controller in the HIL system

Pros	**Cons**
Modular I/O for easy expansion	cRIOs with VxWorks OS are not compatible with DLL files. Has to be converted to .out files to deploy on the cRIO
FPGA target	DLL compatible performance cRIOs are expensive (50 kNOK +)
Small footprint and rugged design	Limited CPU capacity

PXI system

PXI systems are PC-based platforms for measurement and automation systems. The PXI system comes with two different chassis', a standard PXI chassis and a PXI Express chassis. The standard PXI chassis has a dedicated 10 MHz system clock while the PXI Express chassis has a 100 MHz differential system clock. Hence, the I/O update rate would not be limited by the PXI system's clock rate. In Table 3.3 a PXI Express system with a RT controller is considered to be used in the HIL application.

Table 3.3: PXI system with I/O modules

Type	Qty	Specifications	Price Ex-VAT [NOK][1)]
NI PXIe-8100 RT	1	1.66 GHz controller, 512 MB RAM, USB, Ethernet, RS232	8 480,-
NI PXIe-1071	1	4 slot PXI Express chassis	7 700,-
NI PXI-6722	1	Analogue output ±10 V, 13 bit resolution, 182 kS/s per channel, 8ch module	9 865,-
NI PXI-6220	1	16ch analogue input ±10 V, 16 bit resolution, 250 kS/s, 24ch digital I/O	5 930,-
NI SCB-68A	2	68 pin screw terminal connector	5 800,-
NI SHC68-68-EPM	2	68 pin X-series cable, 2 m	2 510,-
Sum			34 355,-

[1)] Prices are gathered from ni.com at 05/03-2015

The PXI system in Table 3.3 would be a suitable solution for this HIL application. When compared to a cRIO-9075 the PXIe-8100 controller is DLL compatible which yields for easier implementation for external code. The system has several available I/O connections for future expansion, as well as two free slots in the PXI chassis. This comes with a cost, hence, the PXI system is over 37% more expensive than a cRIO solution. The pros and cons for this PXI system is listed in Table 3.4.

Table 3.4: Pros and cons for a PXI system as the real-time controller in the HIL system

Pros	Cons
Compatible with DLL files	Expensive solution
Available I/O connections for future expansion	Large footprint because of chassis size
Two free slots for additional I/O modules	

CompactDAQ

A compactDAQ is a data acquisition platform that integrates connectivity and signal conditioning into modular I/O. In order to use a cDAQ for HIL simulation an external real-time target PC or one of the performance versions of the cDAQ with integrated controller is needed.

By comparing Table 3.1 and 3.5 it comes clear that many of the same modules and components are used for the comapctRIO and compactDAQ system. The only thing varying is the chassis module.

Table 3.5: CompactDAQ system with I/O modules

Type	Qty	Specifications	Price Ex-VAT [NOK][1)]
NI cDAQ-9184	1	4 slot Ethernet chassis Ethernet	9 345,-
NI 9263	2	Analogue output ±10 V, 16 bit resolution, 100 kS/s, 4ch module	6 920,-
NI 9215	1	Analogue input ±10 V, 16 bit resolution, 100 kS/s, 4ch module	4 585,-
NI 9401	1	Bidirectional digital I/O, 8ch module	2 470,-
NI 9924	1	DIN-rail screw terminal to 25 pin D-SUB module	511,-
NI SH25F-25M	1	25 pin D-SUB male to female, 1 m	251,-
Sum			24 082,-

[1)] Prices are gathered from ni.com at 05/03-2015

When considering the compactDAQ system in Table 3.5 it is clear that it would not be applicable for the HIL application because of the required external RT target. All three systems are compared against each other in Table 3.6.

Table 3.6: Comparing the considered real-time systems

Feature	PXI system	CompactRIO	CompactDAQ
Expandable I/O modules	✓		
DLL compatible	✓		
Reconfigurable FPGA		✓	
Small footprint		✓	✓
External RT target expansion	✓	✓	✓
Price	34 355,-	25 037,-	24 082,-

Based on the comparison done in Table 3.6 a PXI system will be the most suitable system for the HIL application when performance and implementation is considered. Although the PXI system is considered as the best solution for the HIL application the cost was considered to high in comparison to the compactRIO solution where the compactRIO module was made available throughout the project period from University of Agder. Choosing the compactRIO solution resulted in an immediate start for developing the real-time simulation interface.

3.3 Implementation

This section presents the background for what makes the complete HIL system. The process from building the HIL rig to implementing a dynamic simulation model and the solutions behind is described in the following sections.

3.3.1 Building the HIL System

The components which make up the HIL system should be mounted on a plate. It should be a portable solution, e.g. not heavy or large in terms of dimension. Three main components the HIL system consists of are given below;

- 230VAC/24VDC power supply from Mean Well.
- CompactRIO Real-Time controller with four I/O modules from National Instruments.
- Pump controller card from Bosch Rexroth.

Figure 3.1 shows the planned layout for the test rig. The components are mounted on a DIN rail and is surrounded by cable canals to obtain a proper cable structure. The dimension of the mounting plate is 550 mm×250 mm.

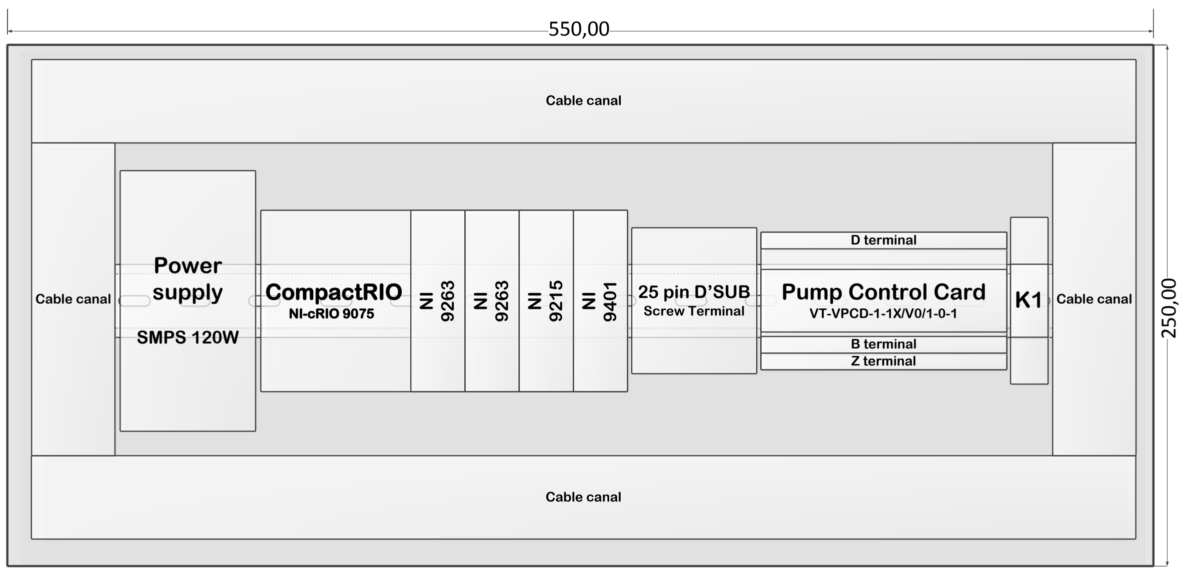


Figure 3.1: Layout for the test rig

The mounting plate was modelled in SolidWorks, exported as a computer aided manufacturing (CAM) file and cut to the right dimensions on a plasma cutter at the University of Agder. The cRIO-9075, VT-VPCD pump card and euro card holder was available from project start, but all other components and equipment had to be ordered. The power supply was ordered from Elfa Distrelec, the I/O modules and D-sub screw terminal was ordered from National Instruments. Cables, wiring, cable canal, DIN rail and cable crimps was provided by NOV.

Figure 3.2 shows the completed test rig with all components mounted and with wired interface between the pump card and the cRIO. Communication from the HIL system to a host PC is done by using a crossed network for TCP/IP communication. A RS232 serial interface is mounted onto the pump cards front panel for communication with the computer used to configure the pump controller card.

Figure 3.2: Test rig with all components and completed wiring

The HIL systems 24VDC electrical interface is shown in Figure 3.3. Three components are connected to the power supply. These are the cRIO, pump card and a digital I/O converter. A power switch is mounted in front of the pump card to reset the pump card if faults or similar occur.

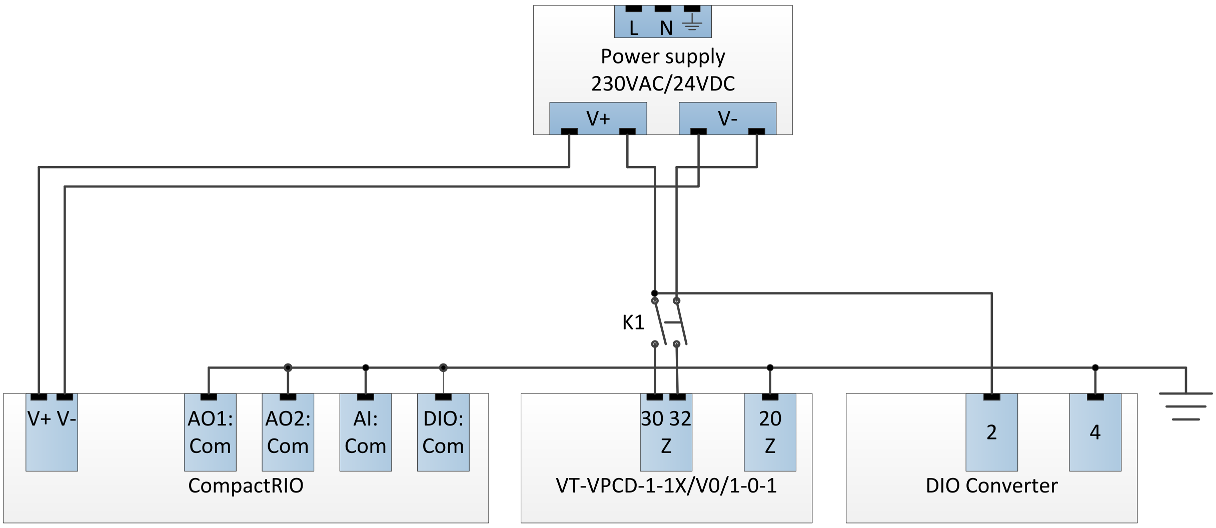


Figure 3.3: Illustration of the 24VDC power supply circuit

The digital I/O converter is a circuit board with a voltage step-up converter and two voltage dividers. The digital I/Os to and from the pump card is operating at 24V, while the inputs to the DIO card on the cRIO is operating at 5V. A data sheet showing the technical data for the pump card is shown in

Appendix A. The circuit board is also used as a system ground reference point. In electrical systems with digital and analogue signals it is important to use a star-coupled ground system. Meaning that the analogue and digital circuits are separated from each other and connected together through a common ground potential. The ground system shown in Figure 3.4 is simplified in terms of the drawn wiring diagram. Figure 3.4 shows the circuit board diagram for the digital I/O converter. A terminal block is soldered onto a circuit board for easily interfacing with the rest of the electrical system on the test rig.

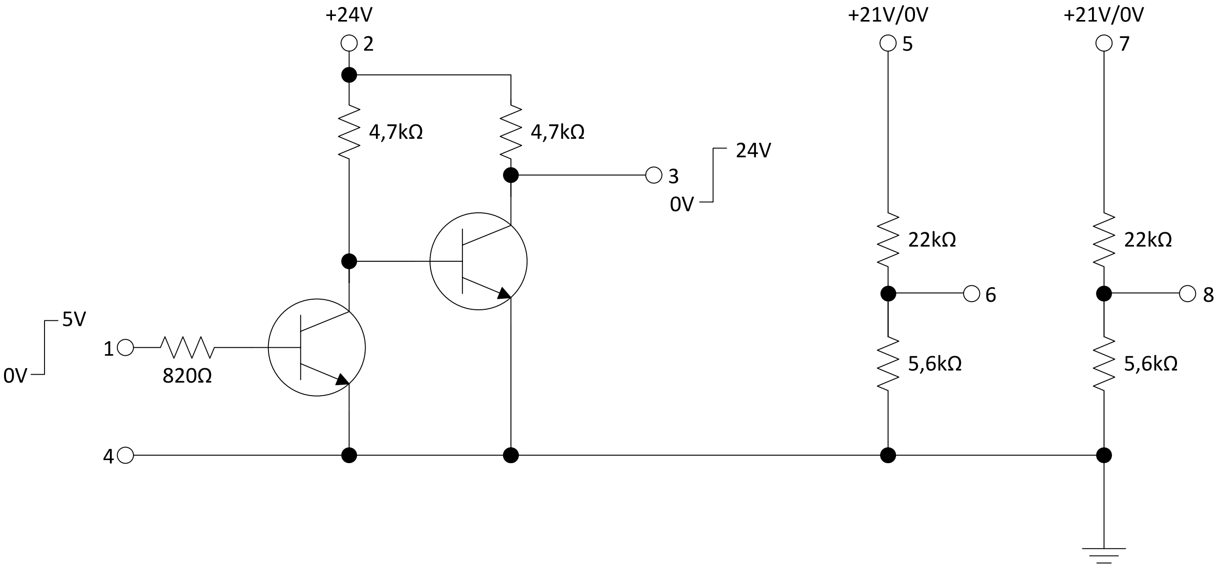


Figure 3.4: Digital I/O converter with step-up converter and voltage divider

Analogue and digital input and outputs are wired from the terminals on the card holder to the cRIO. An illustration of the wired interface between the cRIO and pump card is shown in figure 3.5. The analogue inputs and outputs are configured in LabVIEW and given appropriate names related to the the inputs and outputs on the pump card. Configuring of the I/O modules in LabVIEW is described in section 3.3.3.

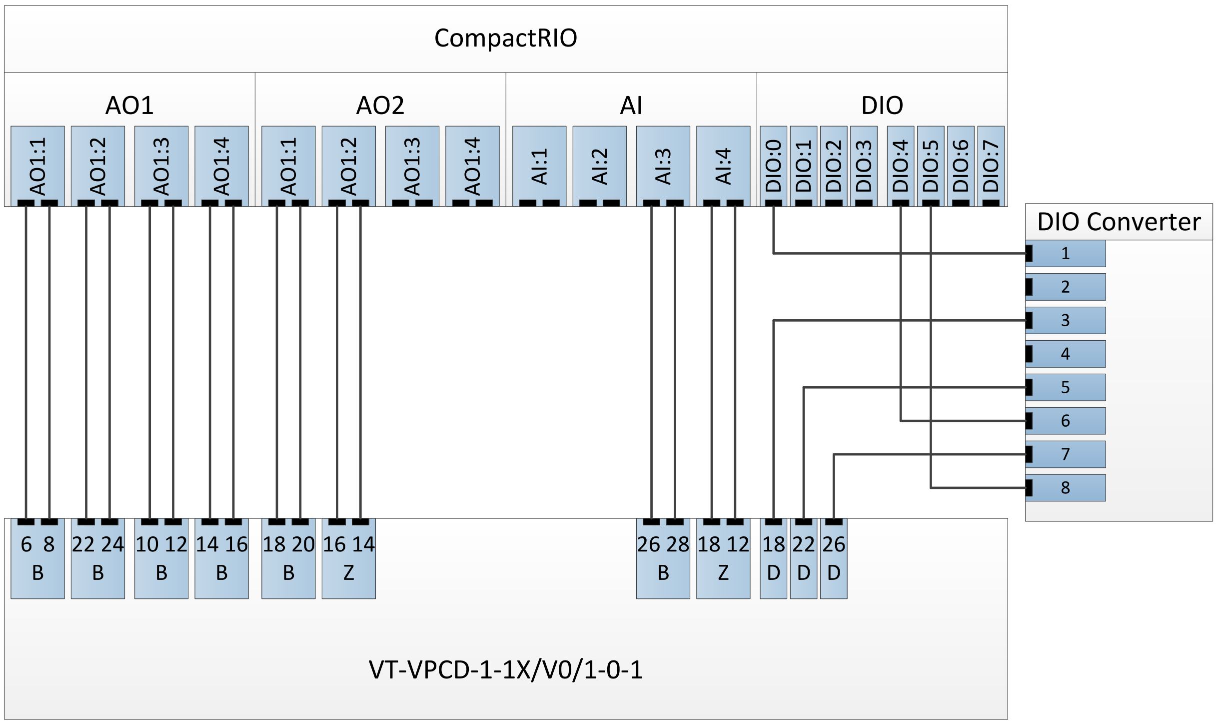


Figure 3.5: Illustration of the electrical interface between cRIO and VT-VPCD-1-1X/V0/1-0-1

Table 3.7 - 3.9 show a description for each terminal connection for terminal D, B and Z.

Table 3.7: Connector configuration for terminal row D

Pin	Description	VT-VPCD-1-1X
2	DI1	Command-Setpoint 1
4	DI2	Command-Setpoint 2
6	DI3	Command-Setpoint 4
8	DI4	Command-Setpoint 8
10	DI5	Slave mode
12	DI6	DI 1-5 valid
14	DI7	n.c
16	DI8	n.c
18	DI9	Enable
20	DO 1	Swivel angle controller active
22	OK	Ready
24	Data +	Local CAN Bus Input/Output
26	DO 2	Pressure controller active
28	Data -	Local CAN Bus Input/Output
30	AO1	n.c
32	AO2	Resulting swivel angle command value

Table 3.8: Connector configuration for terminal row B

Pin	Description	VT-VPCD-1-1X
2	AI3+	Swivel angle command value (Slave) + (U)
4	AI3-	Swivel angle command value (Slave) - (U) reference
6	AI2+	Actual pressure value (A) + (U/I) or (U)
8	AI2-	Actual pressure value (A) - (U/I) or (U) reference
10	AI1+	Actual pressure value MCP-40/4742 + (U)
12	AI1-	Actual pressure value MCP-40/4742 - (U)
14	AI4+	Swivel angle command value + (U/I) or (U)
16	AI4-	Swivel angle command value - (U/I) or (U) reference
18	AI5+	Command pressure value + (U/I) or (U)
20	AI5-	Command pressure value - (U/I) or (U) reference
22	AI6+	Actual pressure value (B) + (U/I) or (U)
24	AI6-	Actual pressure value (B) - (U/I) or (U) reference
26	AO3	Test output (X1)
28	AGND	Analog GND
30	REF-	Reference voltage -10V
32	REF+	Reference voltage +10V

Table 3.9: Connector configuration for terminal row Z

Pin	Description	VT-VPCD-1-1X
2	MA+	Solenoid A+
4	MA-	Solenoid A-
6	MB+	Solenoid B+
8	MB-	Solenoid B-
10	Shield	Shield
12	L1O-	LVDT Valve Power -
14	L1I-	LVDT Valve actual value -
16	L1I+	LVDT Valve actual value +
18	L1O+	LVDT Valve Power +
20	System ground	System ground
22	DO3	Power limitation active
24	DO4	Slavemode active
26	DO5	$\mid$ Command angle - actual angle $\mid <$ window
28	DO6	$\mid$ p command - p actual $\mid <$ window
30	UB	Supply voltage
32	LO	Common

3.3.2 Setting Up the Pump Card for HIL Application

Because all the analogue input and output modules on the cRIO requires voltage signals the pump cards inputs and outputs has to be configured. In Bodac a variety of choices can be made for the terminal connections listed in Table 3.7 - 3.9.

In the "Analog I/O" window all the analogue inputs to the pump card can be configured (see Figure 3.6). The inputs can be configured for both voltage and current input in the ranges ±10 V, 0-10 V and 4-20 mA. The analogue output modules on the cRIO are analogue voltage modules. Hence, all inputs to the pump card has to be configured for voltage inputs. The angle command is a ±10 V signal because the pump swivels to both directions. A negative voltage input yields for a negative swivel angle and visa versa. The pressure command, power command and actual pressures have no negative value, hence a voltage signal is set in the range of 0-10 V is used. The pressure range has an upper limit set to 400 bar for both sensor inputs. This setting applies to the configuration in Figure 2.10b.

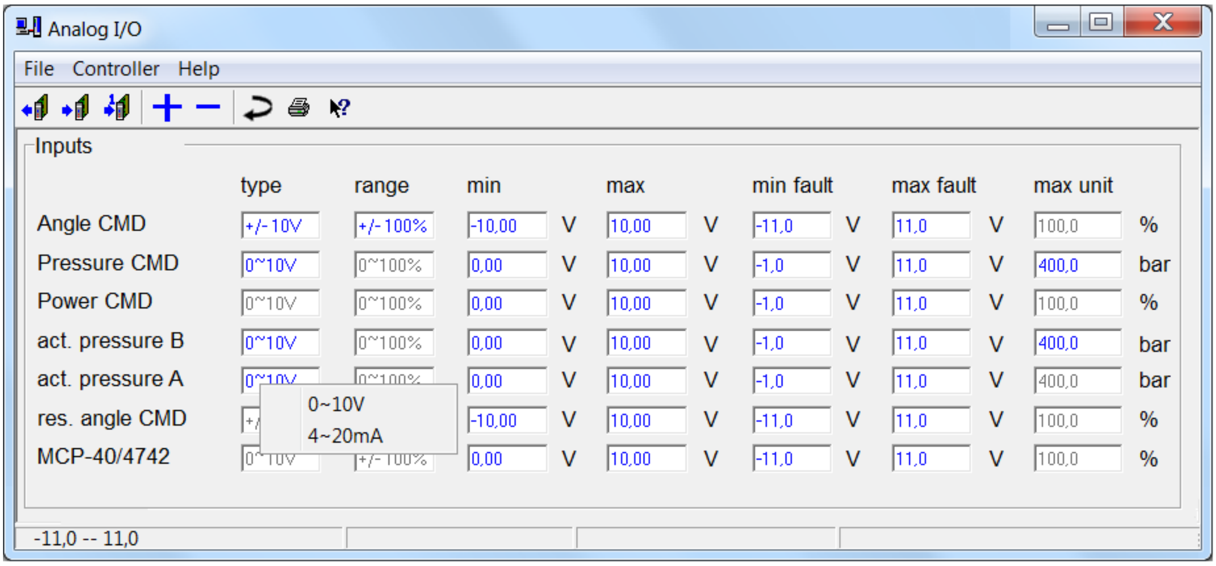


Figure 3.6: Analogue I/O configuration

The valve and angle sensor are configured from the "Configure 1" window, shown in Figure 3.7. Again, the inputs and outputs have to be configured according to voltage signals. The valve output is chosen to a ±10 V output, while the angle sensor is set to a 0-10 V signal as seen from the figure below. The angle sensor measures a relative angle between -100% and 100%. Hence, the angle sensor must operate according to Table 3.10.

Table 3.10: Voltage signals with respect to relative swivel angle

Relative swivel angle	**Voltage signal**
-100%	0 V
0%	5 V
100%	10 V

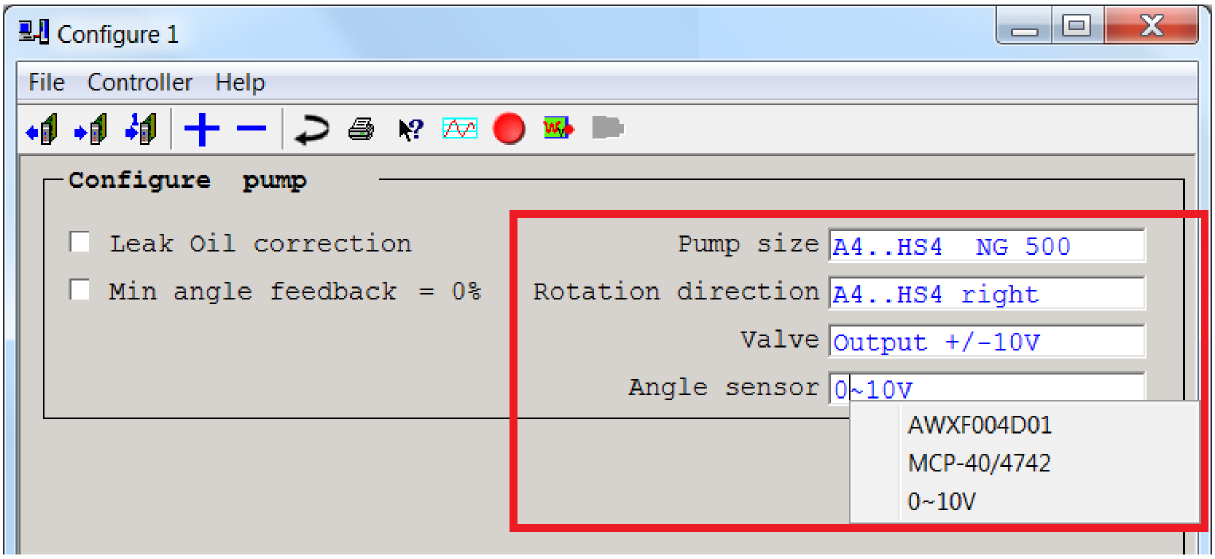


Figure 3.7: Valves and sensors

When the valve output is configured as a ±10 V signal the terminals used when connected to a 4WRE6V-X/822 or 4WRE10V-2X proportional valve are deactivated. From the "Test Jack" window in Bodac the terminals b26 and b28 can be configured for a set of outputs as shown in Figure 3.8. Because the regular valve output is deactivated this output has to be configured for "Valve CMD". The pump size with related control device is chosen under the "Pump size" option. The HIL application is configured for HS4 control for pump sizes 250-, 355- and 500 cm^3.

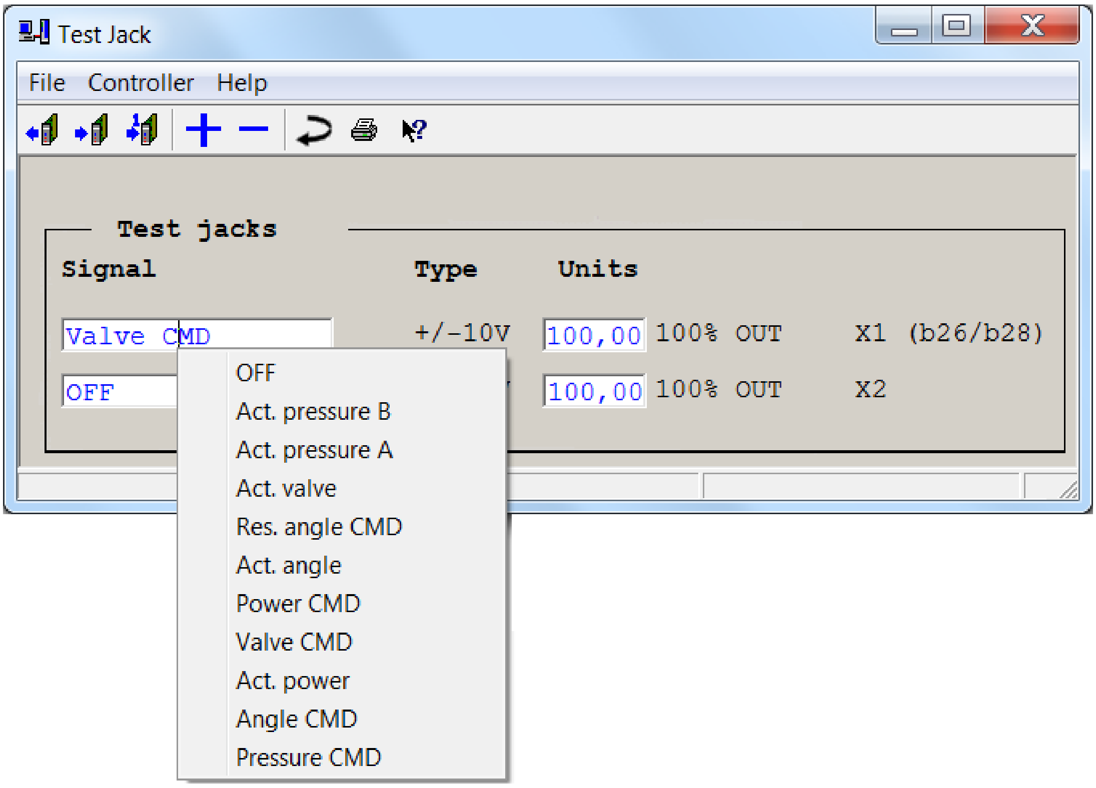


Figure 3.8: Testjack

3.3.3 Setting Up the Real-Time Application

Setting up the compactRIO for the real-time application is done by creating a LabVIEW project on a host PC. LabVIEW programming is done on a host PC and later deployed on its respective location on the compactRIO or FPGA target.

LabVIEW detects the compactRIO and its I/O modules when connected to the host PC via TCP/IP. When configuring the compactRIO one has to chose between two programming modes: Scan interface or FPGA interface. In scan mode one can program the real-time controller, but not the FPGA. In scan mode the FPGA periodically scans the I/O and places it in a memory map, making it available to LabVIEW Real-Time. The periodically scanning makes the programming mode sufficient for applications that require single-point access to I/O at rates of a few hundred hertz [6].

The FPGA interface mode should be used for high performance control, custom triggering, high-speed waveform acquisition and generation [7]. One of the requirements, stated in section 3.1, is a 5 kHz reference signal for the LVDT. To meet this requirement, the compactRIO has to be configured in FPGA interface mode.

When creating a real-time application for compactRIO a flowchart, or similar, describing all software components and communication paths has to be made. Figure 3.9 shows the architecture for the HIL application.

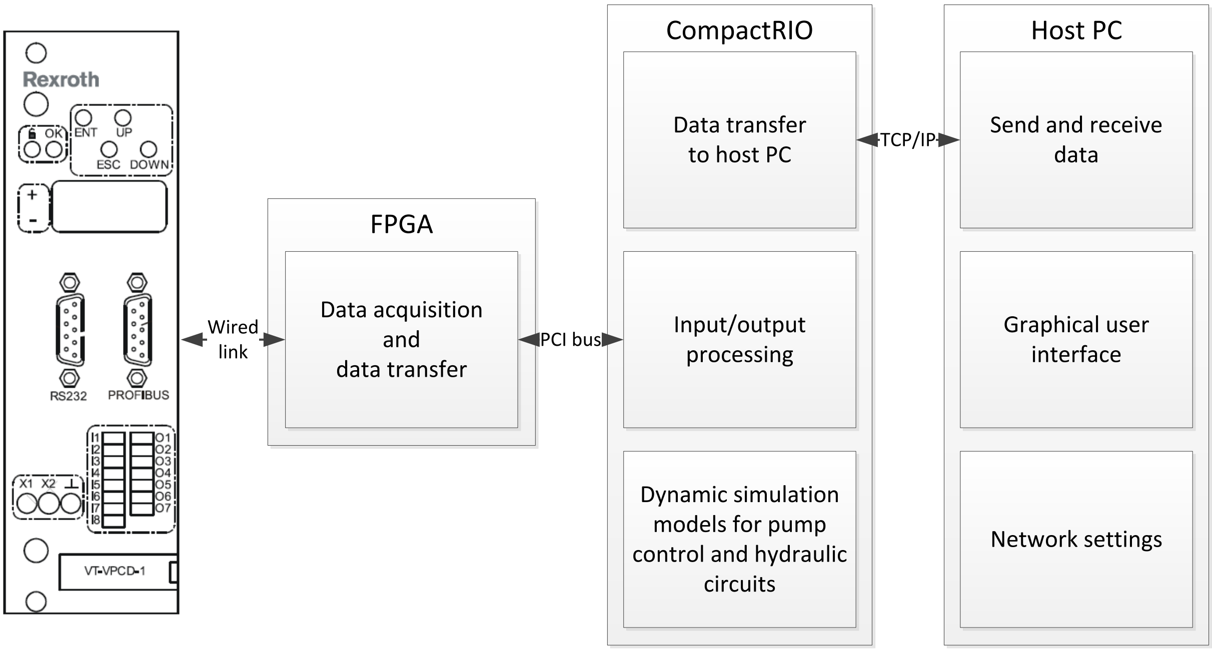


Figure 3.9: HIL system overview

Configuring FPGA Target for Data Acquisition
Configuration of inputs and outputs for the pump card is shown in section 3.3.2. The wired interface between the pump card and the cRIO (see Figure 3.5) is voltage signals in the range ±10 V, for the analogue inputs and outputs. The data acquisition to and from I/O modules on the cRIO are executed on the FPGA target. In LabVIEW project explorer all the inputs and outputs are renamed relative to its wired connection on the pump card.

The analogue input and output modules on the cRIO can be configured for two types of calibration modes: Raw and calibrated. In calibrated mode the analogue output module requires a value in the range of ±10 V. The analogue input module gives a value in the range ±10 V.

When calibrated in raw mode the signals depends upon resolution on the I/O module. NI9263 and NI9215 both has a 16 bit resolution which yields for a binary range with 2^{16} values. The ±10 V range is then converted to a range from -32768 to 32767. Raw calibration mode is used in the HIL application for better freedom of operation in terms of signal processing. The full voltage range in raw calibration mode typically exceeds ±10 V. Hence, all I/O signals has to be calibrated and scaled to the correct range. The I/O signals are calibrated and scaled by measuring the voltage across the terminals using a high resolution multimeter.

Figure 3.10 shows the block diagram for the I/O processing on the FPGA target. FPGA I/O nodes is used for reading and writing to and from the I/O modules. The acquired data is gathered in clusters and sent to and from the cRIO controller via the intern PCI bus, as illustrated in Figure 3.9.

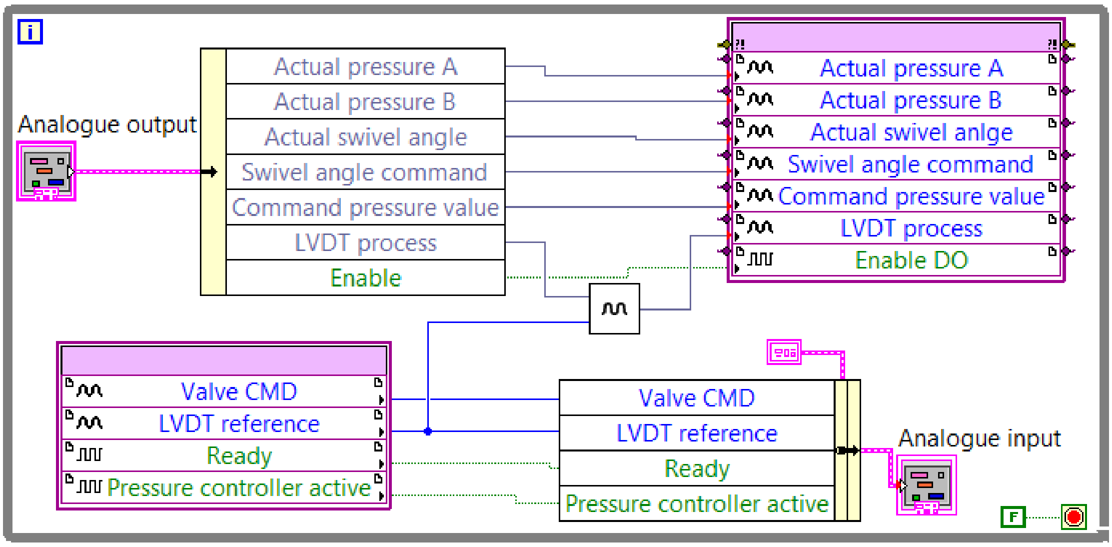


Figure 3.10: FPGA VI for in- and output processing

Because the position feedback sensor for the valve can not be changed in Bodac, the LVDT feedback signal has to be created in order to get a position feedback for the valve position. Figure 3.11 shows the interface between the pump card and the LVDT sensor.

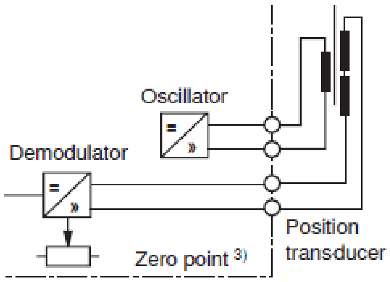
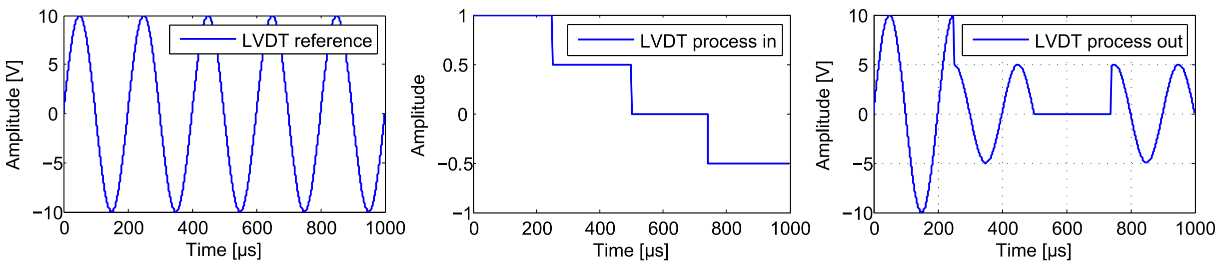


Figure 3.11: Clippings from data sheet showing oscillator and demodulator for the LVDT [16]

The LVDT reference signal is processed directly on the FPGA target because this signal requires high-speed waveform acquisition. The subVI in Figure 3.10 is used to create the LVDT process analogue output signal. The subVI contain logic functions and scaling functions. The principle for how the LVDT process signal out of the subVI is created is shown in Figure 3.12. Block diagram for the subVI can be viewed in Appendix C.



(a) LVDT reference (b) LVDT process in (c) LVDT process out

Figure 3.12: Principal functional description of subVI for LVDT signal processing

The LVDT reference signal has a frequency of 5 kHz, which means that one waveform period is 200 µs. The reference signal is multiplied with the process input to the subVI. The output from the subVI is created by multiplying the two input signals. A logic function checking whether the process signal in is positive or negative. If negative, the out signal is phase shifted 180°.

Configuring cRIO Controller
The main VI for the cRIO consists of three loops:

- Input and output processing
- Control and simulation loop
- Communication to and from host PC

The loops are created based on both deterministic and non-deterministic tasks. The input and output processing loop receives and sends data to the FPGA. Data received from the FPGA is accessed by opening a FPGA VI reference and accessing the clusters by using a read control. Data sent to the FPGA is accessed the same way, using a write control instead of a read control. Figure 3.13 shows the input/output loop for sending and receiving data from the FPGA target.

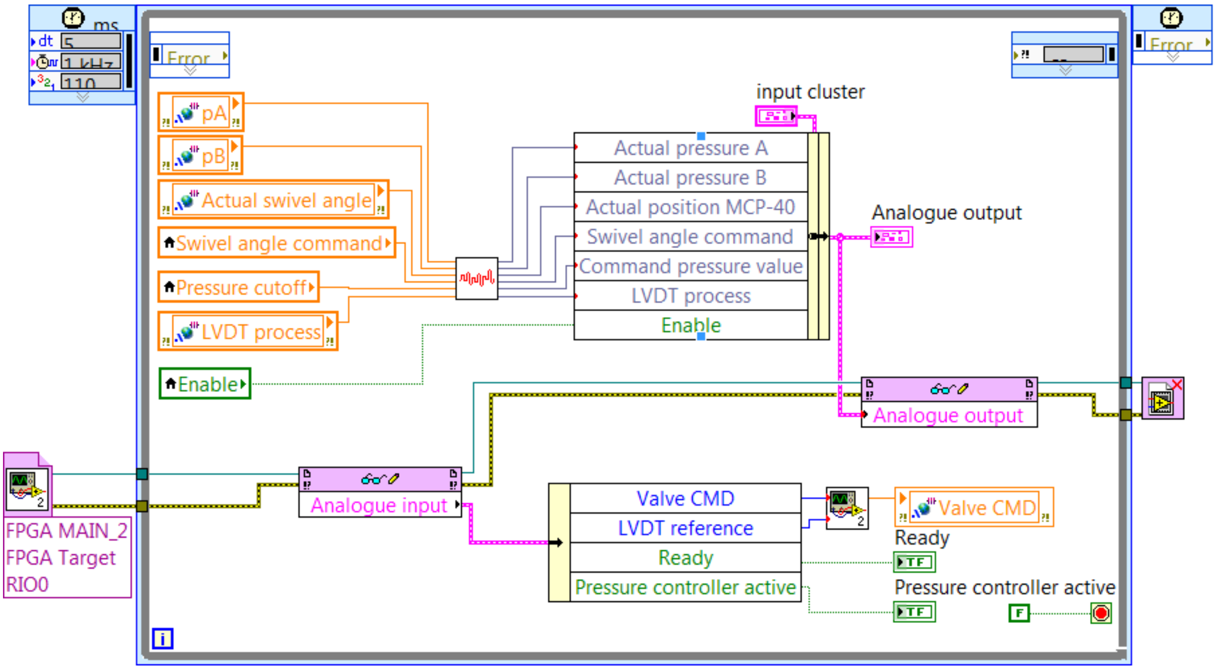


Figure 3.13: Input/output processing to and from FPGA target

A timed loop is used for the input and output processing. As stated in section 2.3, timed loops are similar to while loops, but includes additional features for executing code deterministically. The input and output processing loop is part of a closed loop controller and is therefore considered time-critical. Variables that are part of the closed loop are configured as single-process shared variables with RT FIFO enabled.
A control and simulation loop is used to execute the hydraulic pump control models and a dynamic simulation model fora hydraulic circuit. The control and simulation loop, dynamic model and pump control models are described in section 3.3.6.

Communication to and from the host PC is done from a while loop. The communication rate between the cRIO and host PC can be set from the GUI at the host PC. By default it is set to 50 ms. Communication between these two targets are performed by means of network-published shared variables with RT FIFO deactivated. RT FIFO is deactivated because the communication between the cRIO and host PC is not part of a closed control loop, and no time-critical tasks are executed in the while loop or on the host PC. The while loop can be visualized in Appendix C, along with the documentation for the whole LabVIEW program.

In order to minimize the CPU load during execution all controls, indicators and clusters are hidden from the front panel on the main VI and subVIs on the cRIO. Visualizing the values from indicators and clusters at a high execution rate uses unnecessary computational power. Hiding these results in better system performance in terms of lower CPU load.

Configuring Host PC
On the host PC the GUI for the HIL system is localized. The main screen is divided into two parts, one for system model monitoring and one for settings. LabVIEW front panel is shown in Appendix B. The GUI is created for operators and instructors which uses the HIL system for experimental or educational purposes.

From the settings window various configurations related to the simulated hydraulic circuit and pump control models can be set. Possible configurations are listed below:

- Pressure cut-off setting
- Swivel angle command
- Pump size
- Leakage orifice opening on A- and B side
- Hydraulic volume on A- and B side
- Bias adjustment for valve
- Bias adjustment for pump
- Gain adjustment for pump

From the tab "Network setting" the execution rate for the communication loops between the cRIO and the host PC can be set.

3.3.4 Valve Output Reconfiguration

When the valve output is configured for ±10 V the pump cards functionality is simplified. Table 3.9 shows the terminal connections for solenoid A and solenoid B on pin `b2`, `b4`, `b6` and `b8`. These output terminals are deactivated if the valve output is configured for ±10 V. This again deactivates the "Zeropoint valve" adjustment possibility. The solenoids are actuated by pulse width modulation (PWM) signals. "Zeropoint valve" is adjusted by changing the PWM signals to the solenoids.

In order to add the possibility to adjust "Zeropoint valve" the PWM signals to the solenoids have to be analysed in terms of time period and on-time. The PWM signal is a 24VDC signal. Figure 3.14 shows an electronic circuit used to set a digital input on the cRIO high whenever 24V is applied to the solenoid.

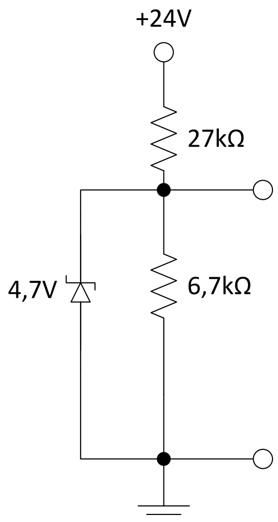


Figure 3.14: Electronic circuit used for PWM signal analysis

The figure above shows a classic voltage regulator circuit with a zener diode, rated at 4.7 V, in parallel with the output resistance. The zener diode makes sure the output voltage hold a constant voltage of 4.7 V, if the 24 V potentioal is high, regardless of the current. Two voltage regulator circuits, one for each solenoid, are needed in order to analyse the PWM signal to the solenoids.

A test VI was created in LabVIEW. The VI is used to test functionality of the PWM signal analysis circuit shown in Figure 3.15.

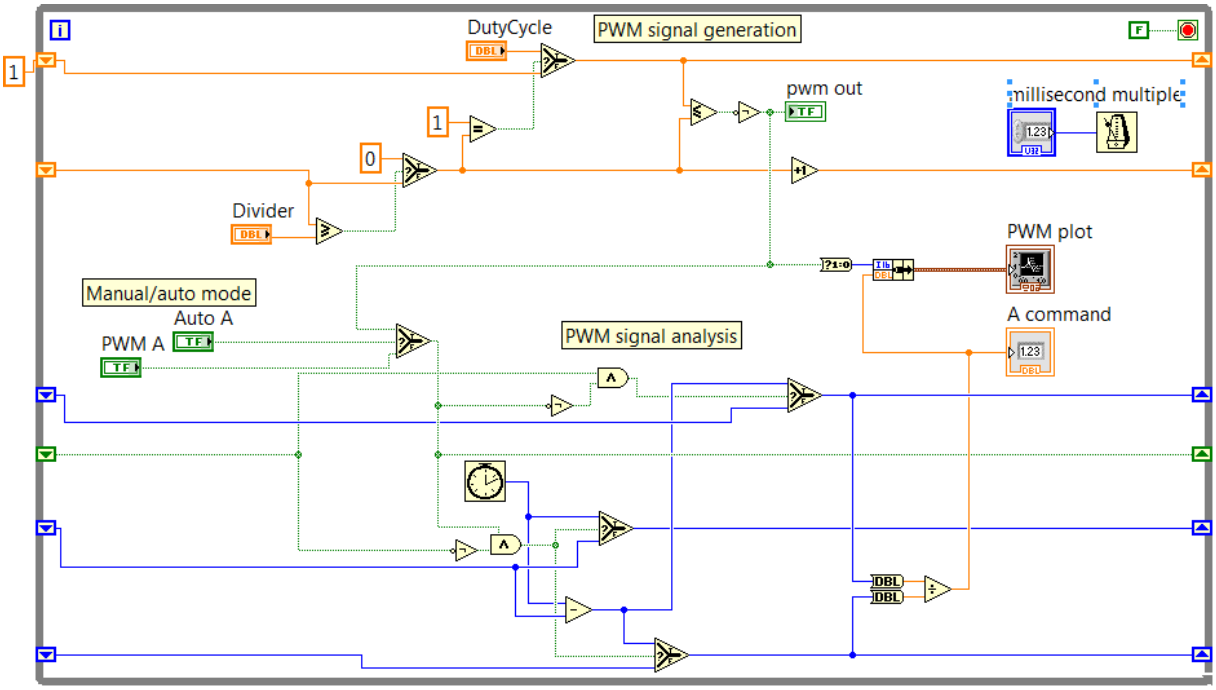


Figure 3.15: LabVIEW VI for PWM analysis

A PWM signal is generated at top of the while loop. The generated PWM signal is used as an input to the PWM signal analysis circuit. Computing the period time and the time the signal is high yields for a relative PWM value, between 0 and 1, when the on-time is divided by the period time. A plot showing a varying PWM signal and the resulting relative PWM value is shown in Figure 3.16.

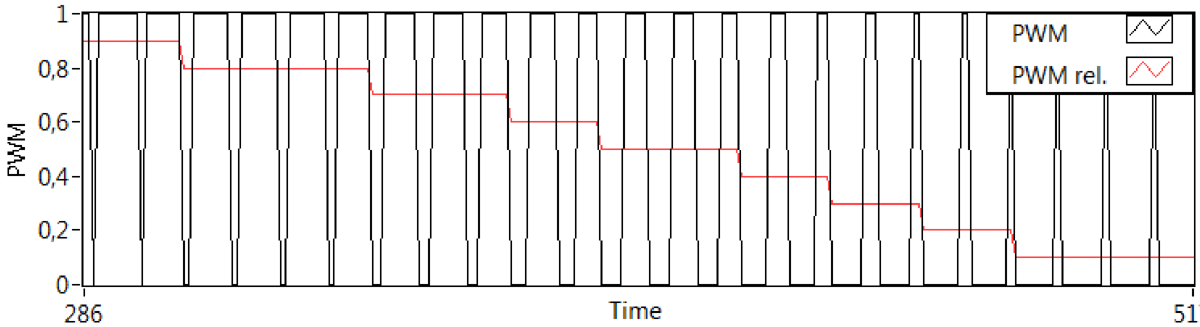


Figure 3.16: Relative PWM value plotted against PWM signal

If this PWM analysis circuit is to be used with the HIL system, a valve or power resistor has to be connected to the solenoid output terminals on the pump card. A wire is connected from `b2` and `b6` (see Table 3.9) to the +24 V terminal on the electronic circuit shown in Figure 3.14.
A test setup with the electronic circuit wired up on a breadboard was set up in order to test the PWM analysis VI on the real PWM signal to the solenoids. The program code for PWM analysis was implemented on the FPGA. Update rate for the digital I/O card, NI9401, is 100 ns. This update rate corresponds to 10 MHz resolution.

Results from the PWM analysis test yielded no sufficient results for the relative PWM value. No pattern was found for the time period, hence an unstable PWM signal was the result from the test. Future work in this area is discussed in section 4.3.

3.3.5 Model Estimation for Pump Control

Because the HIL simulation is based on simulation models of various A4VSG pump sizes, these simulation models have to be derived by estimating models for the proportional valve and axial piston cylinder for swash plate control.
By gathering technical data from the data sheets for the components which make up the hydraulic pump control, a transfer function for each of the pump controls can be estimated. The estimated models are verified by comparing bode plots and step responses.

Estimating a transfer function for both the valve and the axial piston cylinder for swash plate control and combining these transfer functions yields for a combined model for the hydraulic pump control. A block diagram of the principle is shown in Figure 3.17.

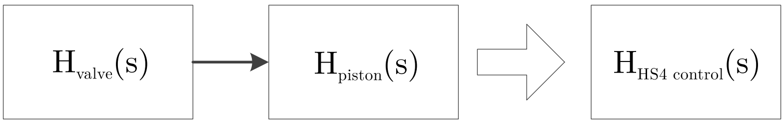


Figure 3.17: Merging transfer function models for valve and cylinder into pump model

For the HS4 control configuration the same proportional valve is used for pump sizes 250-, 355- and 500 cm^3 in the hydraulic control circuit. The valve is 4WRE6V16-2X/G24K4/V-822 from Bosch Rexroth. The valve is a 4/3 proportional directional valve with electrical position feedback with optional integrated electronics. Bode plot and step responses for this valve is shown in Figure 3.18 and Figure 3.19, respectively. The relation between positive and negative step response is assumed to be proportional for the valve.

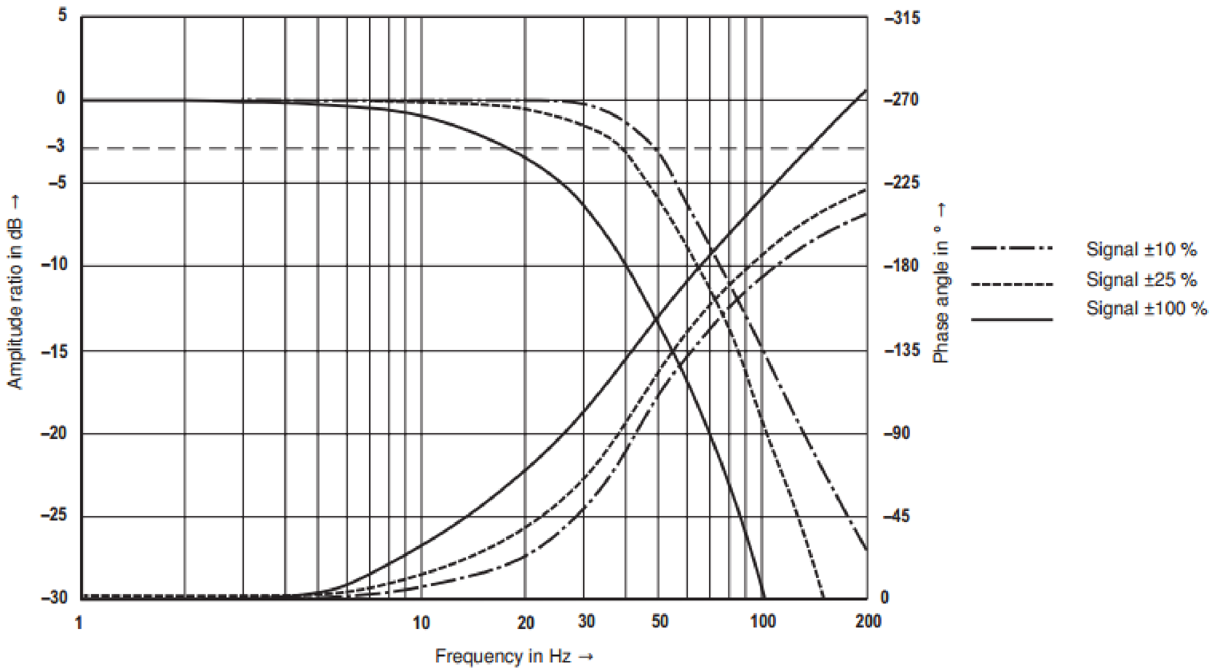


Figure 3.18: Bode plot of frequency response for 4WRE 4/3 proportional valve [16]

By analysing the bode plot and thereunder the phase it comes clear that a 3rd order transfer function could express the dynamics for the proportional valve. The estimated transfer function is based on the ±100% signal in the bode plot and a step response from 0 to 100%.

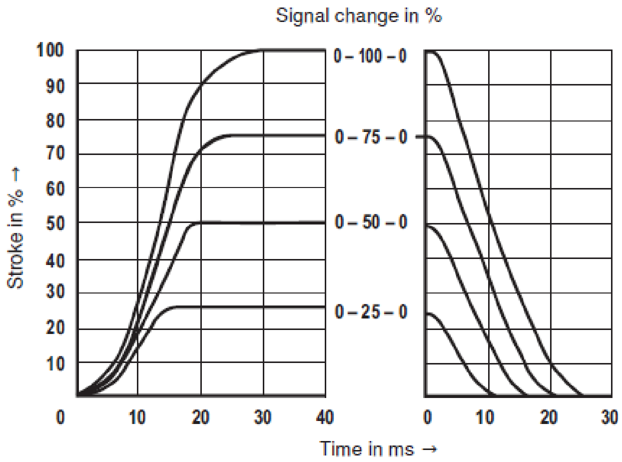


Figure 3.19: Transition function with stepped electric input signals for 4WRE 4/3 proportional valve [16]

There is no frequency or step response available in the data sheet for the axial piston cylinder. A step response for a A4VSG250 with HS4 control is visualized in the data sheet for control devices for A4VSG pumps, shown in Figure 3.20. Hence, by comparing the combined response to the response from the valve an estimation for the axial piston cylinder could be made.

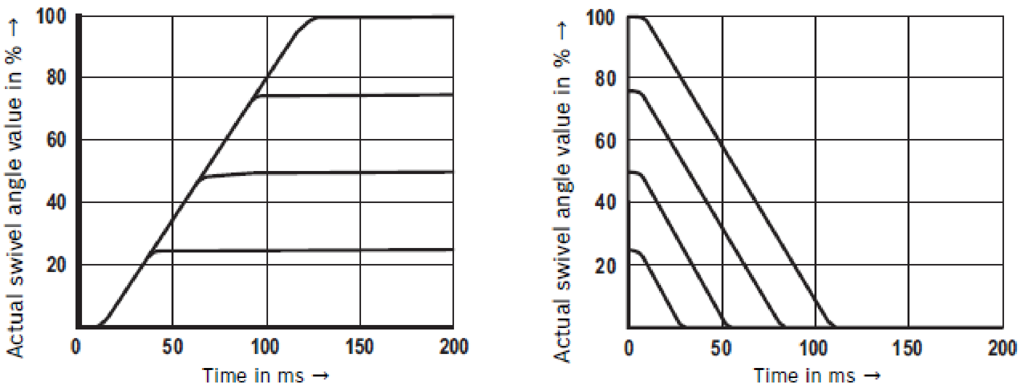


Figure 3.20: Step response example for A4VSG with HS4 control, size 250, actuating pressure $p = 125$ bar [11]

As seen from Figure 3.20 the rise time for a step response is approximately 0.12 s for a A4VSG250. The pump control is assumed to have a linear relation for positive and negative swivel angle control. In Table 3.11 technical data including control time for pump sizes 250-, 355- and 500 cm^3 are shown. Step response for the hydraulic control circuit is based on these control times.

Table 3.11: Technical data for HS4 control

Feature	**Variable**	**Unit**	**A4VSG250**	**A4VSG355**	**A4VSG500**
Displacement	$V_{g\,max}$	cm^3	250	355	500
Control pressure	p_{min}	bar	125	125	150
Control stroke	s_{max}	mm	25.9	25.9	32.6
Control area	A	cm^2	28.3	28.3	38.2
Control volume	$V_{s\,max}$	cm^3	73.2	73.2	124.5
Control time	t_{min} 1)	s	0.12	0.12	0.15
Maximum speed	n	rpm	2200	2000	1800

1) At minimum control pressure, p_{min}

Estimating Transfer Functions

MATLAB is used for system identification and estimation of the transfer functions. The estimated model

will be defined by two transfer functions; one for the proportional valve and one for the axial piston cylinder.
From Figure 3.18 it is determined that a 3rd order transfer function can define the proportional valve. The 3rd order transfer function can be estimated based on the phase plot in Figure 3.18, which approaches 270°. The axial piston cylinder could be expressed as a spring-mass-damper system, which yields for a 2nd order transfer function.
The transfer function for the valve is written on the form shown in Equation 3.2. Usually a transfer function for a proportional valve can by sufficiently expressed by a spring-mass-damper system, but in this case there is extra dynamics, which yields for a 3rd order transfer function.

$$H_{valve}(s) = \frac{1}{k_3 s^3 + k_2 s^2 + k_1 s + 1} \tag{3.2}$$

Where;
$k_1,\ k_2,\ k_3$ = Transfer function coefficients

Transfer function for the piston is written in the form shown in Equation 3.3. This transfer function depends on pump size. See Table 3.11 for physical dimensions of the components which makes up the HS4 control.

$$H_{piston}(s) = \frac{1}{\frac{1}{\omega_{piston}^2} s^2 + \frac{2\cdot\zeta_{piston}}{\omega_{piston}} s + 1} \tag{3.3}$$

Where;
ω_{piston} = Natural frequency for axial piston cylinder
ζ_{piston} = Damping coefficient for axial piston cylinder

By multiplying the transfer function for the valve and piston an estimate of complete hydraulic HS4 control can be made. When an estimate for the piston on a A4VSG500 pump is done, the complete transfer function could be found, as shown in Equation 3.4.

$$H_{HS4\,control}(s) = H_{valve} \cdot H_{piston} \tag{3.4}$$

In MATLAB, a toolbox called "System Identification Toolbox" was used to get an estimate for the transfer function for the valve. This toolbox can use both time response and frequency response data in order to estimate a transfer function or other types of models.
A data set for frequency, amplitude and phase was generated by sampling the data from the bode plot in Figure 3.18. By estimating a 3rd order transfer function with three poles and zero zeros, a preliminary model was created. Since this method did not give a sufficient result when comparing the bode plots, the estimate was improved by using a trail-and-error method. The final transfer function for the valve on the form of Equation 3.2 is shown below in Equation 3.5:

$$H_{valve}(s) = \frac{1}{7.316 \times 10^{-8}\, s^3 + 5.487 \times 10^{-5}\, s^2 + 0.013\,33\, s + 1} \tag{3.5}$$

In Figure 3.21 the step response for the estimated valve model is plotted. From MATLAB the settling time for the stepped response is found by using the `stepinfo` command. The `stepinfo` command takes the step response and a steady-state value and returns performance indicators such as rise time, settling time, overshoot, undershoot and more. With a 3% threshold the settling time is found to be 0.0291 s.

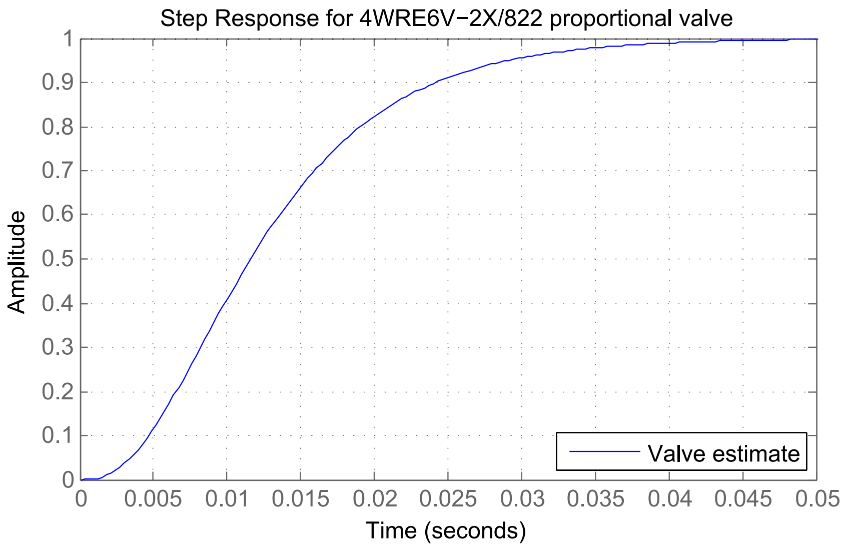


Figure 3.21: Step response for transfer function estimate proportional valve

In Figure 3.22 the bode plot for the estimated valve model is shown. The bode plot can be compared to the actual bode plot from the data sheet (see Figure 3.18) for model verification purposes.

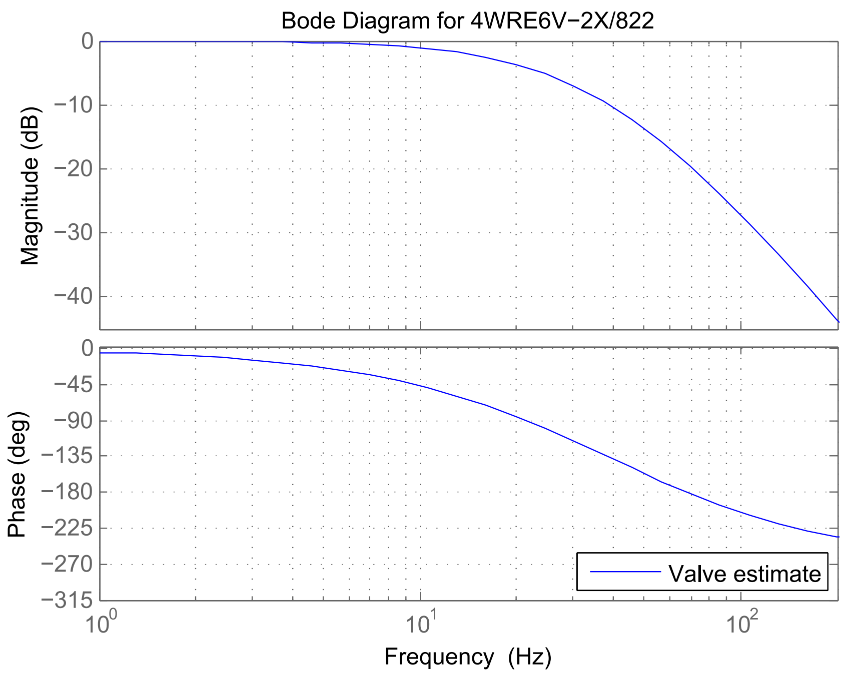


Figure 3.22: Bode plot for the estimated transfer function for the valve model

The estimated model for the axial piston cylinder was derived by first creating a 2[nd] order transfer function and comparing the step response shown in Figure 3.20 with the step response from this transfer function. When this transfer function fits the step response the model for the axial piston cylinder can be estimated by using Equation 3.4. Changing the parameters ω_{piston} and ζ_{piston} until the combined response for the valve and piston is equal to the step response for the 2[nd] order transfer function which again fits the response in Figure 3.20. In Equation 3.3 the natural frequency, ω_{piston}, is set to 43 rad/s and the damping coefficient, ζ_{piston}, is set to 1. The transfer function for the axial piston cylinder is shown in the Equation 3.6 below:

$$H_{piston\,500}(s) = \frac{1}{0.000\,540\,8\,s^2 + 0.046\,51\,s + 1} \tag{3.6}$$

Figure 3.23 shows the step response for the axial piston cylinder. The settling time, with 3% threshold, is 0.133 s.

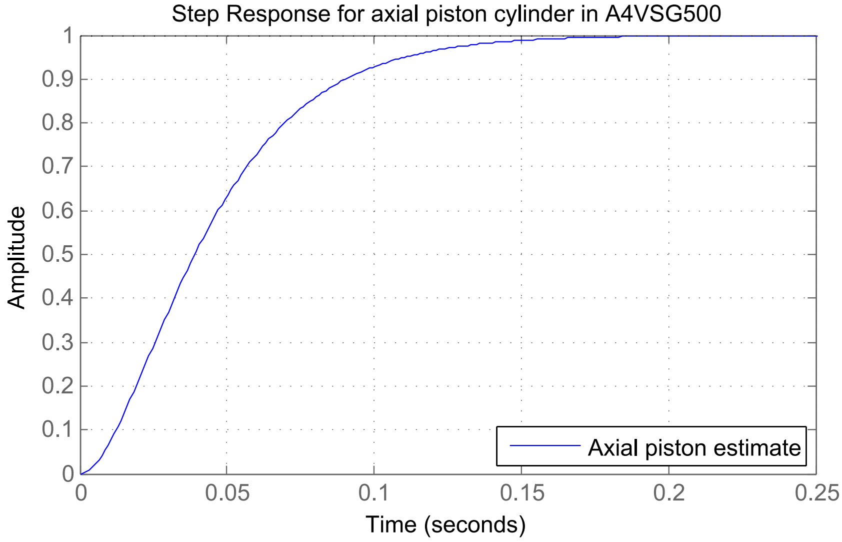


Figure 3.23: Step response for transfer function estimate axial piston

The combined response for the HS4 control can now be calculated. The open loop transfer function for the HS4 control of a A4VSG500 pump is shown in the Equation 3.7 below.

$$H_{A4VSG500}(s) = \frac{1}{3.957 \times 10^{-11}\, s^5 + 3.308 \times 10^{-8}\, s^4 + 9.836 \times 10^{-6}\, s^3 + 0.001\,216\, s^2 + 0.059\,84\, s + 1} \tag{3.7}$$

The control times in Table 3.11 can be compared to the step response of the HS4 control, shown in Figure 3.24. The settling time, with 3% threshold, is shown in the figure to be approximately 0.15 s. The control time in Table 3.11 is set to 0.15 s for a A4VSG500 pump.

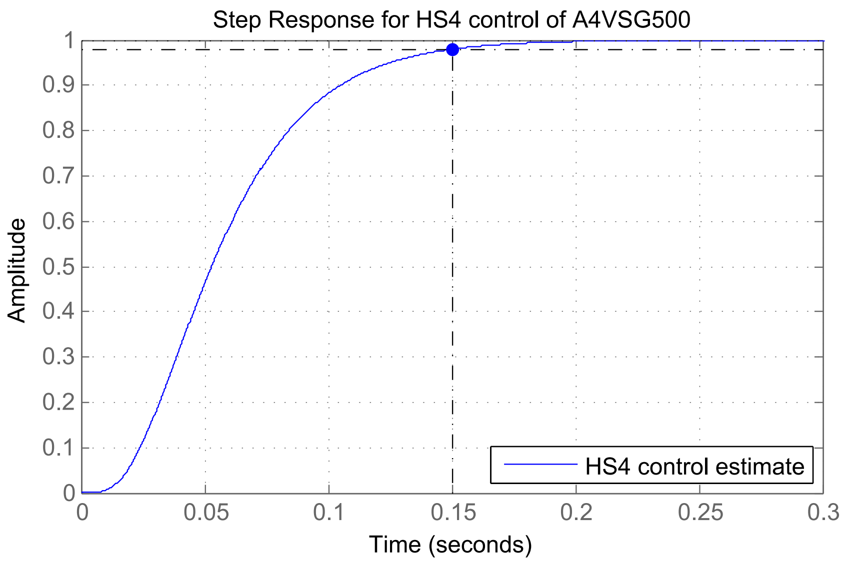


Figure 3.24: Transfer function estimate of combined transfer functions of valve and piston actuator

Figure 3.25 shows the combined step response of the valve and axial piston cylinder, as well as individual step responses for both the valve and axial piston cylinder.

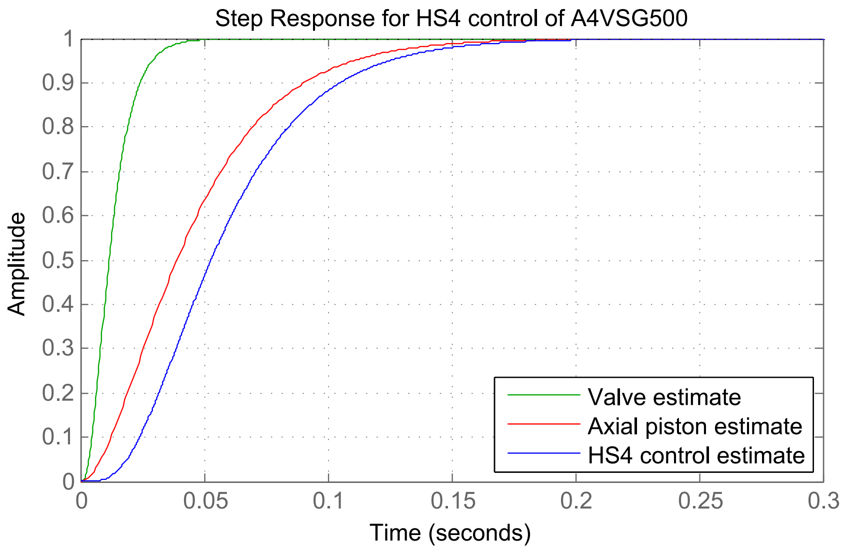


Figure 3.25: Transfer function estimate of combined and individual transfer functions for valve and piston actuator

For pump sizes 250- and 355 cm^3 the control time for HS4 control is the same, as shown in Table 3.11. Hence, the transfer function for the axial piston cylinder is equal for both these pump sizes. The transfer function for the valve is equal to pump sizes 250, 355 and 500 cm^3.
Using the same approach as for the A4VSG500 pump the combined open loop transfer function for pump sizes 250- and 355 cm^3 is found. The transfer function for the axial piston cylinder is shown in Equation 3.8. By combining transfer functions for the valve and piston cylinder the transfer function for HS4 control of A4VSG355/A4VSG250 is found shown in Equation 3.9.

$$H_{piston\,355/250}(s) = \frac{1}{0.000\,318\,9\,s^2 + 0.035\,71\,s + 1} \tag{3.8}$$

Compared to Equation 3.3 the variables ω_{piston} and ζ_{piston} are set to 56 rad/s and 1.0, respectively.

$$H_{A4VSG355/250}(s) = \frac{1}{2.333 \times 10^{-11}\,s^5 + 2.011 \times 10^{-8}\,s^4 + 6.284 \times 10^{-5}\,s^3 +} \cdots$$
$$\cdots \frac{1}{0.000\,849\,9\,s^2 + 0.049\,05\,s + 1} \tag{3.9}$$

Figure 3.26 shows the step response for valve, axial piston cylinder and combines HS4 control response. The settling time, with 3% threshold, is found to be 0.1194 s. In comparison Table 3.11 states that the control time is 0.12 s for both A4VSG250 and A4VSG355.

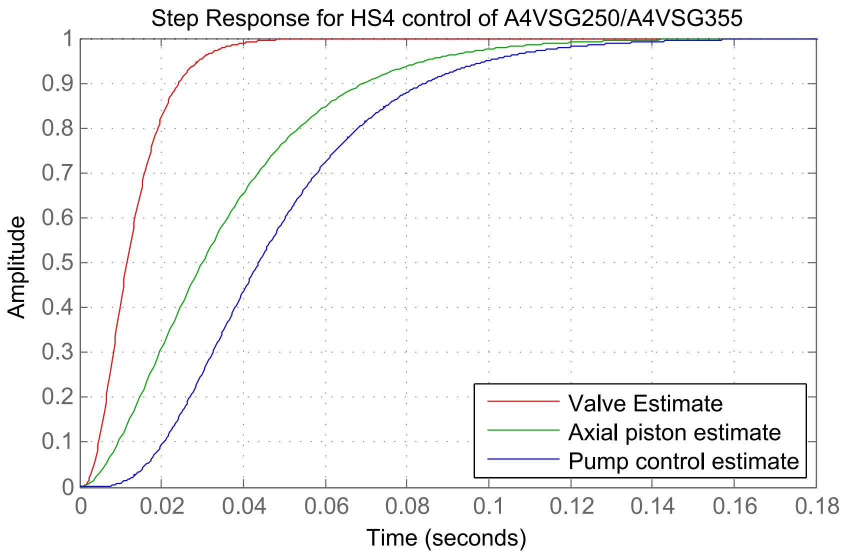


Figure 3.26: Transfer function estimate of combined and individual transfer functions of valve and piston actuator

The assumed equal control time for pump sizes 250- and 355 cm^3 is a simplification in terms of system dynamics. With equal transfer functions describing the hydraulic control the only difference between the pump sizes 250- and 355 cm^3 is maximum speed and output flow.

3.3.6 Dynamic Simulation Modelling

A hydraulic circuit has to be simulated in order to generate pressures and close the loop for the HS4 control. The HS4 control requires feedback from valve position, swivel angle position and pressures on A- and B side for the pump.
NOV uses a leakage circuit as part of a pre-tuning process. The hydraulic diagram for this leakage circuit is shown in Figure 3.27. A feeder supply keeps the pressure level at a minimum of 25 bar. The variable orifices represents the leakage for both the A- and B side. By adjusting the orifice opening area the leakage flow can be adjusted. The shuttle valve for the feeder supply is considered ideal. Hence, no pressure drop or flow restrictions caused by the shuttle valve.

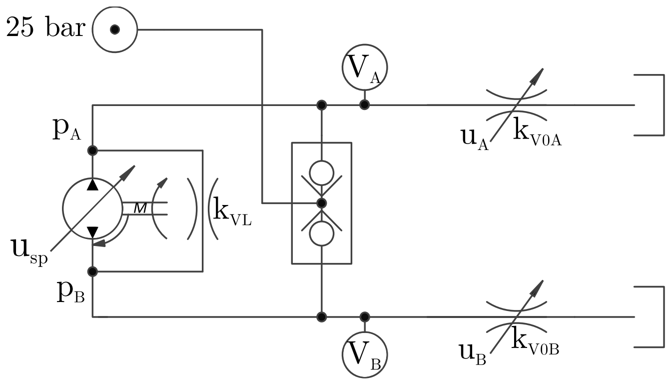


Figure 3.27: Leakage circuit

Mathematical Model

The hydraulic circuit can be expressed by mathematical expressions for each component and pressure node. The process of setting up a dynamic mathematical simulation model is described in the following steps [15]:

1. Determine all continuity- and restriction flows in the system.
2. Identify all volumes in the system and set up the pressure build up equation for each.

3. Identify all orifices and their dependency on the motion of mechanical parts in valves.
4. Determine initial pressure for each volume and initial fluid volume for each accumulator.
5. Calculate pressure gradients for each volume.
6. Update pressure in each volume and volume of each accumulator.
7. If the analysis is not yet concluded then go back to 5.

Starting from step 1 by determining four individual flows in the system. These are the pump flow, pump leakage flow and leakage flows for A- and B side. The flows are denoted Q_P, Q_L, Q_A and Q_B, respectively. The equations for pressure build up for each side of the pump is shown in Equation 3.10 and 3.11.

$$\dot{p}_A = \frac{\beta}{V_A}(-Q_P - Q_L - Q_A) \tag{3.10}$$

$$\dot{p}_B = \frac{\beta}{V_B}(Q_P + Q_L - Q_B) \tag{3.11}$$

Where;

$\dot{p}_A$, $\dot{p}_B$	=	Pressure gradients for A- and B side	]Pa/s]
β	=	Bulk modulus for hydraulic fluid	]Pa]
V_A, V_B	=	Volume for A- and B side	]m^3]
Q_P, Q_L, Q_A, Q_B	=	Pump flow, pump leakage flow, leakage flow A side, leakage flow B side	]m^3/s]

In Equations 3.12 - 3.15 the flows are calculated.

$$Q_P = u_{sp} \cdot D \cdot n \tag{3.12}$$

Where;

u_{sp}	=	Relative swash plate position, in range [-1 1]	[-]
D	=	Displacement	[m^3]
n	=	Speed	[rev/s]

$$Q_L = k_{VL} \cdot SIGN(p_A - p_B) \cdot \sqrt{|p_A - p_B|} \tag{3.13}$$

Where;

k_{VL}	=	Orifice constant	[$\frac{\text{m}^3\,\text{s}^{-1}}{\sqrt{\text{Pa}}}$]
p_A, p_B	=	Pressures for A- and B side	[Pa]

$$Q_A = k_{V0A} \cdot u_A \cdot \sqrt{p_A} \tag{3.14}$$

$$Q_B = k_{V0B} \cdot u_B \cdot \sqrt{p_B} \tag{3.15}$$

Where;

k_{V0A}, k_{V0B}	=	Orifice constant with max opening	[$\frac{\text{m}^3/\text{s}}{\sqrt{\text{Pa}}}$]
u_A, u_B	=	Orifice control, in range [0 1]	[-]

The initial conditions in the system is set based on Equation 3.16 and 3.17. These conditional equations also makes sure the pressure never comes below the level of the feeder pressure, which is 25 bar.

$$p_A = \begin{cases} 25 \text{ bar}, & \text{if} \quad p_A < 25\text{bar} \\ p_A, & \text{otherwise} \end{cases} \tag{3.16}$$

$$p_A = \begin{cases} 25 \text{ bar}, & \text{if} \quad p_B < 25\text{bar} \\ p_B, & \text{otherwise} \end{cases} \tag{3.17}$$

The pressures are updated by means of numerical integration as shown in Equation 3.18 and 3.19.

$$p_A(t+1) = p_A(t) + \dot{p}_A \cdot \text{dt} \tag{3.18}$$

$$p_B(t+1) = p_B(t) + \dot{p}_B \cdot \text{dt} \tag{3.19}$$

The size for the orifices on A- and B-side and the orifice across the pump has to be determined. Equation 3.20 shows the orifice equation for calculating the flow through an orifice.

$$Q_{orifice} = c_D \cdot A_D \sqrt{\frac{2 \cdot \Delta p}{\rho}} \tag{3.20}$$

Where;

$Q_{orifice}$	= Flow through orifice	[m^3/s]
c_D	= Discharge coefficient	[-]
A_D	= Orifice opening area	[m]
Δp	= Pressure drop across orifice	[Pa/s]
ρ	= Hydraulic fluid density	[kg/m^3]

Comparing Equation 3.20 to Equation 3.13 - 3.15 shows that the k_{VL} can be written as Equation 3.21

$$k_{VL} = c_D \cdot A_D \sqrt{\frac{2}{\rho}} \tag{3.21}$$

The flow through the pump leakage orifice is determined by the size of the orifice constant, k_{VL}, and the pressure drop across this orifice. At maximum flow and 300 bar pressure drop the leakage flow across the pump is set to 1% of maximum pump flow. Equation 3.12 is used to calculate maximum flow for an A4VSG500 pump. Maximum speed and displacement for the pump is found in Table 3.11.

$$Q_P = 1 \cdot 0.5[dm^3] \cdot 1800[rpm] = 900[l/min]$$

Hence, the pump leakage flow with a 300 bar pressure drop across the orifice is 9 l/min, or in SI units, 0.000 15 m^3/s. Solving for k_{VL} in Equation 3.13 creates an equation for the orifice constant, shown in Equation 3.22.

$$\begin{aligned} k_{VL} &= \frac{Q_{L\,300bar}}{\sqrt{\Delta p}} \\ k_{VL} &= \frac{0.00015}{\sqrt{300 \times 10^5}} = 2.7386 \times 10^{-8} \frac{\mathrm{m^3/s}}{\sqrt{\mathrm{Pa}}} \end{aligned} \tag{3.22}$$

Leakage orifices for A- and B side has a rated flow at 30 l/min, 0.0005 m^3/s, with 300 bar pressure drop across the orifice. Using Equation 3.23 to calculate the orifice constants.

$$\begin{aligned} k_{V0A} = k_{V0B} &= \frac{Q_{A\,300bar}}{\sqrt{\Delta p}} \\ k_{V0A} = k_{V0B} &= \frac{0.0005}{\sqrt{300 \times 10^5}} = 9.1287 \times 10^{-8} \frac{\mathrm{m^3\,s^{-1}}}{\sqrt{\mathrm{Pa}}} \end{aligned} \tag{3.23}$$

LabVIEW Block Diagram

The dynamic simulation model was set up in LabVIEW using a Control & Simulation loop from the Control Design & Simulation module. The step size for the simulation loop is set to 5 ms, using the Runge-Kutta 1 solver. This solver is a first order differential equation solver based on the Euler Method.

In order to determine the actual swivel angle for the pump shown in Figure 3.27 the transfer functions for the valve and axial piston cylinder for each of the pump types are placed inside a case structure. By changing case the pump type is changed. The case structure for pump select is shown in Figure 3.28.

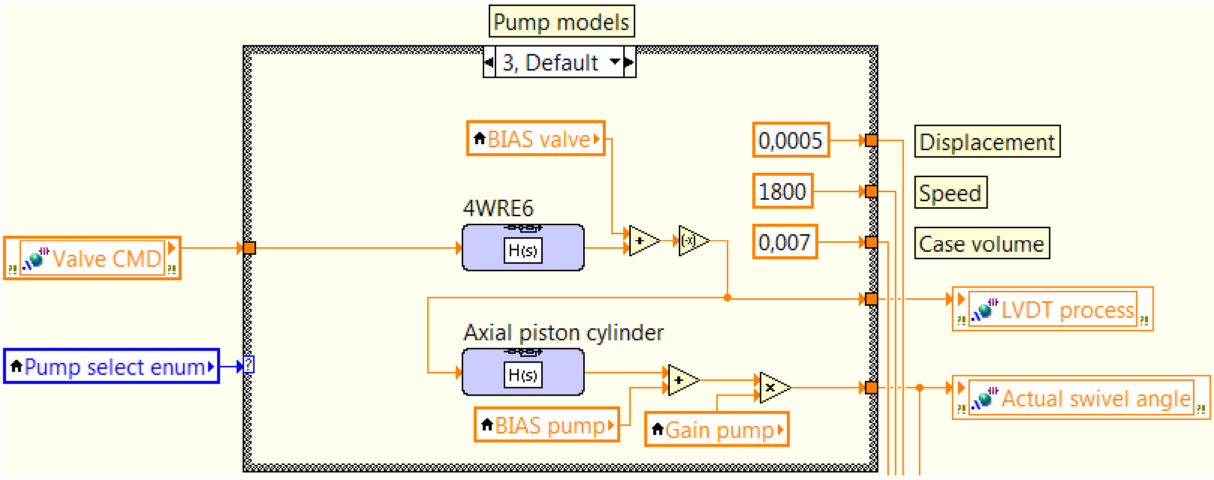


Figure 3.28: LabVIEW Block diagram. Simulated valve and axial piston cylinder control.

The actual swivel angle variable output from the case structure can be used to calculate the flow generated from the pump, Q_P. Figure 3.29 shows the LabVIEW block diagram for calculating the flows Q_P, Q_L, Q_A and Q_B. All variables are calculated in SI units and converted into hydraulic units for visualization.

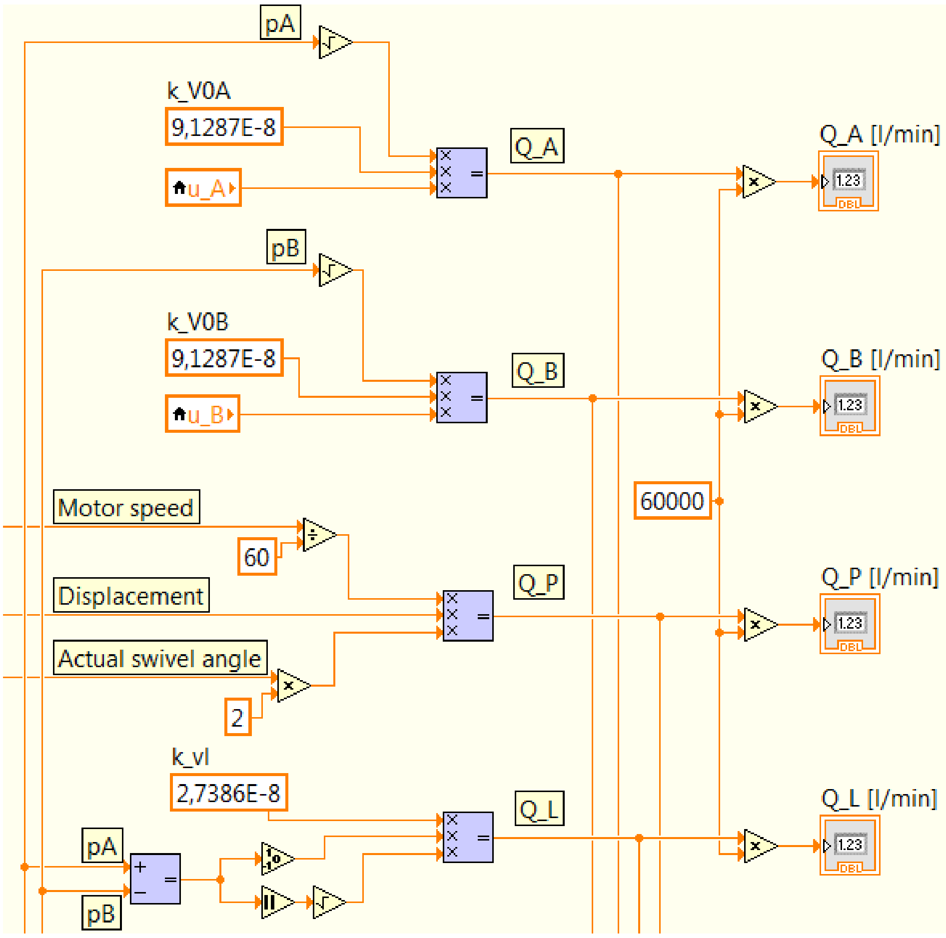


Figure 3.29: LabVIEW Block diagram. Flow calculations.

In Figure 3.29 the pressures p_A and p_B are required to calculate the flows Q_L, Q_A and Q_B. The block diagram for calculating these pressures is shown in Figure 3.30. The pressure gradients are calculated by using Equation 3.10 and 3.11. Both integrators are initialized to 2 500 000 Pa, which equals 25 bar. A lower limit is also set to 2 500 000 Pa in order to fully define Equation 3.16 and 3.17.

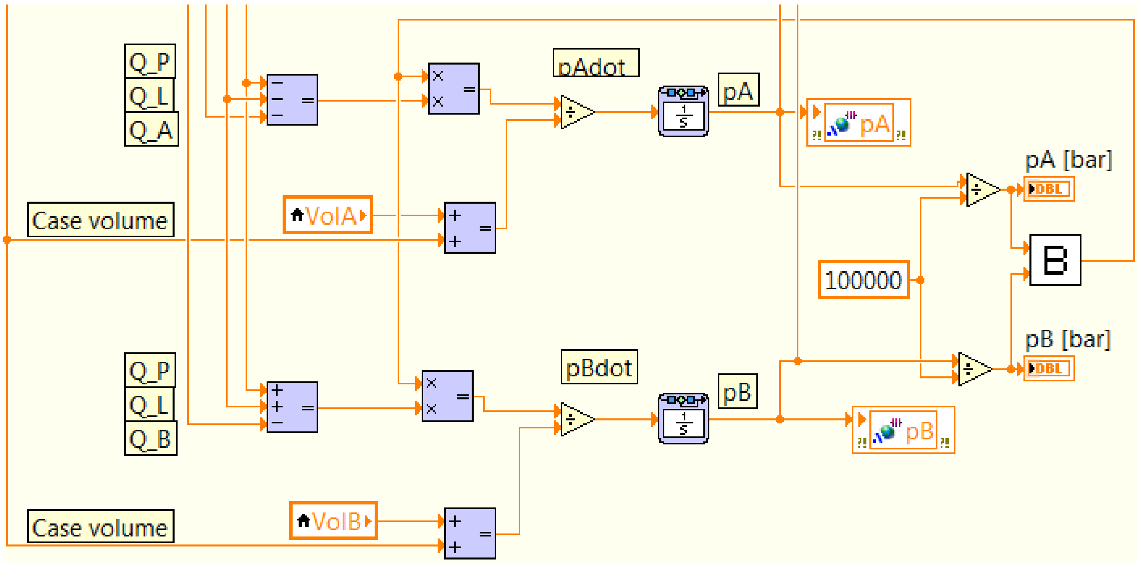


Figure 3.30: LabVIEW Block diagram. Calculation of pressure gradients and pressures.

3.3.7 External Simulation Model

Investigating complex hydraulic circuit causes the process of setting up the governing equations for a dynamic simulation a time consuming process. Modelling the hydraulic circuit in SimulationX and exporting a .dll file simplifies this process.

As stated in section 3.2 the cRIO-9075 does not support .dll files. The cRIO VxWorks operating system can execute .out shared libraries by using the Call Library Function Node in LabVIEW. An .out file can be compiled from C-code using the GNU Compiler Collection (GCC). The process of compiling an .out file with GCC is a thorough process. The process from modelling the hydraulic simulation model in SimulationX to executing the same simulation model on the RT target is described in the list under [17,18]:

1. Model the hydraulic system in SimulationX. Keep in mind the control signals from the pump card and required feedback from the model.
2. Export C-code from SimulationX using Microsoft Visual C/C++ Express Edition compiler.
3. Compile an .out file from the exported C-code using GCC compiler with VxWorks distribution.
4. Upload the .out file to cRIO using file transfer protocol (FTP).
5. Call the on the Rt target using a Call Library Function Node.
6. Configure inputs, outputs and parameters equal to the function and parameter names in the exported C-code.
7. Wire the inputs, outputs and parameters in LabVIEW and execute code.

Compiling an .out files was considered to complex in terms of time consumption. The HIL system should have the possibility for rapid swapping of simulation models. Also, the HIL system should have a versatile architecture without required knowledge of several programming languages in order to swap simulation model. To simplify the process of implementing an external simulation model, an external RT target with .dll file support is used. The external RT target is described in section 3.3.8.

3.3.8 HIL System With External Real-Time Target

In order to increase the computational power and open the possibility to run computationally complex hydraulic simulation models an external RT target is implemented. In this configuration the cRIO acts as an I/O processing module. A network switch is be used to set up the communication between the external RT target, the cRIO and a host PC. An overview of the HIL system with an external RT target is shown in Figure 3.31.

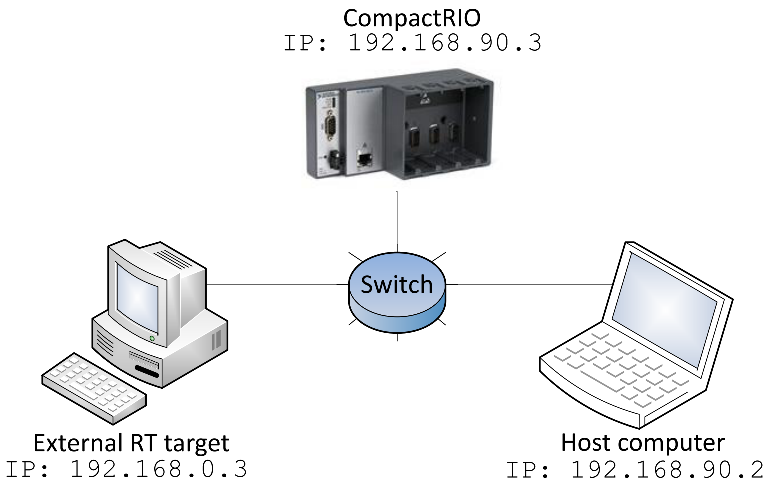


Figure 3.31: HIL system overview with an external RT target

The program structure is set up such that the cRIO handles communication to host PC and external RT target. The control and simulation loop which previously executed on the cRIO is deployed and executed on the external RT target, releasing some of the computational power for the cRIO. A detailed visualization of the HIL systems architecture with VI tasks is shown in Figure 3.32.

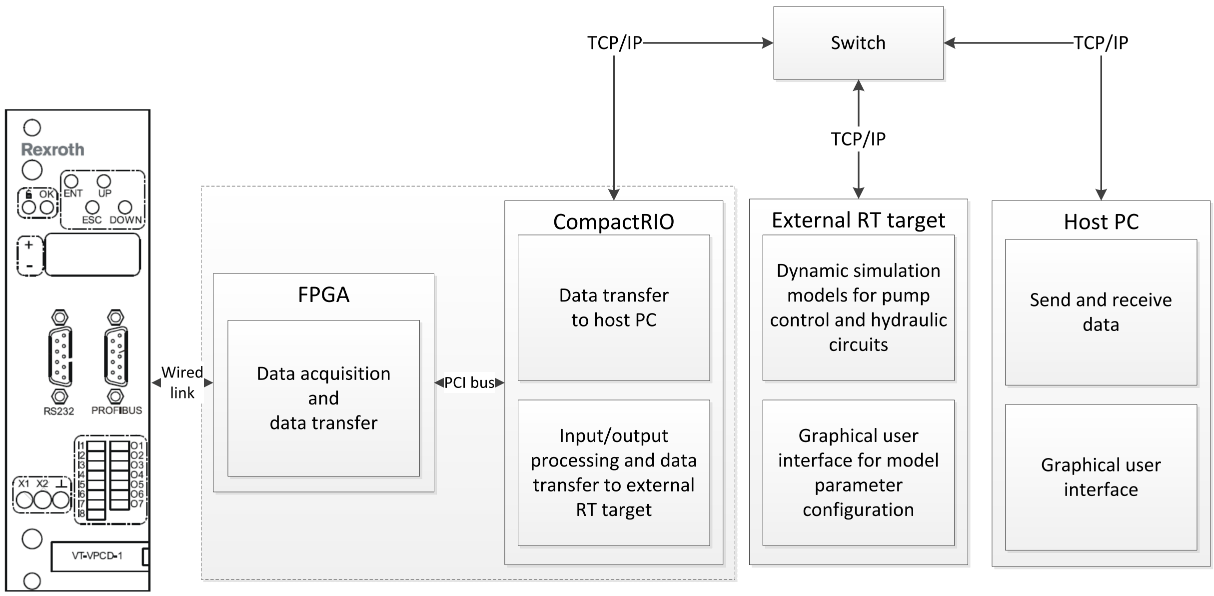


Figure 3.32: HIL system overview showing VI execution tasks

Maintaining a deterministic closed loop operation with Ethernet communication can be troublesome. Communication between the cRIO and the external RT target makes up the closed loop system. Network published shared variables with RT FIFO enabled is used for sending and receiving data between the two targets. The RT FIFO option yields for deterministic communication from the cRIO to the Shared Variable Engine and from the Shared Variable Engine to the external RT target and the other way around. Uncertainties lies within the TCP/IP communication, which is considered non-deterministic. Network published shared variables are executed even though the variable is updated or not since last iteration. Executing a shared variable with the value from last iteration causes and error to be written to the error output terminal. Hence, by wiring the error terminal from the shared variables on the external RT target and the cRIO the determinism of the closed loop can be determined. A communication rate at $1\,ms$ for the VIs included in the closed loop did not cause errors. Hence, the closed loop executed in a deterministic manner. System performance for the HIL system with external RT target is discussed in section 3.4.3.

The external RT target used in this setup is a conventional desktop computer with a network card from National Instruments. The PC is booted from an USB stick which is configured for LabVIEW Real-Time. The whole process of converting a desktop PC to a LabVIEW Real-Time target is described in [19]. This setup can execute .dll files through LabVIEWs Control Design and Simulation Module by using External Model Interface. This again opens the possibility for rapid integration of exported simulation models from SimulationX to be used with the HIL system. LabVIEW documentation for this HIL system is included in Appendix D. The LabVIEW program is configured with the same dynamic simulation model as for the HIL system without the external RT target.

3.3.9 Stand-Alone Application

Creating a stand-alone application ease the use for the HIL system. A stand-alone application does not require the user to have LabVIEW installed on the computer used to interface with the HIL application. To make the application independent from LabVIEW a build specification for the real-time target has to be created. This real-time application must be configured as a start-up application, causing the application to start executing when the cRIO is powered up. Figure 3.33 shows the application window used to create a build specification for a real-time application. Operations describing how the stand-alone application was created is listed below:

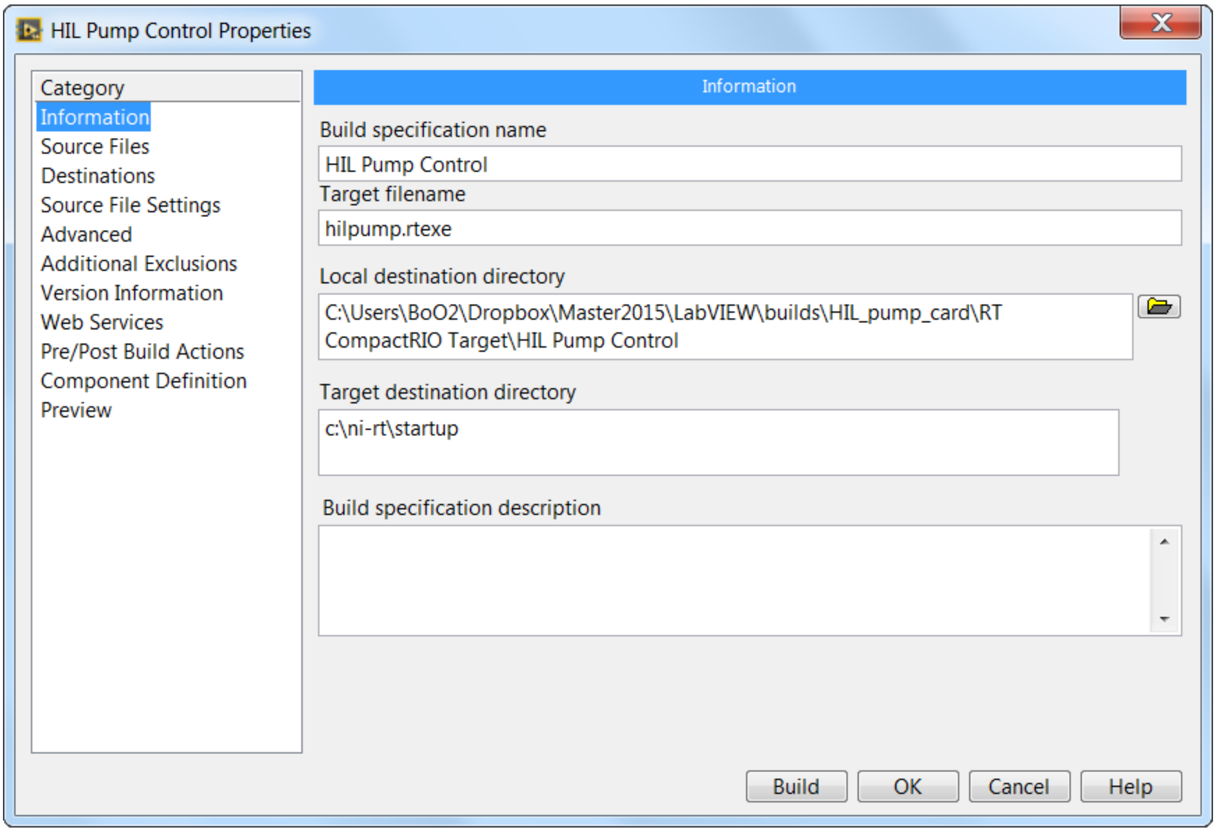


Figure 3.33: Creating a build specification for a real-time application

1. Right-click **Build Specifications** under RT target and select **New - Real-Time Application**. Chose suitable names for build specification and start-up file.
2. Under **Source Files** select the top-level VI as a Startup VI. The top-level VI for the real-time application is "cRIO MAIN.vi".
3. All remaining settings are used as default. Click **Build** to create the build specification.
4. Right-click the build specification and chose "Set as startup".
5. Right-click the build specification and deploy to the real-time target.

6. Right-click **Build Specifications** under host PC and select **New - Application(EXE)**. Chose a suitable name for the application and select destination folder.
7. Under **Source Files** select the top-level VI for the host application. The top-level VI for the host application is "HOST MAIN.vi".
8. All remaining settings are used as default. Click **Build** to create the EXE application.
9. Right-click **Build Specifications** under host PC and select **New - Installer**. Chose a suitable name for the install files and select destination folder.
10. Under **Source Files** select the application created in step 8.
11. All remaining settings are used as default. Click **Build** to create the install files.

A user manual describing how to operate the HIL system is located in Appendix E. The stand-alone application is created for the HIL system with the cRIO, not for the HIL system with external RT target.

3.4 Validation and Testing

In this section the HIL system is validated by means of signal range deviation, real-time performance and system performance. The application areas for the HIL system is presented, including the process of tuning a simulated pump control using Bodac.

3.4.1 Validation of I/O Signals

Due to the HIL system signals are processed through the cRIOs I/O voltage modules the signals have to be validated. A command value in LabVIEW has to give the same command value in Bodac to the pump card and visa versa. All the I/O signals are scaled because all I/O modules are calibrated in raw data mode. These scaled values has to be verified whether the values are equal or unequal on the pump card and cRIO controller.

The 16 bit I/O modules yields for 2^{16}, or 65536, different raw data values. Scaled from -10V to 10V the values are in the range between -32768 and 32767. A raw data value of 32767 does not correspond to 10V because the full range is typically from -10.4V to 10.4V. Hence, all I/O signals has to be scaled individually.

The input signal calibration to the pump card the is done by comparing the value set in LabVIEW to the value read from the analogue inputs on the pump card and visualized in Bodac. For output signals from the pump card the calibration is done by setting the controller in manual and comparing values set in Bodac with values read in LabVIEW. If the value does not correspond to each other, the scaling constant is changed until acceptable result is obtained. Table 3.12 shows the validation of input signals to the pump card.

Table 3.12: Scaled and calibrated I/O signals

Pressure CMD [bar]		Pressure A [bar]		Pressure B [bar]		Angle CMD [%]		Actual angle [%]	
LabVIEW	Bodac	LabVIEW	Bodac	LabVIEW	Bodac	LabVIEW	Bodac	LabVIEW	Bodac
0	1.3	0	0.8	0	0	-100	-100.0	-100	-100.0
50	50.8	50	50.3	50	49.6	-80	-80.1	-80	-80.1
100	100.6	100	100.2	100	99.6	-60	-60.3	-60	-60.0
150	150.5	150	150.1	150	149.7	-40	-40.1	-40	-40.0
200	200.4	200	199.9	200	199.5	-20	-20.1	-20	-20.1
250	250.3	250	249.8	250	249.6	0	-0.2	0	-0.1
300	299.8	300	299.5	300	299.3	20	19.9	20	19.9
350	349.7	350	349.4	350	249.3	40	39.9	40	39.8
400	399.6	400	399.3	400	399.5	60	60.0	60	60.0
						80	80.0	80	79.9
						100	100.0	100	99.8

Analogue I/Os are configured in Bodac, see Figure 3.6. The signal range and type of source is set here. In the pressure command, A pressure and B pressure a DC voltage at 10V corresponds to a pressure level at 400 bar. In Table 3.13 the scaling constants which represents the full range of the signal is shown.

Table 3.13: Scaling constants for pump card input signals

I/O signal	Scaling constant
Pressure CMD	30510
Pressure A	30540
Pressure B	30640
Angle CMD	30680
Actual angle	30600

The same process is done for the output signals. The valve command signal is set in Bodac and read in LabVIEW for comparison. Table 3.14 shows the calibrated data for the valve command signal. The scaling constant used to obtain the results shown in the Table 3.14 is 31300.

Table 3.14: Scaled and calibrated valve CMD signal

Valve CMD [%]	
Bodac	LabVIEW
-100	-99.9
-80	-80.1
-60	-60.0
-40	-40.2
-20	-20.3
0	0.2
20	19.8
40	39.9
60	59.9
80	80.0
100	100.0

In order to verify the LVDT process signal the pump card controller is set inactive in Bodac from Steuer menu. Hence, the valve stroke can be adjusted manually. By checking the relative actual valve position in Bodac against the manually set valve command in Bodac the LVDT process signal can be verified. The ideal working principle of the simulated LVDT signal is previously described in section 3.3.3, and Figure 3.12 showing the ideal LVDT signals to and from the pump card. The LVDT reference signals resolution is limited due to the sampling rate for the I/O modules on the cRIO. From Equation 3.1 the number of samples for one waveform period is found to be 20 samples for I/O modules rated at 100 kS/s. Hence, the resolution for the signal is not ideal and possible reasons for signal deviation from actual value is introduced. Table 3.15 shows the scaled data, read from Bodac.

Table 3.15: Scaled and calibrated signals from simulated LVDT

Valve CMD [%]	Actual valve [%]
Bodac	Bodac
-100	-98.9
-80	-82.1
-60	-62.7
-40	-42.2
-20	-20.8
0	0.0
20	19.8
40	41.4
60	62.2
80	81.4
100	98.5

The scaling constant used for the LVDT process signal was determined by replacing the LVDT process with a control in LabVIEW. By manually setting the value in LabVIEW and reading the corresponding value in Bodac the signal could be proper scaled. The scaling constant for the LVDT process signal is set to 30000.

Based on experimental values the scaling constant for the LVDT reference signal is set to 19200. The scaling constant has a lower value compared to the other scaling constants due to the lower expected signal range for the pump card. Hence, for a real LVDT the induced voltage in the secondary coils are lower than the voltage across the main coil. Scaling constants for valve command and LVDT signals are shown in Table 3.16.

Table 3.16: Scaling constants for pump card output signals

I/O signal	Scaling constant
Valve CMD	31300
LVDT process	30000
LVDT reference	19200

3.4.2 Pump Tuning

One of the main purposes of the proposed HIL system is to perform a pump tuning in Bodac. In order to cover all parts of the pump tuning process the simulation model has to contain inaccuracies, as in real life. In the dynamic simulation model these inaccuracies are added in the form of bias for valve, bias for pump and gain factor for pump. See Figure 3.28 for LabVIEW block diagram of the inaccuracies added to the dynamic simulation model.

When configuring a pump card in Bodac requires the user to use a variety of different application windows. An overview of the menu architecture in Bodac is shown in Figure 3.34. The figure does not include all menu windows, but all application windows required to perform a pump tuning are listed.

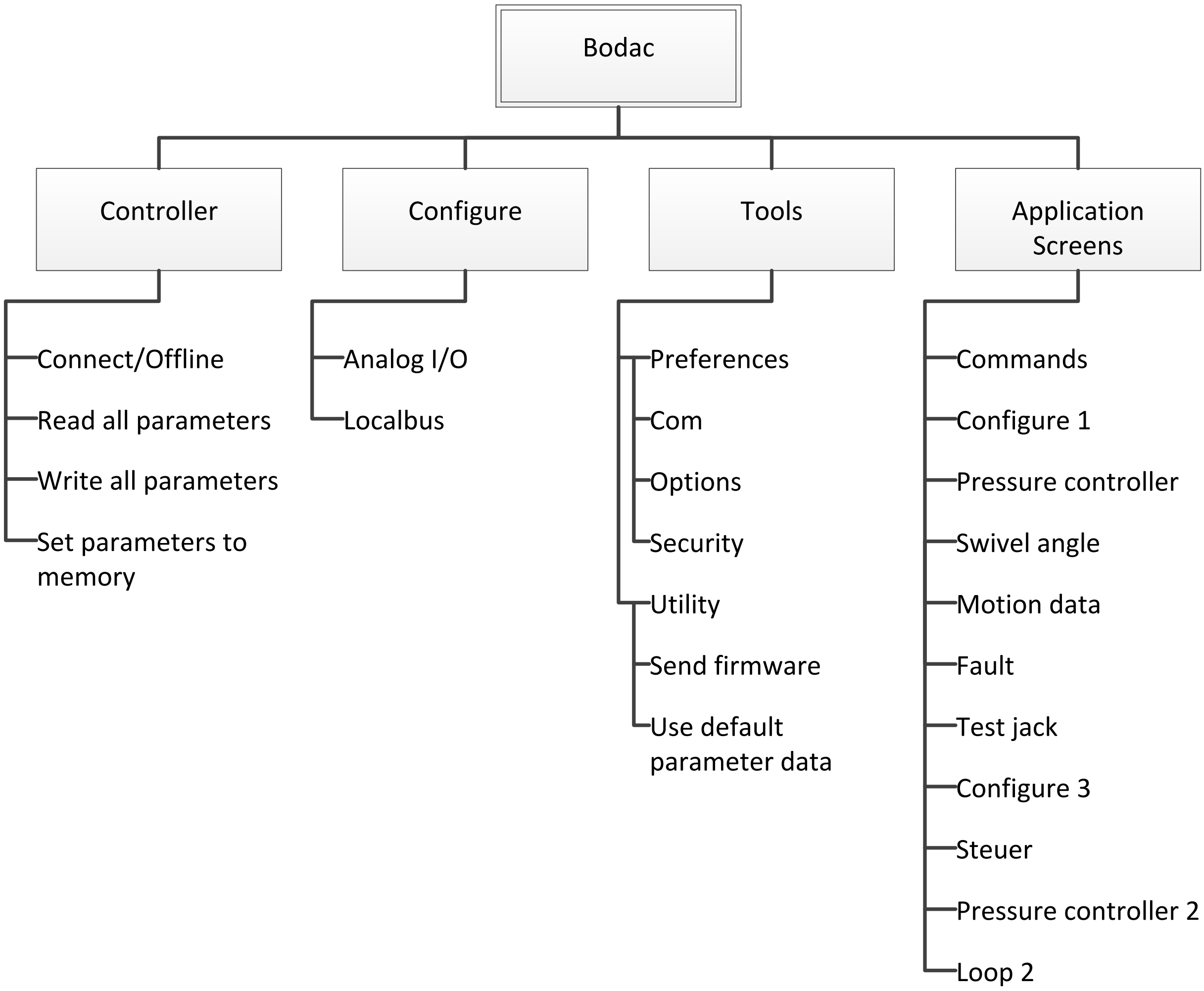


Figure 3.34: Tree architecture of the menu in Bodac with the most important features

NOV has a pump tuning procedure to tune the pump card in Bodac, which is described in the list below:

1. Update firmware to wanted version.
2. Loop all sensors and determine that the wiring is correct.
3. Configure analogue I/Os.
4. Set command values for pressure command, power command, angle ramp and pressure ramp to valve and pump configuration parameters.
5. Chose pump type, pump size and feedback sensors.
6. Set pressure controller to wanted setting.
7. Chose which fault scenarios that should stop the pump.
8. Configure test jacks.
9. Configure settings for fast I-correction.
10. Coarse adjustment of the proportional valve.
11. Start pump.
12. Fine adjustment of proportional valve and pump.
13. Set command values for pressure command, power command, angle ramp and pressure ramp to regular parameters.

HIL simulation yields for a simplified pump tuning due to the required precautions that has to be taken into account when tuning a pump in real-life environment. Safety considerations related to oil spillage and other scenarios related to tuning of a physical pump does not apply when tuning to pump card for HIL simulation.

The idea when performing a pump tuning for educational purposes is that the instructor engineer initializes the LabVIEW program with wanted settings for the HIL simulation from a host computer connected to the cRIO via TCP/IP. The person being educated connects another computer, where Bodac is installed, to the pump card with a RS232-to-USB cable. Hence, a pump tuning sequence can be performed with many of the same preconditions for a physical pump tuning. A user manual for the HIL system is found in Appendix E.

Step-by-Step Pump Tuning with HIL System

A step-by-step guide describing the process of configuring a Bosch Rexroth VT-VPCD pump card is presented in the following steps. A description of menu bar functions is shown in Figure 3.35. Use increment and decrement parameter to change parameter settings. Settings are then written to the pump cards memory. If the pump card looses its power the parameter settings are lost. In order to permanently write all parameters to the pump card, go to "Controller/Set Parameters To Memory". This will burn the settings to the pump cards erasable programmable read-only memory (EPROM).

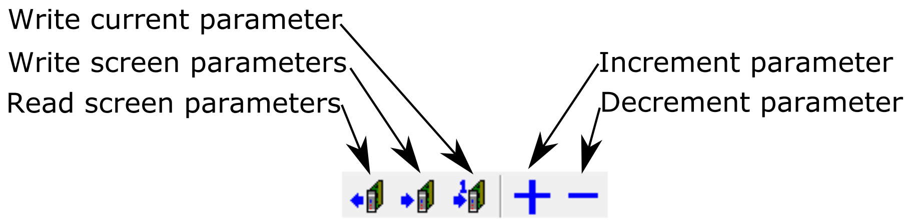


Figure 3.35: Bodac menu bar description

1. Go to "Tools/Utilities/Send firmware". Update the firmware to the latest version used by NOV. Use version L04-B23-2.95-1.

2. Go to "Tools/Analog I/O". Configure inputs and outputs according to Table 3.17.

Table 3.17: Analog I/O configuration

	Type	Range	Min	Max	Min fault	Max fault	Max unit
Angle CMD	±10V	±100%	-10.0V	10.0V	-11.0V	11.0V	100%
Pressure CMD	0-10V	0-100%	0.0V	10.0V	-1.0V	11.0V	400 bar
Power CMD	0-10V	0-100%	0.0V	10.0V	-1.0V	11.0V	100%
Act. pressure B	0-10V	0-100%	0.0V	10.0V	-1.0V	11.0V	400 bar
Act. pressure A	0-10V	0-100%	0.0V	10.0V	-1.0V	11.0V	400 bar
Res. angle CMD	±10V	±100%	-10.0V	10.0V	-11.0V	11.0V	100%
MCP-40/4742	0-10V	0-100%	0.0V	10.0V	-1.0V	11.0V	100%

3. Go to "Application Screens/Commands". Set Angle ramp and Pressure ramp to 0.01s. Power command is set to 100%.

4. Go to "Application Screens/Configure 1". Set up the pump card configuration as in Table 3.18.

Table 3.18: Configure 1 settings

Bodac menu	Setting	Comment
	Configure pump	
Leak oil correction	Off	
Min angle feedback	Off	
Pump size	A4..HS4 250/355/500	Chose right pump size
Rotation direction	A4..HS4..Right	
Valve	Output ±10V	
Angle sensor	0∼10V	
	Configure digital outputs	
DO5 Angle CMD	10% & 0.1s	Activates DO5 within limit
DO6 Pressure CMD	10 bar & 0.1s	Activates DO6 within limit
DO7 Rectangle	Off	Used for watchdog timer
	Adjustment zero point	
Zero point valve	Adjustment of proportional valve	Deviations between set point and process value is shown right in the window
Zero point pump	Adjustment of swivel angle	Deviations between set point and process value is shown right in the window
	Adjustment gain	
Gain 0∼10V	Set gain	Adjust gain corresponding to a range from -100% to 100%
	Configure digital input	
Enable set ok	On	Loss of Enable will generate card failure

5. Go to "Application Screens/Pressure controller". Four different settings for the pressure controller can be set by setting the binary digital inputs to the pump card. Default setting is Binary 0 and other binary settings are not in use.
   For winch applications: Set P to 0.89 and I to 200ms.
   For main control valve applications: Set P to 0.89 and I to 300ms.

6. Go to "Application Screens/Faults". Faults can be configured to activate fail safe mode for the pump card if set to STOP. For the HIL application all faults should be set to OFF.

7. Go to "Application Screens/Test Jack". Chose which parameters can be measured from X1 and X2 test jacks on the pump card. The signal configured to X1 is made available at terminals b26/b28. Set test jack output X1 to the Valve CMD parameter.

8. Go to "Tools/Preferences/Security/Login". In order to access all application screen one must login to security level 2 or 3. Go to "Tools/Preferences/Security/Login". Password is "2" for logging on to level 2.

9. Go to "Application Screens/Configure 3". Table 3.19 shows the settings and parameters used in the "Configure 3" window. This configuration assigns a new I-correction parameter to the PI controller if the pressure is within the pressure window.

Table 3.19: Configure 3 settings

Bodac menu	Setting	Comment
I Pressure Window	20 bar	NOVs standard value
Swivel angle fault	STOP/50%/0.5s	
Fast-I-correction (+) on after	70ms	
Fast-I-correction (-) in pressure control	ON	
Pressure limit	-100%	
I-correction (-) off for	0ms	This function is not in use
Velocity fast-I-correction	1000ms	
Pressure control with pressure sensor over shuttle valve	OFF	Shuttle valve not installed

10. Go to "Application Screens/Steuer". From this menu the proportional valve can be controlled in manual mode. Manual mode is activated by choosing "Controller inactive". Option is used for valve and swivel angle adjustment.

11. Start simulated pump by setting "Enable" to high in LabVIEW.

12. Go back to "Application Screens/Configure 1". Set controller in manual mode and adjust "Zero point valve" so that "Valve CMD" and "Actual Valve" are as close as possible. Set "Controller active" from "Steuer" menu.

13. Adjust the swivel angle by changing the value of "Zero point pump". Note for which value the pressure starts building up for A and B side. Set "Zero point pump" to the value in the middle of these values.

14. Set "Controller inactive" and go from -100% to 100%. Adjust the gain such that "Actual angle" is as close to -100% and 100%, respectively. Set "Controller active" from "Steuer" menu.

15. Go to "Controller/Set Parameters To Memory". This will burn all settings to the pump cards EPROM.

3.4.3 HIL System Performance

The performance in terms of cycle time, CPU load and memory usage is an important measure when verifying the HIL systems performance. National Instruments have a software called Distributed System Manager where the CPU load, memory usage, faults, network status and other features can be monitored. Loop time operating cycles directly affect the CPU load. Lowering the loop time increases the CPU load, and visa versa.

The simulation loops are executed in a variety of cycle times based on critical timed loops and non-critical timed loops. The FPGA target executes continuously with a cycle frequency up to 40 MHz. On the cRIO controller there are three loops. One loop for input and output processing to and from the FGPA which executes at a rate of 5 ms. A Control & Simulation loop also executes at 5 ms. The last loop, the communication to and from the host PC, executes at a rate of 50 ms. No time critical operations are performed on the host PC – only monitoring and sending of non-critical command signals to the cRIO.

VI and loop priorities are explained in section 2.3. The VIs and loops are given different priorities based on the determinism. Table 3.20 shows the priorities assigned to VIs and loops. The VIs included in the project is shown in Appendix C and the loops are described in section 3.3.3.

Table 3.20: Priorities for VIs and loops

VI/loop	Type	Deployed on	Priority
FPGA MAIN	VI	FPGA target	Time-Critical Priority
LVDT scale	VI	FPGA target	Time-Critical Priority
cRIO MAIN	VI	cRIO controller	Normal Priority
Input scale	VI	cRIO controller	Normal Priority
Output Scale	VI	cRIO controller	Normal Priority
HOST MAIN	VI	Host computer	Normal Priority
FPGA IO processing	While loop	FPGA target	Normal Priority
Input/output processing	Timed loop	cRIO controller	110*
Simulation loop	Control & Simulation loop	cRIO controller	110*
Host PC communication	While loop	cRIO controller	Normal Priority
RT target communication	While loop	Host PC	Normal Priority

* Between high- and time critical priority

Three loops are considered time-critical. These are;

- FPGA IO processing - while loop
- Input/output processing - timed loop
- Simulation loop - control and simulation loop

The while loop deployed on the FPGA executes continuously at a rate up to 40 MHz, hence the system performance will be limited by the timed loop or the control and simulation loop deployed on the cRIO. In Figure 3.36 the distributed CPU load and the CPU load trends are shown with a step time at 5 ms for both the timed loop and the control and simulation loop. The CPU load is distributed to two main priorities, normal and timed structures, with ∼20% CPU load for each. The timed loop and the control and simulation loop are considered timed structures.

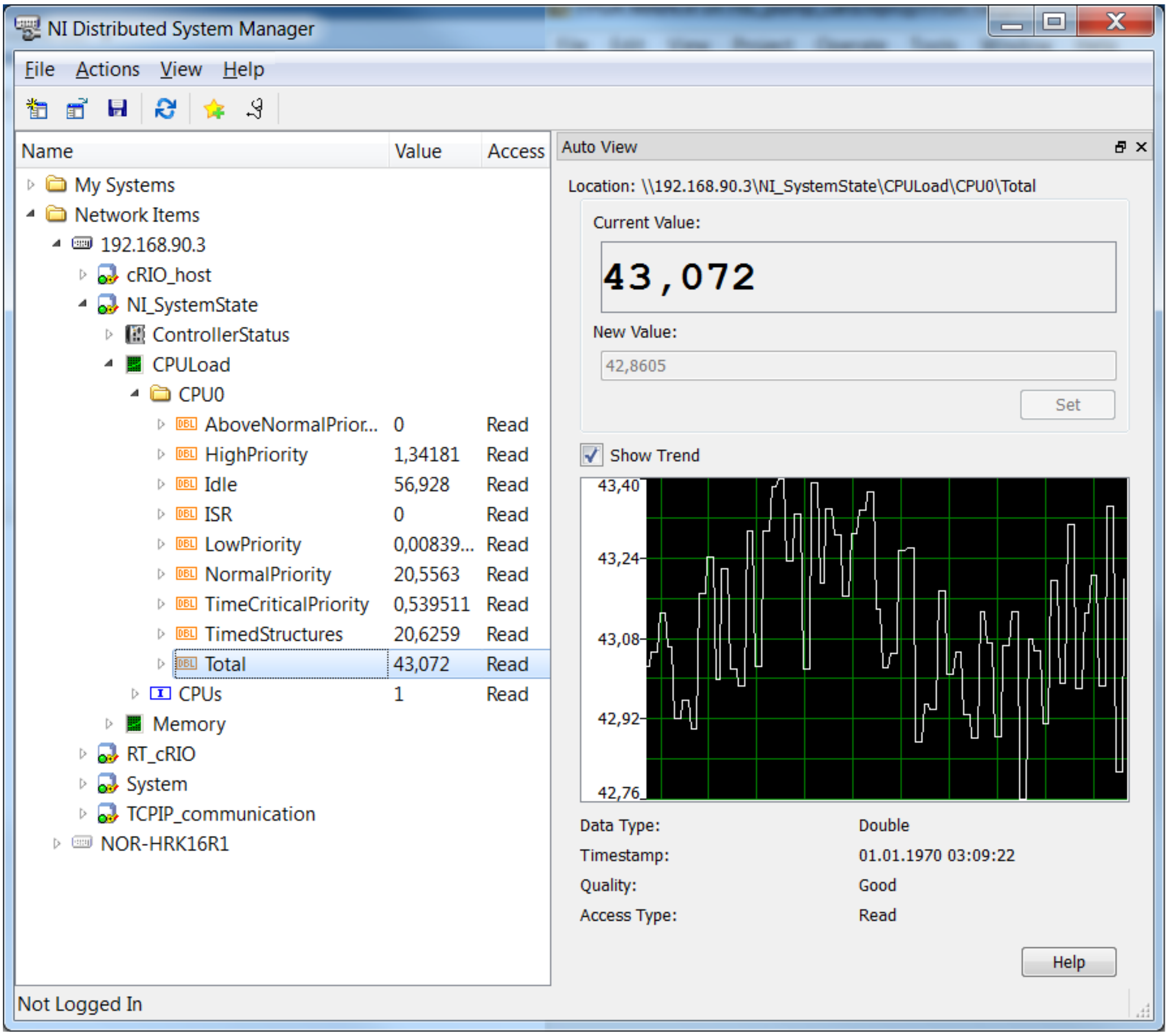


Figure 3.36: CPU load with 5ms step time visualized in Distributed System Manager

An execution rate at 5 ms yields for a CPU load ∼43%, with only ∼20% CPU load caused by deterministic tasks. Hence, the step time can be lowered to execute at a rate closer to the systems limit. Figure 3.37 shows the CPU load at 2 ms execution rate for the timed loop and the control and simulation loop.

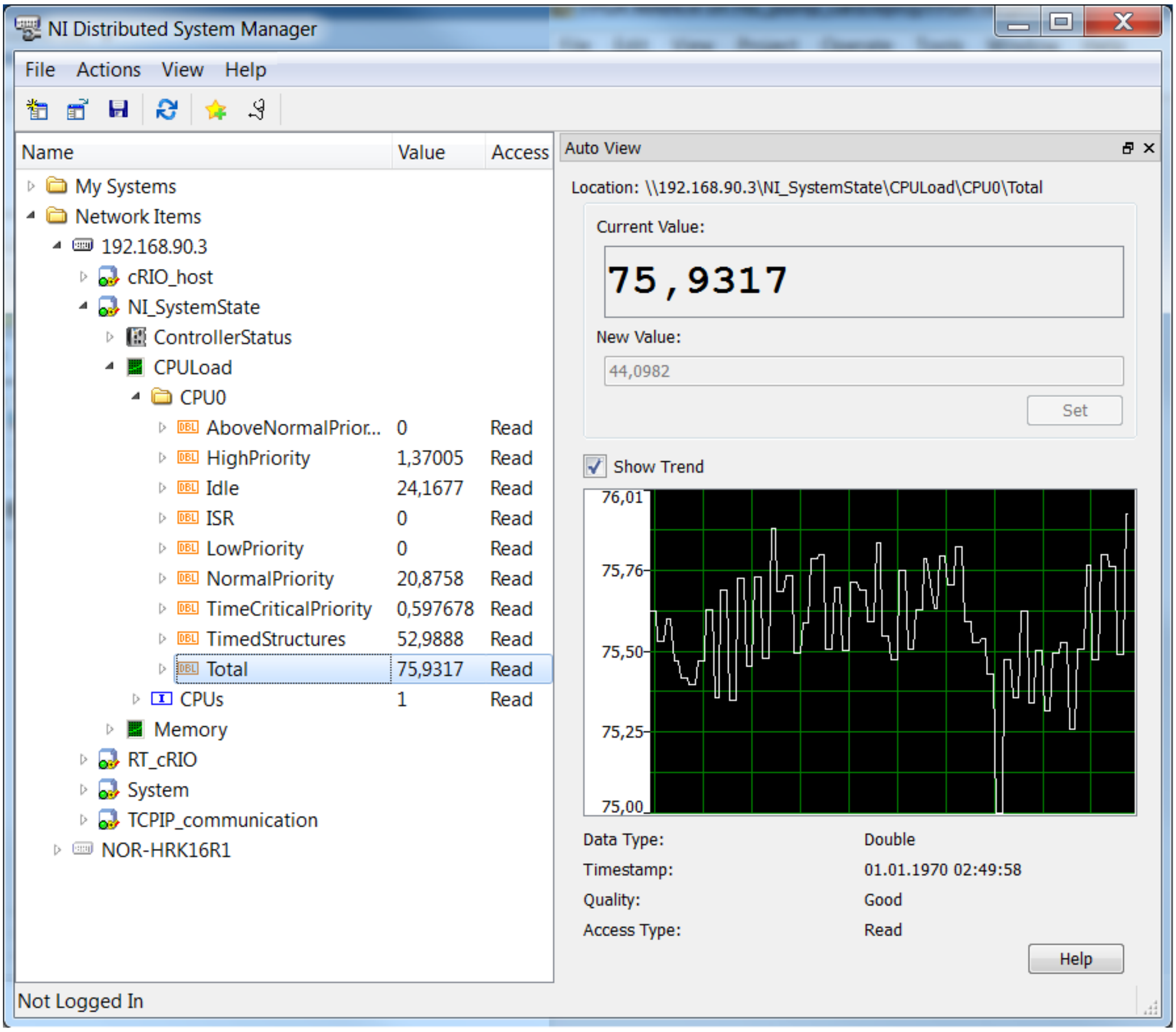


Figure 3.37: CPU load with 2 ms step time visualized in Distributed System Manager

Figure 3.37 shows an increase in the total CPU load to ∼75% and an increase in CPU load to ∼53% for deterministic tasks executed in the timed structures. A step time at 1 ms caused the CPU load to reach 100%. At a CPU load at 100% the deterministic task will not be executed within an acceptable time which caused unacceptable jitter. Hence, the systems performance is limited to an execution rate at 2 ms.

External RT Target

By setting up the HIL system with an external RT target the computational power for the HIL system increases. The desktop computer used in as an external RT target has the configuration shown in Table 3.21.

Table 3.21: Desktop computer hardware configuration

Component	Specifications
CPU	Intel Core i5, 3.33 GHz, Quad Core
Memory	8 GB RAM
Ethernet	National Instruments GigE Vision Adapter

Compared to the cRIO a HIL system with an external RT target can execute simulation models at a higher rate. Figure 3.38 shows the CPU load for each core for an execution rate at 1 ms. Executing the same dynamic simulation model at a higher rate on the external RT target yields for only an average of 2% CPU load for each core. Seen in comparison to Figure 3.37 the computational power is increased many times. Distrubuted System Manager shows a CPU load at ~78% for the cRIO when used as a I/O processing unit executing at a rate of 1 ms.

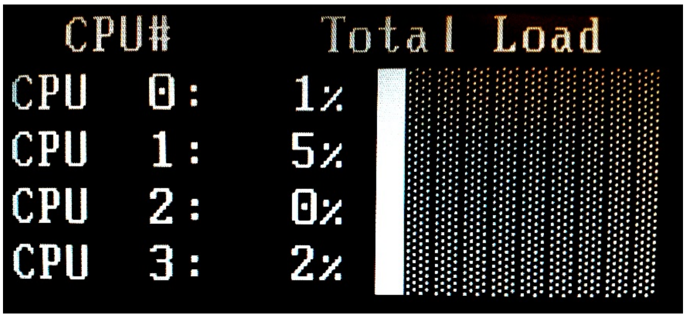


Figure 3.38: CPU load for each core on external RT target

Chapter 4

Discussions

4.1 Simulation Model

Estimated transfer functions for proportional valve and axial piston cylinder are made based on bode plots and step responses from data sheets. The bode plots and step responses are assumed to be simplified in terms of the components dynamic characteristics. Hence, the estimated transfer functions are compared to simplified responses due to a lack of log data from actual response. This again leads to possible deviations between actual response from the hydraulic components and the estimated transfer function models.

The estimated models for proportional valve and axial piston cylinder will be sufficient for pump tuning purposes used together with the modelled leakage circuit. If special fault scenarios, or similar, is to be simulated the dynamics of the hydraulic control system is important. Hence, the estimated models will be insufficient due to the simplified dynamic characteristics. For special test scenarios the transfer functions should be based on estimations from parameter identification or frequency response from an experimental setup. This is further discussed in section 4.3.

The dynamic mathematical model set up for the leakage circuit is modelled with some ideal components. The pressure source and the shuttle valve (see Figure 3.27) are considered ideal, causing the pressure to always stay equal to or above 25 bar. In a physical system this would not be the case due to system dynamics.

The leakage circuit is not set up in order to examine this hydraulic circuit. The purpose is to create model feedback from the hydraulic circuit where simple modifications to the circuit can be done.

4.2 System Performance

System performance has been considered for both HIL systems; with and without external RT target. The HIL systems have various application areas where one HIL system is more applicable than the other and visa versa. The HIL system with the cRIO as RT target features a 400 MHz CPU which, in many cases, comes up short. The system is applicable for simple mathematical simulation models like the leakage circuit used for the pump tuning process.

Real-time simulation of more complex models requires more computational power. This can be solved by using an external RT target. The setup for the external RT target is described in section 3.3.8. The system performance for this HIL setup depends on the computational power for the PC used as an external RT target. Normally, this system can execute more complex models at a higher execution rate than the cRIO.

The cRIO used throughout the project period is property of University of Agder. Hence, NOV must purchase a new cRIO in order to make use of the HIL system. Considerations related to system performance are important when purchasing a new cRIO. Following questions has to be answered:

- Should the cRIO be .dll file compatible?
- Should the cRIO be able to execute complex exported simulation models?

A .dll file compatible cRIO with the computational power to execute exported simulation models yields for one of the cRIOs from the performance series. This setup remove the need for an external RT target, but increases cost due to the high price on cRIOs in the performance series.

NI cRIO-9075, or similar cRIO from National Instruments value series, will need an external RT target to execute exported simulation models due to limited computational power. An advantage with an external RT target is that the computational power is significantly higher than cRIOs from National Instruments performance series.

4.3 Future Work

In order to expand the possible application areas for the HIL system, there are several measures that can be taken. Future work to be done includes:

- Determining the valve command signal from PWM signals
- Develop new simulation models for proportional valve and axial piston cylinder
- Crane simulator model exported from SimulationX
- Implementing HIL system with DMA cards for hydraulic valve and motor control

Valve Command from PWM Signals
As described in section 3.3.4 the valve command signal from the pump card loses the zeropoint valve adjustment possibility when the valve output is chosen to $\pm 10\,\mathrm{V}$. In order to determine the valve command with the possibility for zeropoint adjustment the PWM signal to the coils has to be identified in terms of period time and on-time.

Simulation Models for Proportional Valve and Axial Piston Cylinder
Simulation models for the proportional valve and axial piston cylinder are based on what is assumed to be simplified step responses. Future work to be done in this area is to perform a frequency analysis on the hydraulic control circuit in order to determine the systems actual dynamic response. The frequency response data can be used to develop new transfer function models, or similar, based on actual response rather than simplified step- and frequency response from data sheets.

Crane Simulator Model
NOV have a crane model which is modelled in SimulationX. The simulation model, or parts of the simulation model, can be exported for execution on the external RT target. The crane model is modelled with custom made hydraulic components, based on experimental data and physical dimensions. Hence, by running a HIL simulation with this setup, fault scenarios can be provoked in order to investigate these.

DMA Cards
NOV uses DMA cards from Hydraulic Control Systems (HCS) for both open-loop and closed-loop control of hydraulic valves and motors. DMA cards can be implemented to the HIL system for use with and without the pump controller card. HIL simulation with DMA cards and pump controller cards can help with rapid testing and developing of new crane applications with high focus on optimized control of the hydraulic system.

NOVs crane hydraulic department in Molde also sees the HIL rig as a addition to their development. The department has given the comment below based on the HIL rig with both HCS DMA cards and Bosch Rexroth VT-VPCD pump cards:

> —We envision that such HIL rig will primarily be useful for us when implementing our crane model for simulation. This in order to provoke and understand why errors occur and see how the system behaves in extreme situations. In addition, we envision that such HIL rig could be used for education.
>
> — Gunnar Thompson-Kvernland, Crane Hydraulics Molde Norway

Chapter 5

Conclusions

A test rig with components for HIL simulation is built. A cRIO-9075 is installed together with a pump controller card from Bosch Rexroth. This setup makes it possible to run real-time simulations with the pump card as a hardware-in-the-loop. Mathematical models for the hydraulic HS4 control system are derived by analysing bode plots and step responses. The proportional valve and axial piston cylinder is expressed by 3rd and 2nd order transfer functions, respectively.

A dynamic simulation models are set up in LabVIEW. The dynamic simulation model simulates a hydraulic leakage circuit. By adjusting hydraulic model parameters the pump cards behaviour, subjected to various load cases, can be examined. The leakage circuit is applicable for pump tuning purposes. In Bodac the pump card can be configured and optimized for a pump control. The test rig is configured as a stand-alone application, which results in a HIL system that executes when the HIL rig is powered up and an application file is ran on a host computer. The HIL system can be operated without using LabVIEW. The dynamic simulation model can be configured from a GUI. The GUI is built for instructors and operators of the HIL system. Tuning of the pump controller card is done through the application windows in Bodac.

The cRIO-9075 comes up short when the simulation models become more complex. The HIL system is set up with an external RT target in order to increase the computational power for simulation purposes. The system performance in terms of CPU load increased over 70 times with an external RT target. Hydraulic models modelled in SimulationX can be exported to .dll files and executed on the external RT target in order to examine known fault scenarios related to the pump controller card.

Exported simulation models from SimulationX have not been implemented for experimental tasks, only for verification of operation. A LabVIEW project with an external RT target is created to enable an implementation of exported models from SimulationX.

Bibliography

[1] S. Röck, "Hardware in the loop simulation of production system dynamics," *German Academic Society for Production Engineering*, vol. 5, p. 8, 2011.

[2] C. Dufour, C. Andrade, and J. Bélanger, "Real-Time Simulation Technologies in Education: a Link to Modern Engineering Methods and Practices," International Conference on Engineering and Technology Education, March 2010.

[3] K. Popovici and P. J. Mosterman, *Real-Time Simualtion Technologies: Principles, Methodologies and Applications.* Computational Analysis, Synthesis, and Design of Dynamic Systems, CRC Press, 2012.

[4] XILINX, "7 Series FPGAs Overview." DS180 (v1.16.1), December 2014.

[5] D. R. King, *Mobile Robotics Experiments with DaNI.* Colorado School of Mines, 2011.

[6] National Instruments, "What Is NI CompactRIO?." Available: http://www.ni.com/compactrio/whatis/ (Accsessd: 27.02.2015), February 2015.

[7] National Instruments, "NI LabVIEW for CompactRIO Developer's Guide." Available: www.ni.com/pdf/products/us/fullcriodevguide.pdf (Accessed 04.03.2015), 2014.

[8] National Instruments, "Creating Deterministic Applications (RT Module)." Available: http://zone.ni.com/reference/en-XX/help/370622H-01/lvrtconcepts/builddeterapps_rt/ (Accessed: 21.04.2015), June 2010.

[9] National Instruments, "Understand the Priority-Based Scheduling Model." Available: http://zone.ni.com/reference/en-XX/help/370622H-01/lvrtbestpractices/rtpriorities/ (Accessed 16.04.2015), June 2010.

[10] National Instruments, "Using Local and Global Variables Carefully." Available: http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/using_local_and_global/ (Accessed: 28.04.2015), June 2012.

[11] Bosch Rexroth, "Digital control eletronics for A4VS... axial piston units with HS4 control and A2V... with EO4 control." RE30028/03.09, March 2009.

[12] National Instruments, "Measuring Position and Displacement with LVDTs." Available: http://www.ni.com/white-paper/3638/en/ (Accessed: 05.02.2015), September 2006.

[13] Macro Sensors, "LVDT Basics." Available: http://www.macrosensors.com/downloads/misc/LVDT-Basics.pdf (Accsessd: 27.02.2015), April 2014.

[14] National Instruments, "LVDT fundamentals." Available: http://www.ni.com/white-paper/4498/en/ (Accessed: 06.02.2015), September 2009.

[15] M. R. Hansen and T. O. Andersen, "Hydraulic Componets and Systems." University of Agder and Aalborg University, August 2012.

[16] Bosch Rexroth, "4/2 and 4/3 proportional directional 1/22 valves, direct operated, with electrical position feedback, without/with integrated electronics (OBE)." RE29061/11.12, November 2012.

[17] National Instruments by elgeeko, "The Definitive Guide: Programming NI VxWorks Real-Time Controllers in C/C++." Available : https://decibel.ni.com/content/docs/DOC-13537.pdf (Accessed: 01.05.2015), September 2012. Version 19.

[18] National Intruments by elgeeko, “The Definitive Guide: Programming NI VxWorks Real-Time Controllers in C/C++ (gcc Method).” Available: https://decibel.ni.com/content/docs/DOC-14237.pdf (Accessed 01.05.2015), December 2010. Version 5.

[19] National Instruments, “Converting a Desktop PC to a LabVIEW Real-Time Target.” Available: http://www.ni.com/tutorial/2733/en/ (Accessed 04.05.2015), September 2012.

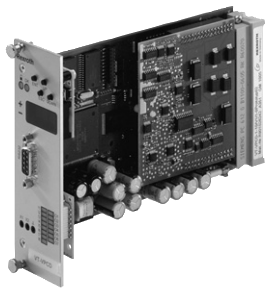
Appendix A

Digital Control Electronics for the Axial Piston Pumps A4VS... with HS4 Control

Digital control electronics for the axial piston pumps A4VS... with HS4 control and A2V... with EO4 control

Type VT-VPCD

RE 30028
Edition: 2013-07
Replaces: 2012-06



H7190

- Component series 1X
- The HS4 or EO4 control is used for the electro-hydraulic swivel angle and pressure control as well as for the power limitation of axial piston variable displacement pumps.

Features

The control system with HS4 control consists of the following assemblies:

- A4VS...HS4 axial piston pump with attached 4WRE6-2X/822 proportional valve including position transducer for swivel angle and valve position sensing
- Recommended pressure transducer HM20-2X for recording the system pressure
- VT-VPCD control electronics to implement all electrical functions necessary for HS4 control

The control system with EO4 control consists of the following assemblies:

- A2V...EO4 axial piston pump (housing and/or installation pump) with attached proportional valve including position transducers for swivel angle and valve position sensing
- Recommended pressure transducer HM20-2X for recording the system pressure
- VT-VPCD control electronics to implement all electrical functions necessary for the EO4 control

Parameterization is effected via a serial interface. The user-specific data can be exactly reproduced and is protected against unintended or unauthorized adjustment.

Continued on page 2

Contents

Features (continued)

- Digital inputs for calling pre-set parameters [1)]
- Ramp times for swivel angle and pressure command values [1)]
- Analog inputs for command and actual values [1)]
- Enable input and collective fault output [1)]
- Oscillators/demodulators for two inductive measuring systems
- Clocked, flow-controlled output stage
- Switching power supply unit for the internal supply voltages
- Function and status display LEDs
- 2 measuring sockets configurable via display and/ or Bodac
- Serial interface RS 232

[1)] Please note the respective bus documentation.

- Up to 32 control electronics can be interconnected for parameterization and diagnosis via the local bus
- Size selection (size 40 to 1000 for A4VS...HS4, size 500 to 1000 for A2V...EO4) and parameterization via BODAC
- Parameterization for pump A4VHO 450 HS4
- Valve position controller
- Pressure controller with subordinate swivel angle controller
- Parameterizable power limitation
- Leakage compensation
- Master/slave capability
- Mooring capability
- Oscilloscope function
- Parameterizable test output
- Diagnosis display

Ordering code

01				02		03		04		05		06
VT–VPCD	**–**	**1**	**–**	**1X**	**/**		**/**	**1**	**–**		**–**	**1**

01	Digital control electronics for controlling axial piston variable displacement pumps	**VT–VPCD**
02	Component series 10 to 19 (10 to 19: Unchanged installation and connection dimensions)	**1X**
03	For axial piston pump A4VS...HS4 with swivel angle sensor AWX F004 D01 and for axial piston pump A2V...EO4 (housing pump) with swivel angle sensor MCP-40/4742	**V0**
	For axial piston pump A2V...EO4 (installation pump) with swivel angle sensor DK 100 (only available without bus connection)	**V100**
04	With display	**1**
05	Without bus connection	**0**
	PROFIBUS DPV0	**P**
	DeviceNet	**D**
	CANopen	**C**
06	With valve output stage	**1**

Preferred types	Material number
VT-VPCD-1-1X/V0/1-0-1	R901044346
VT-VPCD-1-1X/V0/1-P-1	R901089559

PC system requirements:
- Windows XP, Windows Vista, Windows 7
- RAM (recommendation: 256 MB)
- 250 MB of available hard disk capacity

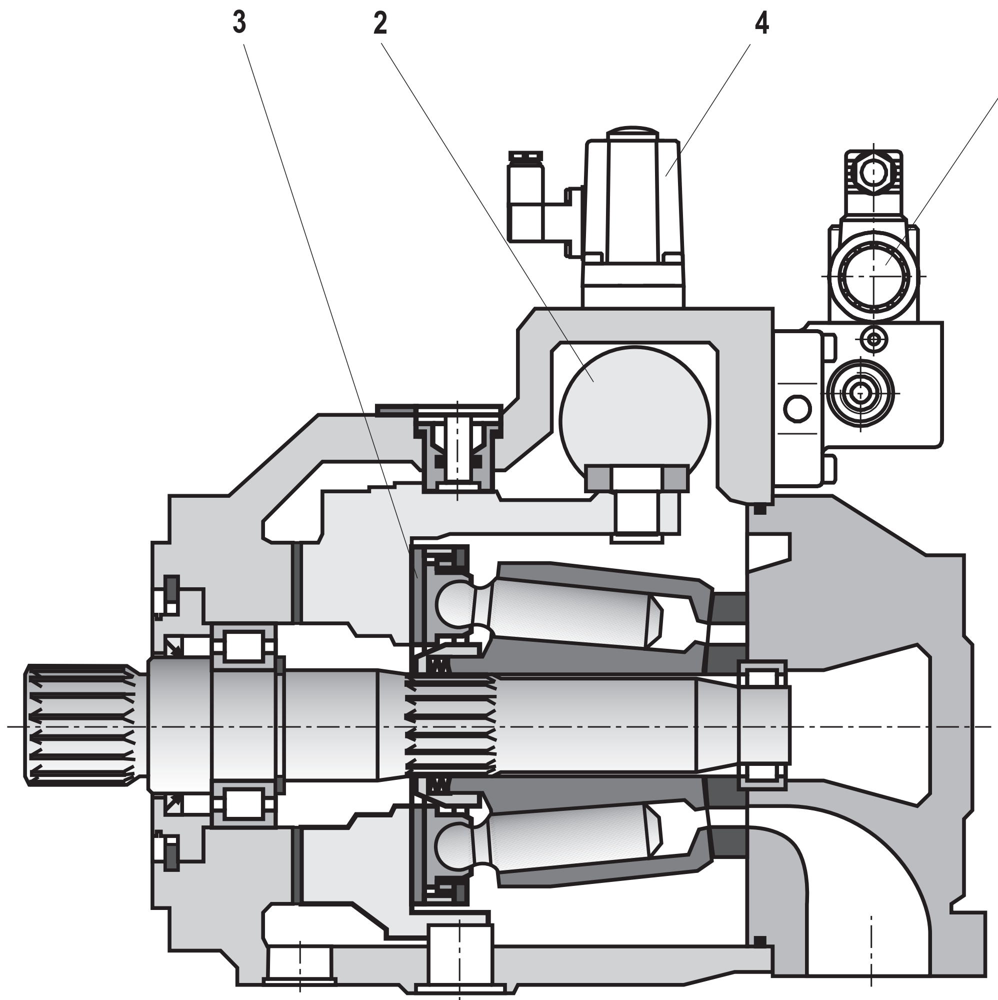
Required accessories:
- PC program BODAC: CD ordering information: SYS-HACD-BODAC-01 (R900777335) or free download on the Internet at www.boschrexroth.com/vpcd
- Interface cable: Cable set VT-HACD-1X/03.0/HACD-PC (R900776897) or standard 1:1 cable

Suitable card holder:
- 19 inch racks VT 19101, VT 19102, VT 19103 and VT 19110 (see data sheet 29768)
- Open card holder VT 3002-2X/64G (see data sheet 29928), mat. no. R900991843 (only for control cabinet installation!)
- Connection adapter VT 10812-2X/64G (see data sheet 30105), mat. no. R900713826

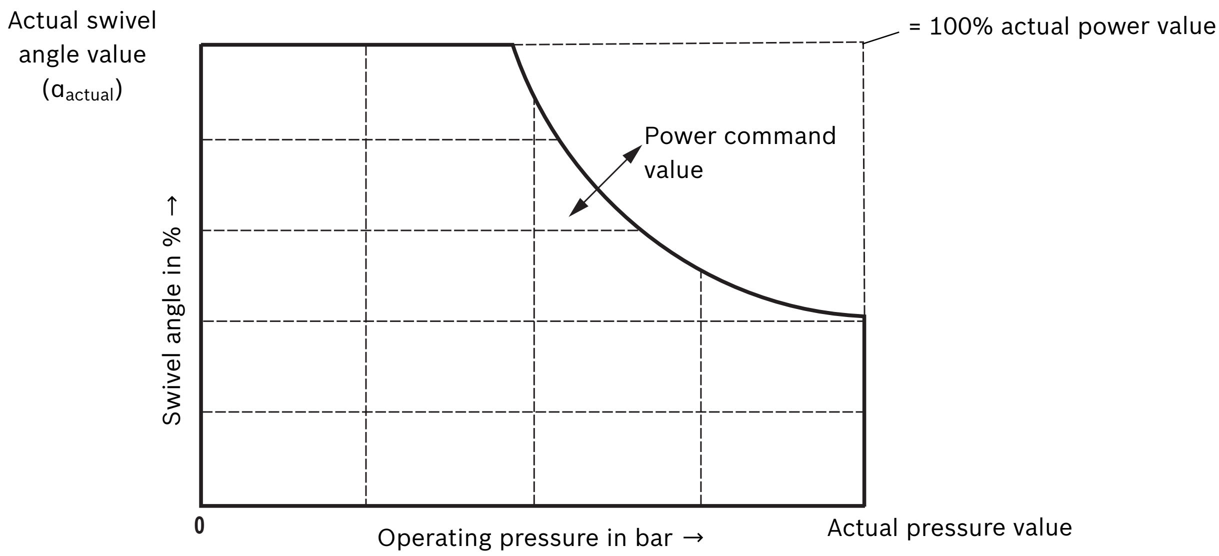
Functional description using the A4VS axial piston pump with HS4 control as an example

The swivel angle and pressure control as well as the power limitation of the A4VS... variable displacement pump are effected by an electrically controlled proportional valve (1). Via the actuating piston (2) of the pump, this valve determines the position of the swash plate (3).
If the pump does not rotate, in case of depressurized high-pressure and actuating system and if enable is not operated, the swash plate is held in the "Zero" swivel angle position by the spring centering.
The position of the swash plate is determined by an inductive position transducer (4), the actual pressure value is recorded by a pressure transducer. Both actual values are supplied to the VT-VPCD control electronics and linked with each other by the software.
The actual power value is calculated from the product of actual pressure value and actual swivel angle value. The controller software ensures by means of a minimum value generator that the controller corresponding to the working point is always active.

In the static condition (i.e. swivel angle command value equals actual swivel angle value, power command value equals actual power value or pressure command value equals actual pressure value) the valve control spool is in central position.
If the superior controllers demand e.g. an increase in the swivel angle (corresponds to an increase in the flow), the valve control spool must be deflected out of the central position until the swivel angle has reached the required value.
The sectional drawing shows the A4VS... variable displacement pump with HS4 control; the proportional valve (1) is controlled using the VT-VPCD control electronics.
Notice for the HS4 control:
With de-energized proportional valve and pump with clockwise rotation and if the actuating pressure is available, the pump swivels to swivel angle α = 0 (A4VSO design) or α = –100% (A4VSG design).

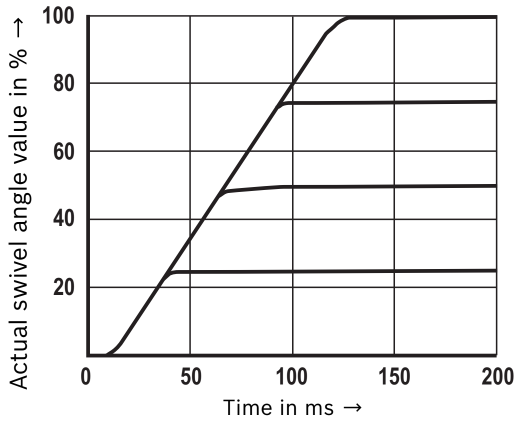


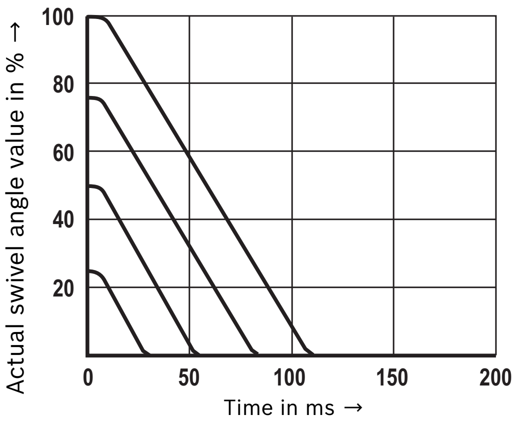
Static characteristic curve



Transition function with swivel angle command value step

Example A4VS with HS4 control, size 250, actuating pressure *p* = 125 bar





Functional description of the control electronics

The control electronics is set-up as printed circuit board in Europe format 100 x 160 mm, fitted on both sides. It comprises a switching power supply unit [1] creating all internally required voltages.

The central unit is a microcontroller controlling the entire process and realizing the controller functions. Data for configuration, command values and parameters are stored in a FLASH in a non-volatile form.
Four binarily coded digital inputs are used to call up parameter sets (command values) from the memory in which you can store a maximum of 16 sets. A call-up activates a command value for the swivel angle, the pressure and the power limitation as well as ramp times for swivel angle and pressure.

More control inputs have the following functions:

"Command value valid":	Release of the parameter set addressed by the current call-up (H active)
"Enable":	Activation of the control (H active)
Comment:	H active = High active (level 16 V to U_B) L active = Low active (level 0 V to 5 V) L/H edge = Low High edge

Via the differential inputs AI7, AI5 and AI4 [3], the analog command values for the swivel angle, the pressure and the maximum power are specified. With a positive swivel angle command value, the pump swivels in "counterclockwise" swivel direction (= flow direction P → B). The digital call-up command values are added to the analog command values; the total of both command values is supplied to the controller input via the relevant ramp generators.

The controller output signal controls the output stage [6] depending on the command/actual value differences.

The position of the valve control spool [11], the swivel angle of the variable displacement pump [12a, 12b or 12c] and the system pressure [13] are measured and supplied to the control loop via evaluation electronics [7].

For the pressure control, different modes are provided:

Depending on the configuration, the pressure controller works with one or two pressure sensors.

- Open circuit:
  1 sensor, optionally current or voltage
- Closed circuit:
  2 sensors, optionally current or voltage

In the closed circuit, both pressure sensors are evaluated. As soon as the control electronics is in pressure control, the larger of the two pressures determines the control behavior. To compensate control deviations (pressure command value-actual pressure value), the pressure controller can also swivel the pump to the opposite side as well as beyond its specified swivel angle command value.

The switching outputs are configured via BODAC.
The following functions may be selected:

- Swivel angle control active DO1
- Pressure control active DO2
- Power limitation active DO3
- Slave mode active DO4
- Swivel angle in the accuracy window DO5
- Pressure in the accuracy window DO6
- Rectangular 32 Hz DO7

The test output (b26 or measuring socket X1) is also configured via BODAC. It is used for the analog output of internal variables.

Enable and error messages

Setting the enable input activates the control. If no command value call-up is activated, parameter set 0 is set.

Error logics identify the following faults:

- Cable brake or short circuit in the actual valve value recording
- Cable brake or short circuit in the actual swivel angle value recording
- Cable break at the pressure transducer (only current interface)
- Closed-loop errors (i.e. control deviations between swivel angle command value and actual swivel angle value)

An error is displayed at output d22. The "OK" message goes out, signal level is 0 V.

Errors are also shown at the display.

Parameterization and diagnosis

Using the serial interface [2], the pump size is selected and the leakage oil correction and the sequence control are activated or deactivated and switching outputs and the test output are configured via BODAC at the front-side D-Sub socket. Via the local bus, up to 32 control electronics can be connected. Via BODAC, every control electronics is assigned a bus address. Reconnection of the serial interface cable is not required. More information in document 30028-01-B.

Display elements and measuring sockets

The freely configurable measuring sockets X1/X2 located at the front plate serve to display the process signals. Configuration see online help.

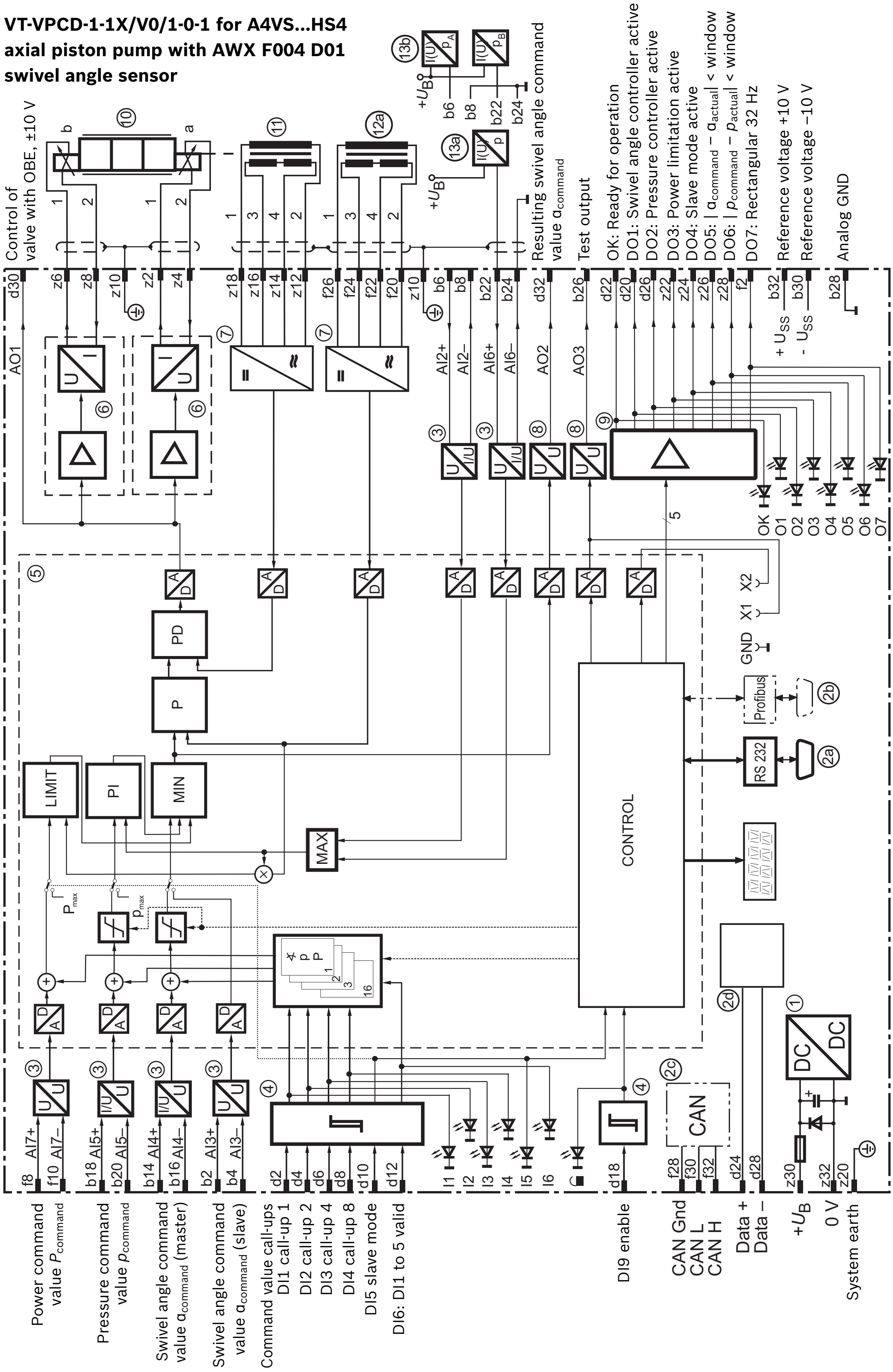
LEDs display the following states:

LED "🔓" (green):	Enable active
LED "OK" (green):	OK ready for operation
LEDs "I1"..."I4" (yellow):	Binarily coded command value call-ups
LED "I5" (yellow)	Slave mode
LED "I6" (yellow)	Command value valid
LED "I7" (yellow)	Not assigned

[ ] = assignment to the block diagrams on pages 7, 9 and 11

Block diagram

VT-VPCD-1-1X/V0/1-0-1 for A4VS...HS4 axial piston pump with AWX F004 D01 swivel angle sensor



1 Power supply unit
2a Serial interface
2b PROFIBUS interface, optional
2c CAN interface, optional
2d Local bus
3 Voltage and/or current input
4 Level converter
5 Microcontroller
6 Flow-controlled output stage
7 Oscillator/demodulator
8 Voltage output
9 Output driver
10 Proportional valve
11 Position transducer of valve
12a Swivel angle sensor of pump
13a Pressure sensor current (open circuit)
13b Pressure sensors current (closed circuit)

Pin assignment of the male multipoint connector

VT-VPCD-1-1X/V0/1-0-1 for A4VS...HS4 axial piston pump with AWX F004 D01 swivel angle sensor

Row d

Pin	Short denomination	Description
2	DI1	Command value call-up 1, H active
4	DI2	Command value call-up 2, H active
6	DI3	Command value call-up 4, H active
8	DI4	Command value call-up 8, H active
10	DI5	Slave mode, H active
12	DI6	DI1 to DI5 valid, H active
14		n.c.
16		n.c.
18	DI9	Enable, H active
20	DO1	Swivel angle controller active, H active
22	OK	OK output, H active
24	Data +	Local bus
26	DO2	Pressure controller active, H active
28	Data –	Local bus
30	AO1	Control of valve with OBE, ±10 V
32	AO2	Resulting swivel angle command value for master/slave operation

Row b

Pin	Short denomination	Description
2	AI3+	Slave swivel angle command value (in case of slave operation)
4	AI3–	Slave swivel angle command value, reference
6	AI2+	Actual pressure value p_A, (I or U)
8	AI2–	Actual pressure value $p_{A,}$ reference
10		n.c.
12		n.c.
14	AI4+	Swivel angle command value
16	AI4–	Swivel angle command value, reference
18	AI5+	Pressure command value
20	AI5–	Pressure command value, reference
22	AI6+	Actual pressure value p_B, (I or U)
24	AI6–	Actual pressure value p_B, reference
26	AO3	Test output (measuring socket X1)
28	AGND	Analog GND
30	REF–	Reference voltage –10 V
32	REF+	Reference voltage +10 V

Row z

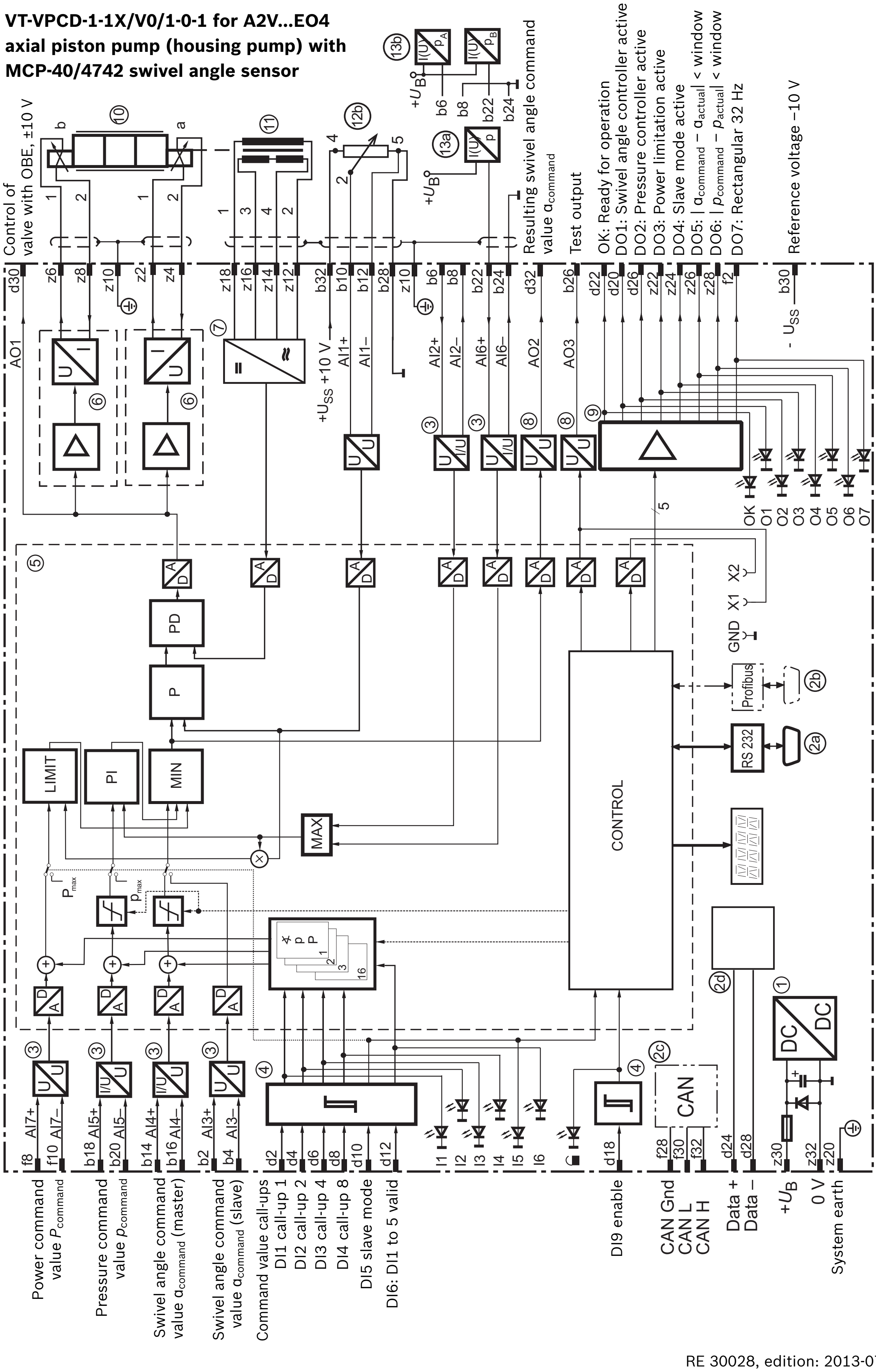
Pin	Short denomination	Description
2	MA+	Solenoid A +
4	MA–	Solenoid A –
6	MB+	Solenoid B +
8	MB–	Solenoid B –
10	Shield	Shield
12	L1O–	Position transducer of valve, feed –, pin 2
14	L1I–	Position transducer of valve, actual value –, pin 4
16	L1I+	Position transducer of valve, actual value +, pin 3
18	L1O+	Position transducer of valve, feed +, pin 1
20	System earth	System earth
22	DO3	Power limitation active, H active
24	DO4	Slave mode active, H active
26	DO5	$\lvert \alpha_{command} - \alpha_{actual} \rvert$ < window, H active
28	DO6	$\lvert p_{command} - p_{actual} \rvert$ < window, H active
30	UB	Supply voltage $+U_B$
32	LO	Supply voltage 0 V

Row f

Pin	Short denomination	Description
2	DO7	Rectangular 32 Hz
4		n.c.
6		n.c.
8	AI7+	Power command value
10	AI7–	Power command value, reference
12		n.c.
14		n.c.
16		n.c.
18		n.c.
20	L2O–	Swivel angle sensor of pump, feed –, pin 2
22	L2I–	Swivel angle sensor of pump, actual value –, pin 4
24	L2I+	Swivel angle sensor of pump, ctual value +, pin 3
26	L2O+	Swivel angle sensor of pump, feed +, pin 1
28	CAN Gnd	CAN bus reference
30	CAN L	CAN bus input/output
32	CAN H	CAN bus input/output

Block diagram

VT-VPCD-1-1X/V0/1-0-1 for A2V...EO4 axial piston pump (housing pump) with MCP-40/4742 swivel angle sensor



1 Power supply unit
2a Serial interface
2b PROFIBUS interface, optional
2c CAN interface, optional
2d Local bus
3 Voltage and/or current input
4 Level converter
5 Microcontroller
6 Flow-controlled output stage
7 Oscillator/demodulator
8 Voltage output
9 Output driver
10 Proportional valve
11 Position transducer of valve
12b Swivel angle sensor of pump
13a Pressure sensor current (open circuit)
13b Pressure sensors current (closed circuit)

Pin assignment of the male multipoint connector

VT-VPCD-1-1X/V0/1-0-1 for A2V...EO4 axial piston pump (housing pump) with MCP-40/4742 swivel angle sensor

Row d

Pin	Short denomination	Description
2	DI1	Command value call-up 1, H active
4	DI2	Command value call-up 2, H active
6	DI3	Command value call-up 4, H active
8	DI4	Command value call-up 8, H active
10	DI5	Slave mode, H active
12	DI6	DI1 to DI5 valid, H active
14		n.c.
16		n.c.
18	DI9	Enable, H active
20	DO1	Swivel angle controller active, H active
22	OK	OK output, H active
24	Data +	Local bus
26	DO2	Pressure controller active, H active
28	Data –	Local bus
30	AO1	Control of valve with OBE, ±10 V
32	AO2	Resulting swivel angle command value for master/slave operation

Row b

Pin	Short denomination	Description
2	AI3+	Slave swivel angle command value (in case of slave operation)
4	AI3–	Slave swivel angle command value, reference
6	AI2+	Actual pressure value p_A, (I or U)
8	AI2–	Actual pressure value $p_{A,}$ reference
10	AI1+	Swivel angle sensor of pump, pin 2
12	AI1–	Swivel angle sensor of pump, pin 5
14	AI4+	Swivel angle command value
16	AI4–	Swivel angle command value, reference
18	AI5+	Pressure command value
20	AI5–	Pressure command value, reference
22	AI6+	Actual pressure value p_B, (I or U)
24	AI6–	Actual pressure value p_B, reference
26	AO3	Test output (measuring socket X1)
28	AGND	Analog GND and swivel angle sensor of pump, pin 5
30	REF–	Reference voltage –10 V
32	REF+	Reference voltage +10 V and swivel angle sensor of pump, pin 4

Row z

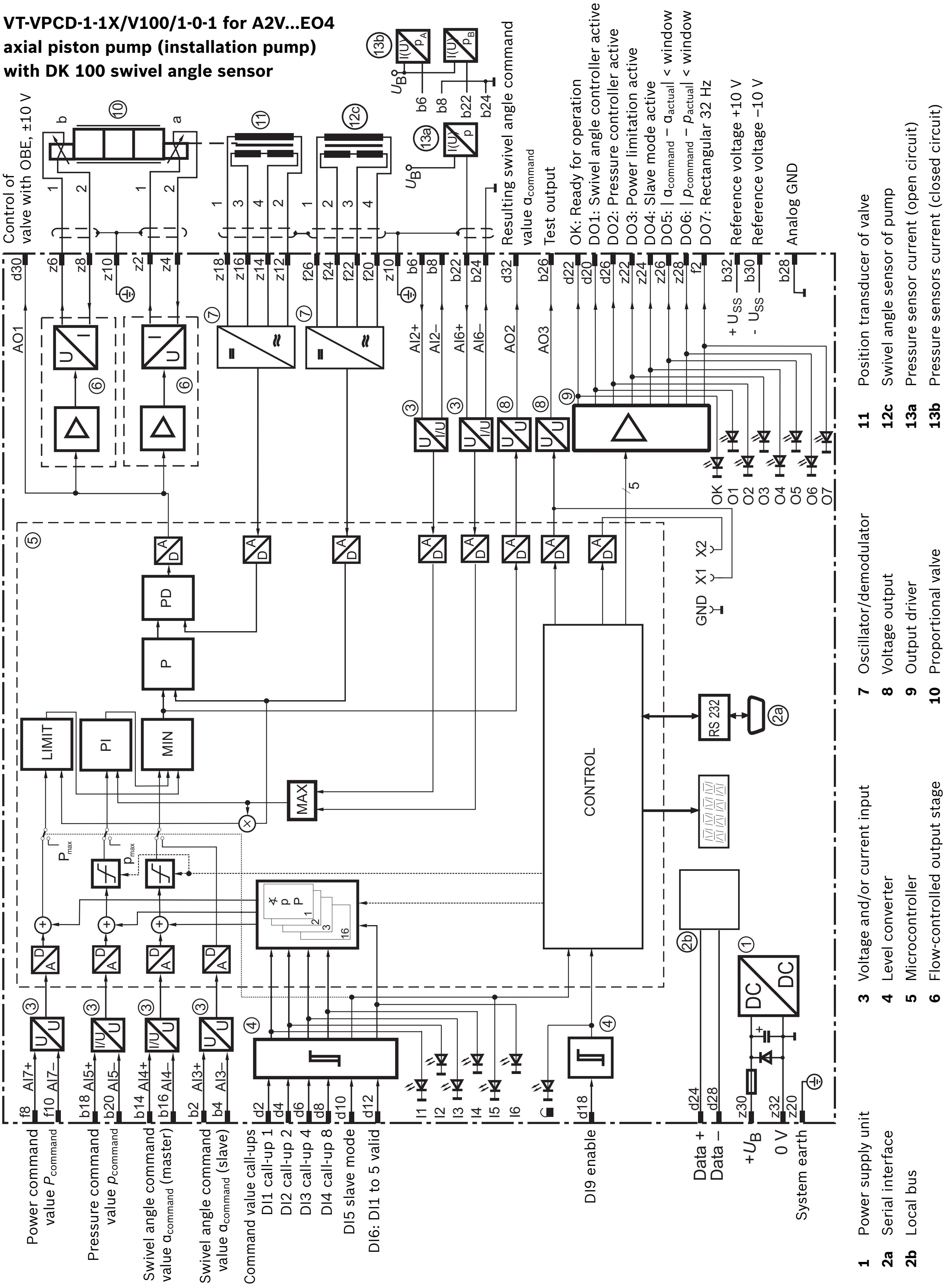
Pin	Short denomination	Description
2	MA+	Solenoid A +
4	MA–	Solenoid A –
6	MB+	Solenoid B +
8	MB–	Solenoid B –
10	Shield	Shield
12	L1O–	Position transducer of valve, feed –, pin 2
14	L1I–	Position transducer of valve, actual value –, pin 4
16	L1I+	Position transducer of valve, actual value +, pin 3
18	L1O+	Position transducer of valve, feed +, pin 1
20	System earth	System earth
22	DO3	Power limitation active, H active
24	DO4	Slave mode active, H active
26	DO5	$\mid \alpha_{command} - \alpha_{actual} \mid$ < window, H active
28	DO6	$\mid p_{command} - p_{actual} \mid$ < window, H active
30	UB	Supply voltage $+U_B$
32	LO	Supply voltage 0 V

Row f

Pin	Short denomination	Description
2	DO7	Rectangular 32 Hz
4		n.c.
6		n.c.
8	AI7+	Command value power
10	AI7–	Command value power, reference
12		n.c.
14		n.c.
16		n.c.
18		n.c.
20		n.c.
22		n.c.
24		n.c.
26		n.c.
28	CAN Gnd	CAN bus reference
30	CAN L	CAN bus input/output
32	CAN H	CAN bus input/output

Block diagram

VT-VPCD-1-1X/V100/1-0-1 for A2V...EO4 axial piston pump (installation pump) with DK 100 swivel angle sensor



Pin assignment of the male multipoint connector

VT-VPCD-1-1X/V100/1-0-1 for A2V...EO4 axial piston pump (installation pump) with DK 100 swivel angle sensor

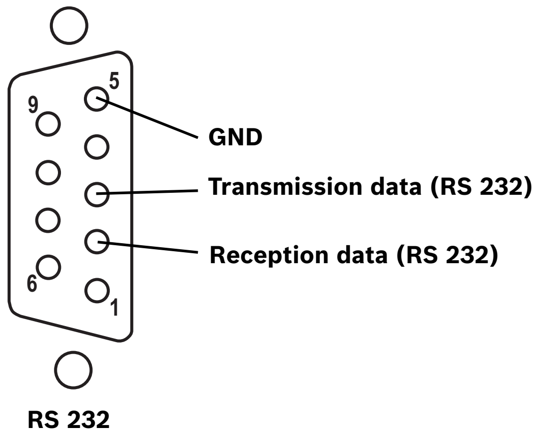
Row d		
Pin	**Short denomination**	**Description**
2	DI1	Command value call-up 1, H active
4	DI2	Command value call-up 2, H active
6	DI3	Command value call-up 4, H active
8	DI4	Command value call-up 8, H active
10	DI5	Slave mode, H active
12	DI6	DI1 to DI5 valid, H active
14		n.c.
16		n.c.
18	DI9	Enable, H active
20	DO1	Swivel angle controller active, H active
22	OK	OK output, H active
24	Data +	Local bus
26	DO2	Pressure controller active, H active
28	Data –	Local bus
30	AO1	Control of valve with OBE, ±10 V
32	AO2	Resulting swivel angle command value for master/slave operation

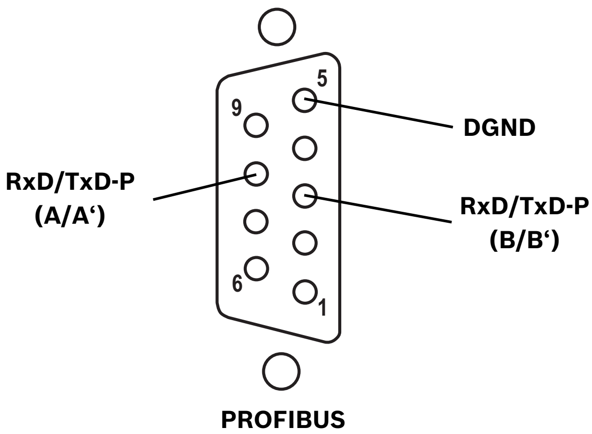
Row b		
Pin	**Short denomination**	**Description**
2	AI3+	Slave swivel angle command value (in case of slave operation)
4	AI3–	Slave swivel angle command value, reference
6		n.c.
8		n.c.
10		n.c.
12		n.c.
14	AI4+	Swivel angle command value
16	AI4–	Swivel angle command value, reference
18	AI5+	Pressure command value
20	AI5–	Pressure command value, reference
22	AI6+	Actual pressure value p_B, (I or U)
24	AI6–	Actual pressure value p_B, reference
26	AO3	Test output (measuring socket X1)
28	AGND	Analog GND
30	REF–	Reference voltage –10 V
32	REF+	Reference voltage +10 V

Row z		
Pin	**Short denomination**	**Description**
2	MA+	Solenoid A +
4	MA–	Solenoid A –
6	MB+	Solenoid B +
8	MB–	Solenoid B –
10	Shield	Shield
12	L1O–	Position transducer of valve, feed –, pin 2
14	L1I–	Position transducer of valve, actual value –, pin 4
16	L1I+	Position transducer of valve, actual value +, pin 3
18	L1O+	Position transducer of valve, feed +, pin 1
20	System earth	System earth
22	DO3	Power limitation active, H active
24	DO4	Slave mode active, H active
26	DO5	$\vert \alpha_{command} - \alpha_{actual} \vert$ < window, H active
28	DO6	$\vert p_{command} - p_{actual} \vert$ < window, H active
30	UB	Supply voltage $+U_B$
32	LO	Supply voltage 0 V

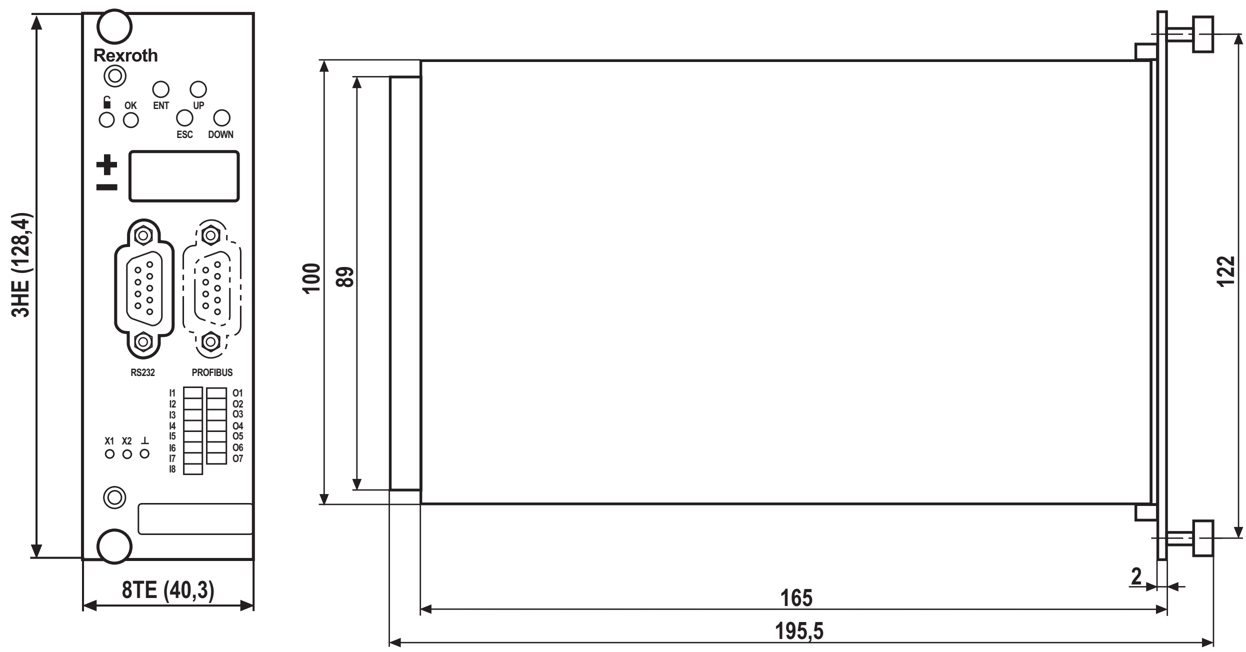
Row f		
Pin	**Short denomination**	**Description**
2	DO7	Rectangular 32 Hz
4		n.c.
6		n.c.
8	AI7+	Command value power
10	AI7–	Command value power, reference
12		n.c.
14		n.c.
16		n.c.
18		n.c.
20	L2O–	Swivel angle sensor of pump, feed –, pin 4
22	L2I–	Swivel angle sensor of pump, actual value –, pin 3
24	L2I+	Swivel angle sensor of pump, actual value +, pin 2
26	L2O+	Swivel angle sensor of pump, feed +, pin 1
28		n.c.
30		n.c.
32		n.c.

Pin assignment of the D-Sub sockets on the front plate





Dimensions (dimensions in mm)



Technical data (for applications outside these parameters, please consult us!)

Valve 4WRE6-2X/822 for HS4 control		
Current consumption per solenoid	I_{max}	2.5 A
Control current with constant swivel angle	I_a	Solenoid a: 450 mA
	I_b	Solenoid b: 700 mA
Solenoid coil resistance:		
Cold value at 20 °C	R	2.7 Ω
Max. hot value	R	4.05 Ω
Electrical connection		Plug-in connection according to DIN EN 175301-803
Protection class according to EN 60529		IP 65
Position transducer to the valve 4WRE6-2X/822		
Carrier frequency	f	5 kHz
Coil resistance (at 20 °C):		
Between ports 1 and 2	R	150 ± 11 Ω
Between ports 3 and 4	R	50 ± 3.5 Ω
Electrical connection		Plug-in connection according to DIN 43650-BFZ-Pg9
Protection class of the plug-in connection according to EN 60529		IP 65
Swivel angle sensor type AWX F004 D01		
Carrier frequency	f	5 kHz
Coil resistance (at 20 °C):		
Between ports 1 and 2	R	110 Ω
Between ports 3 and 4	R	560 Ω
Electrical connection		Plug-in connection according to DIN 43650-BFZ-Pg9
Protection class of the plug-in connection according to EN 60529		IP 65
Closed-loop control quality of the HS4 control		
Hysteresis	%	≤ 0.2
Repetition accuracy	%	≤ 0.2
Linearity deviation of the swivel angle	%	≤ 1.0
Linearity deviation of the pressure	%	≤ 1.5 of the maximum measuring pressure of the pressure transducer

Technical data (For applications outside these parameters, please consult us.)

VT-VPCD-1-1X/.../1-0-1 control electronics		
Operating voltage	U_B	24 VDC
Operating range		
Upper limit value	$U_B(t)_{max}$	30 V
Lower limit value	$U_B(t)_{min}$	21 V
Current consumption	I_{max}	3.5 A
Oscillator frequency (valve position transducer, swivel angle)	f	Approx. 5 kHz at 10 V_{SS}
Digital inputs	Signal	log 0 = 0 to 5 V log 1 = 16 V to U_B
Digital outputs	Signal	log 0 = 0 to 5 V log 1 = U_B – 3 V I_{max} = 30 mA, short-circuit-proof
Analog inputs AI1...AI7 can be configured as voltage input		
AI3, AI4	U	±10 V
AI1, AI2, AI5, AI6, AI7	U	0 to 10 V
Input resistance	R_e	100 kΩ
Resolution	U	5 mV for range ±10 V, 2.5 mV for range 0 to 10 V
Non-linearity	U	< 10 mV
Analog inputs AI2, AI4, AI5 and AI6 can be configured as current inputs		
Range	I	4 to 20 mA
Input resistance	R_e	100 Ω (voltage drop at actual pressure value input with 4 mA approx. 1.7 V, with 20 mA approx. 3.5 V)
Leakage current		0.15 % (with 500 Ω between pin AI x - and 0 V)
Resolution	I	5 µA [12 bit]
Analog outputs AO1, A02 and A03		
Output voltage	U	±10 V
Load	$R_{L\,min}$	1 kΩ
Resolution	U	10 mV (11 bit)
Residual ripple	U	±25 mV (without noise)
Reference voltage		
Voltage	U	±10 V
Current	I_{max}	30 mA
Residual ripple	U	< 20 mV
Scan time	T	2 ms
Serial interface		RS 232 (front plate), D-Sub socket 9-pole
Type of connection		64-pole male multipoint connector, DIN 41612, design G
Local bus, distance to the furthermost device	l	Max. 280 m line length
Card dimensions		Euro-card 100 x 160 mm, DIN 41494
Front plate dimensions:		
Height		3 HE (128.4 mm)
Width soldering side		1 TE (5.08 mm)
Width component side		7 TE (35.56 mm)
Admissible ambient temperature range	ϑ	0 to 50 °C
Storage temperature range	ϑ	−20 to +70 °C
Weight	m	0.2 kg

Notice:
For information on the **environment simulation testing** for the areas EMC (electromagnetic compatibility), climate and mechanical load, see data sheet 30028-U.

Project planning/maintenance instructions/additional information

Product documentation for the VT-VPCD

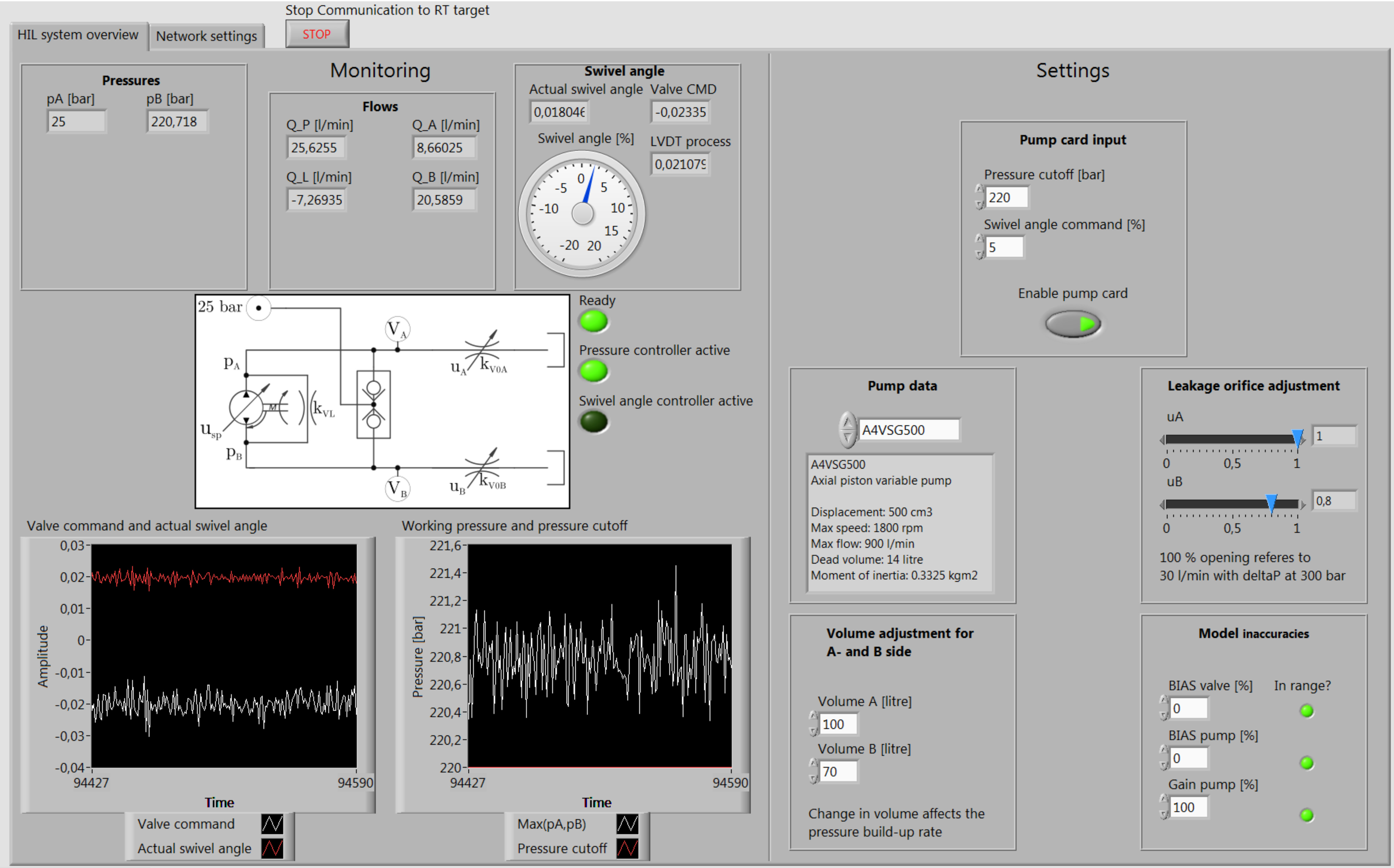
- 30028
  Technical data sheet (this document)
- 30028-B
  Installation and operating instructions
- 30028-01-B
  Commissioning and operating instructions
- 30028-U
  Environmental compatibility statement
- 30028-01-Z
  Commissioning instructions, PROFIBUS interface
- 30028-02-Z
  Commissioning instructions, CANopen interface
- 30028-03-Z
  Commissioning instructions, DeviceNet interface

- The control electronics may only be unplugged and plugged when de-energized.
- Only carry out measurements at the card using instruments with R_i > 100 kΩ.
- For switching analog command values and digital call-ups, use relays with gold-plated contacts (low voltages, low currents).
- Always shield command and actual value cables; connect shielding to system earth on the card-side, open at one side.
- Recommendation: Up to a length of 50 m, use the line type LiYCY 1.5 mm^2 for solenoid line, for position transducer line use cable type LiYCY 0.5 mm^2, shielded. For greater lengths, please contact us.
- The distance to aerial lines or radios must be at least 1 m!
- Do not lay solenoid conductors and signal lines near power lines.
- Commissioning and programming of the control electronics are described in detail in the operating instructions 30028-B.
- For perfect control results, the quality of the sensors is important.

Bosch Rexroth AG
Hydraulics
Zum Eisengießer 1
97816 Lohr am Main, Germany
Phone +49 (0) 93 52 / 18-0
documentation@boschrexroth.de
www.boschrexroth.de

Appendix B

Graphical User Interface



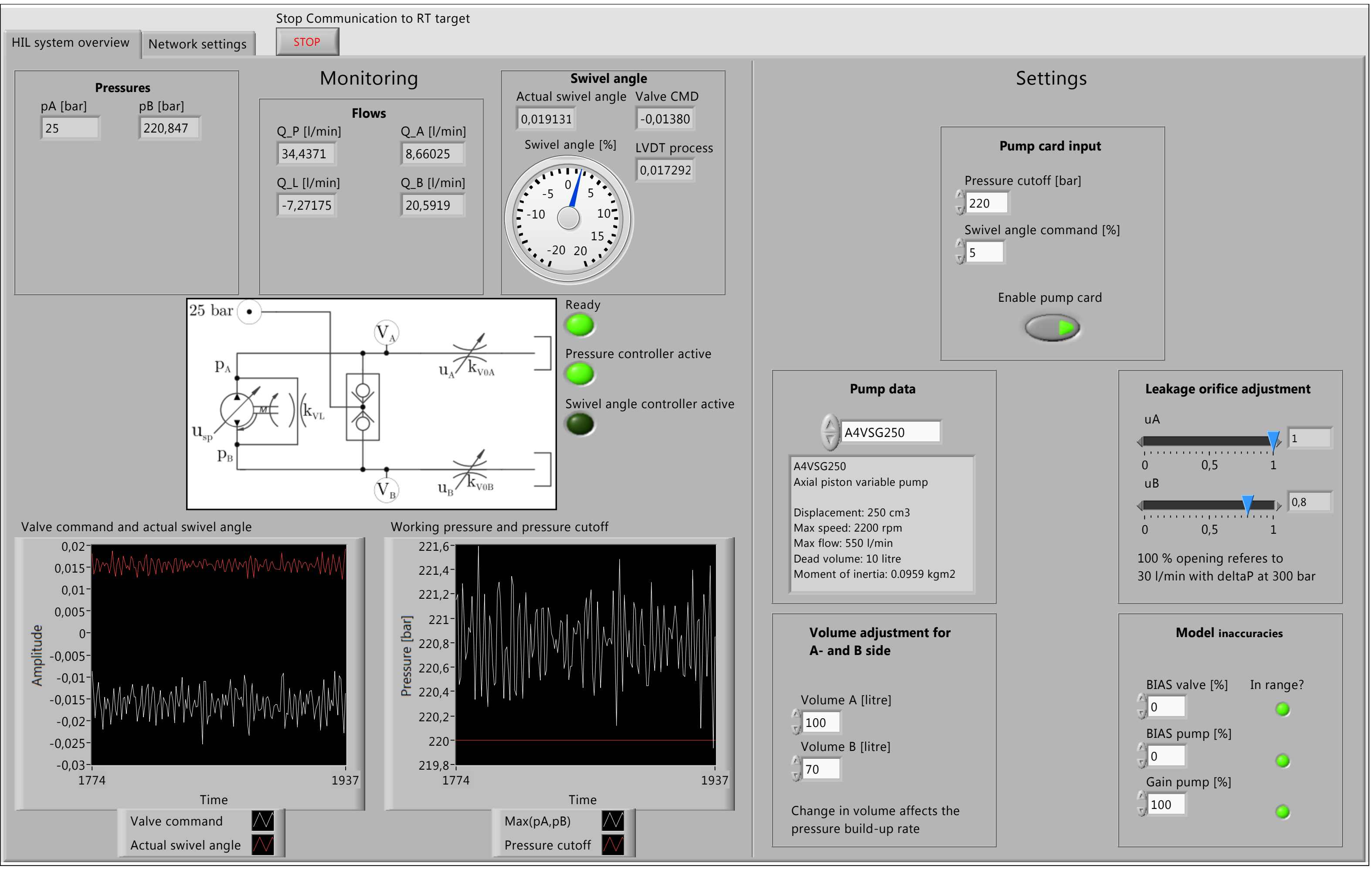
HIL system overview
Network settings
Stop Communication to RT target
STOP
Monitoring
Pressures
pA [bar]
25
pB [bar]
220,718
Flows
Q_P [l/min]
25,6255
Q_A [l/min]
8,66025
Q_L [l/min]
-7,26935
Q_B [l/min]
20,5859
Swivel angle
Actual swivel angle
0,018046
Valve CMD
-0,02335
Swivel angle [%]
LVDT process
0,021079
25 bar
Ready
Pressure controller active
Swivel angle controller active
Valve command and actual swivel angle
Amplitude
Time
Valve command
Actual swivel angle
Working pressure and pressure cutoff
Pressure [bar]
Time
Max(pA,pB)
Pressure cutoff
Settings
Pump card input
Pressure cutoff [bar]
220
Swivel angle command [%]
5
Enable pump card
Pump data
A4VSG500
A4VSG500
Axial piston variable pump
Displacement: 500 cm3
Max speed: 1800 rpm
Max flow: 900 l/min
Dead volume: 14 litre
Moment of inertia: 0.3325 kgm2
Leakage orifice adjustment
uA
1
uB
0,8
100 % opening referes to
30 l/min with deltaP at 300 bar
Volume adjustment for
A- and B side
Volume A [litre]
100
Volume B [litre]
70
Change in volume affects the
pressure build-up rate
Model inaccuracies
BIAS valve [%]
0
In range?
BIAS pump [%]
0
Gain pump [%]
100

Appendix C

LabVIEW Program

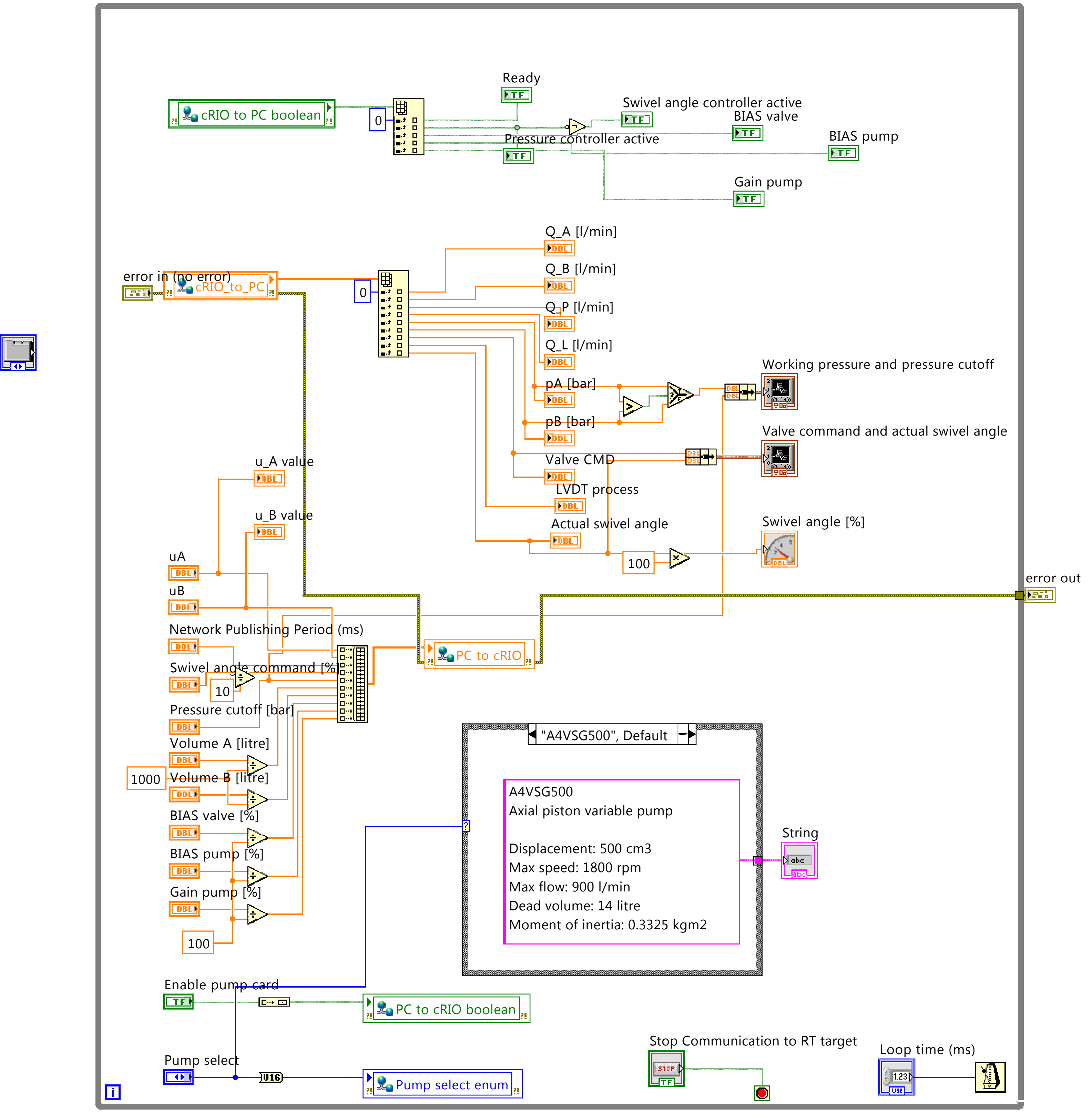
HOST MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\HOST MAIN.vi
Last modified on 29.04.2015 at 14:28
Printed on 05.05.2015 at 11:06

Front Panel



HOST MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\HOST MAIN.vi
Last modified on 29.04.2015 at 14:28
Printed on 05.05.2015 at 11:06

Block Diagram



cRIO MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\cRIO MAIN.vi
Last modified on 05.05.2015 at 10:21
Printed on 05.05.2015 at 11:06

Front Panel

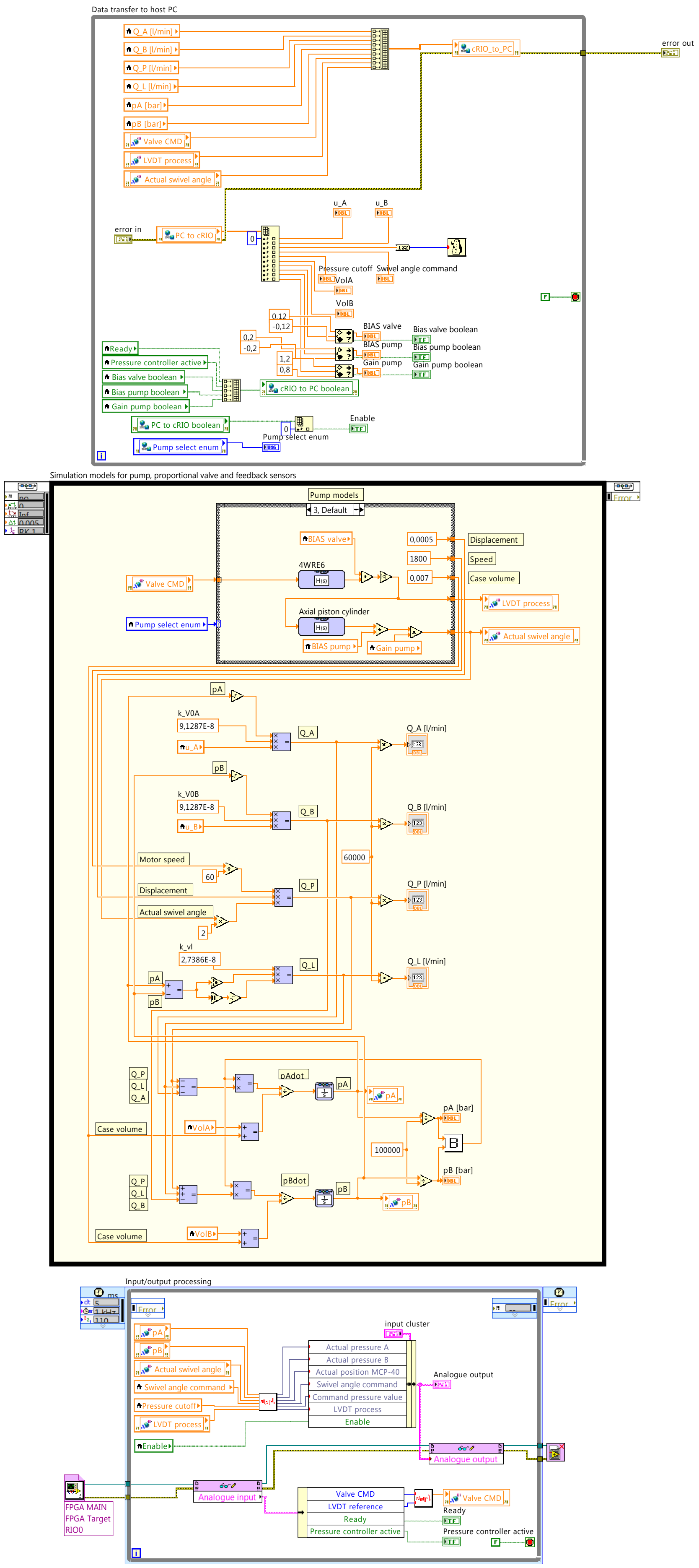
cRIO MAIN.vi

C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\cRIO MAIN.vi

Last modified on 05.05.2015 at 10:21

Printed on 05.05.2015 at 11:06

Block Diagram

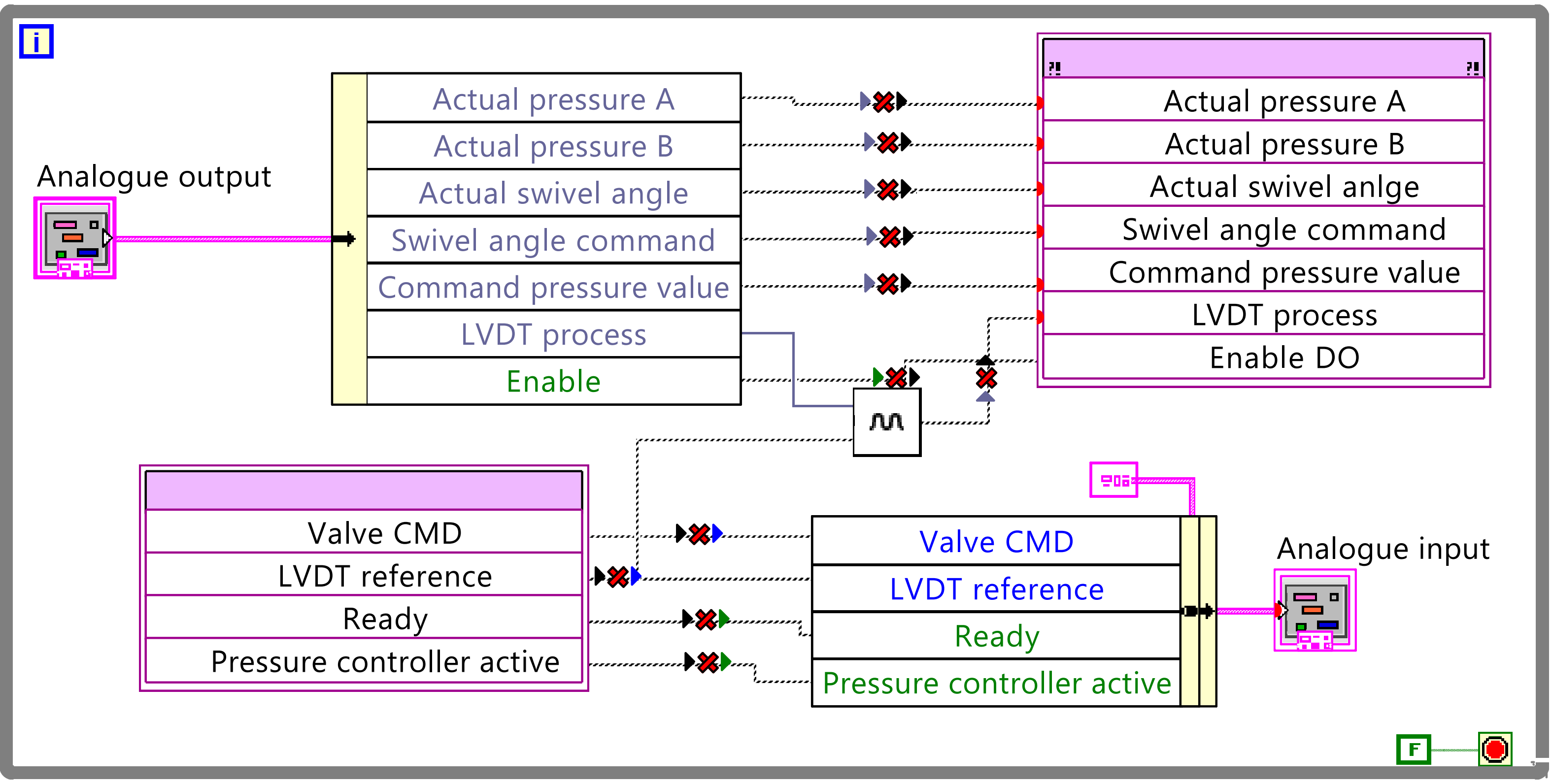


FPGA MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\FPGA MAIN.vi
Last modified on 05.05.2015 at 09:22
Printed on 05.05.2015 at 11:06

Front Panel

FPGA MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\FPGA MAIN.vi
Last modified on 05.05.2015 at 09:22
Printed on 05.05.2015 at 11:06

Block Diagram



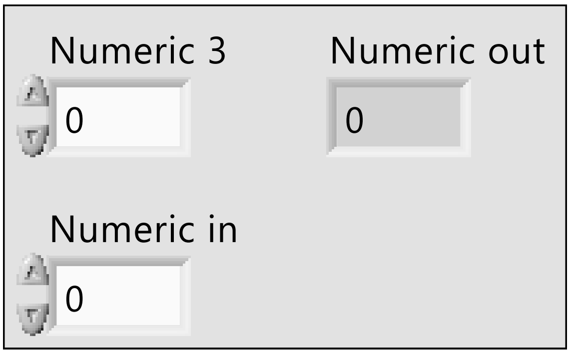
Beta calc.vi

C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Beta calc.vi

Last modified on 24.04.2015 at 15:01

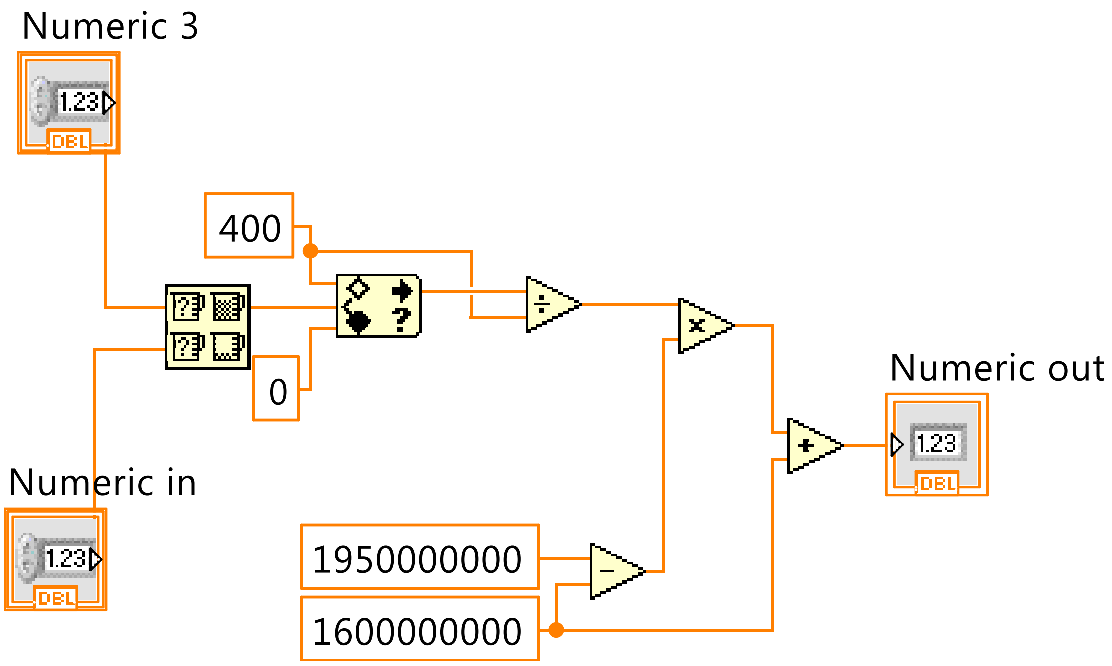
Printed on 05.05.2015 at 11:06

Front Panel



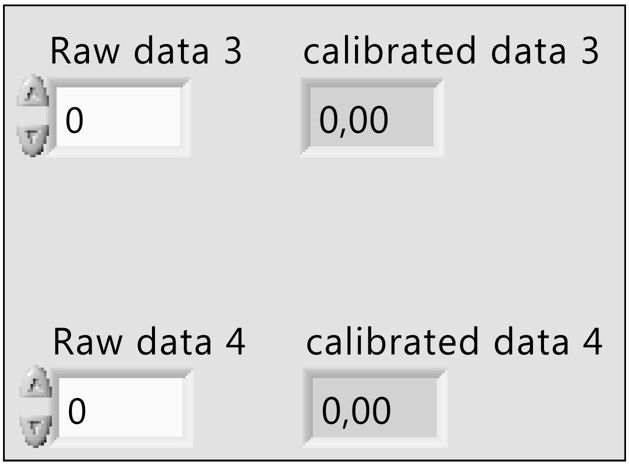
Beta calc.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Beta calc.vi
Last modified on 24.04.2015 at 15:01
Printed on 05.05.2015 at 11:06

Block Diagram



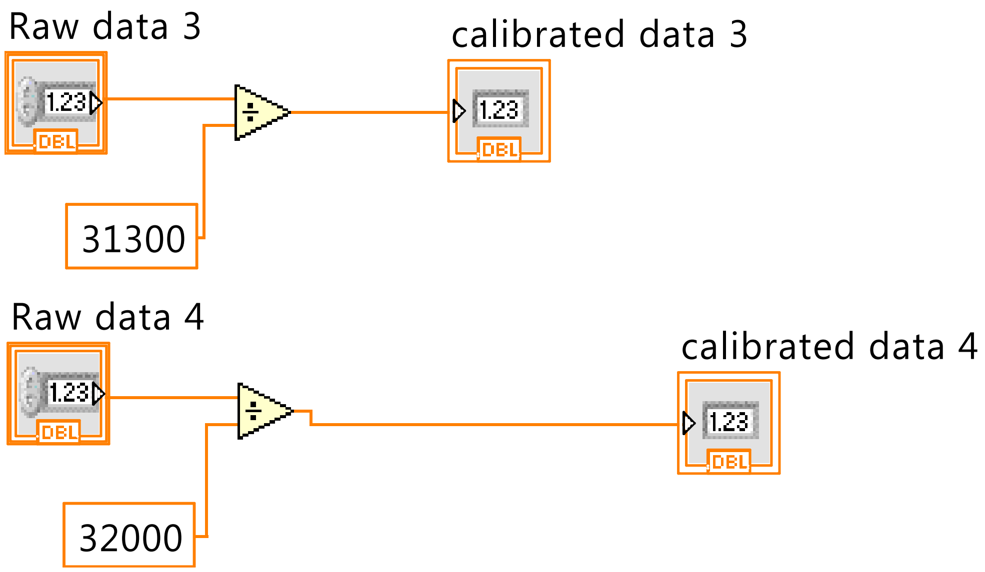
Input scale2.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Input scale2.vi
Last modified on 05.05.2015 at 09:12
Printed on 05.05.2015 at 11:06

Front Panel



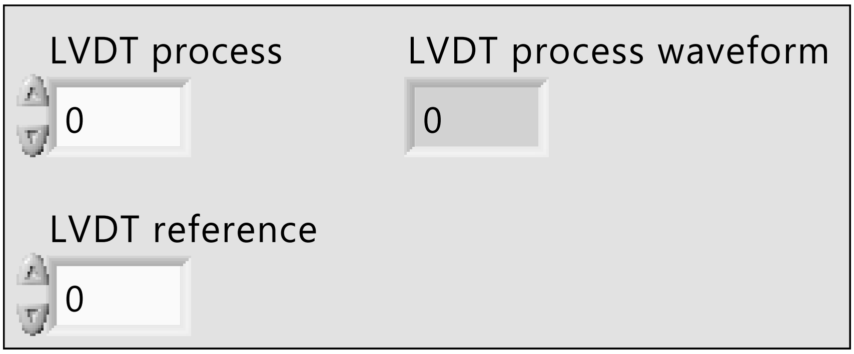
Input scale2.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Input scale2.vi
Last modified on 05.05.2015 at 09:12
Printed on 05.05.2015 at 11:06

Block Diagram



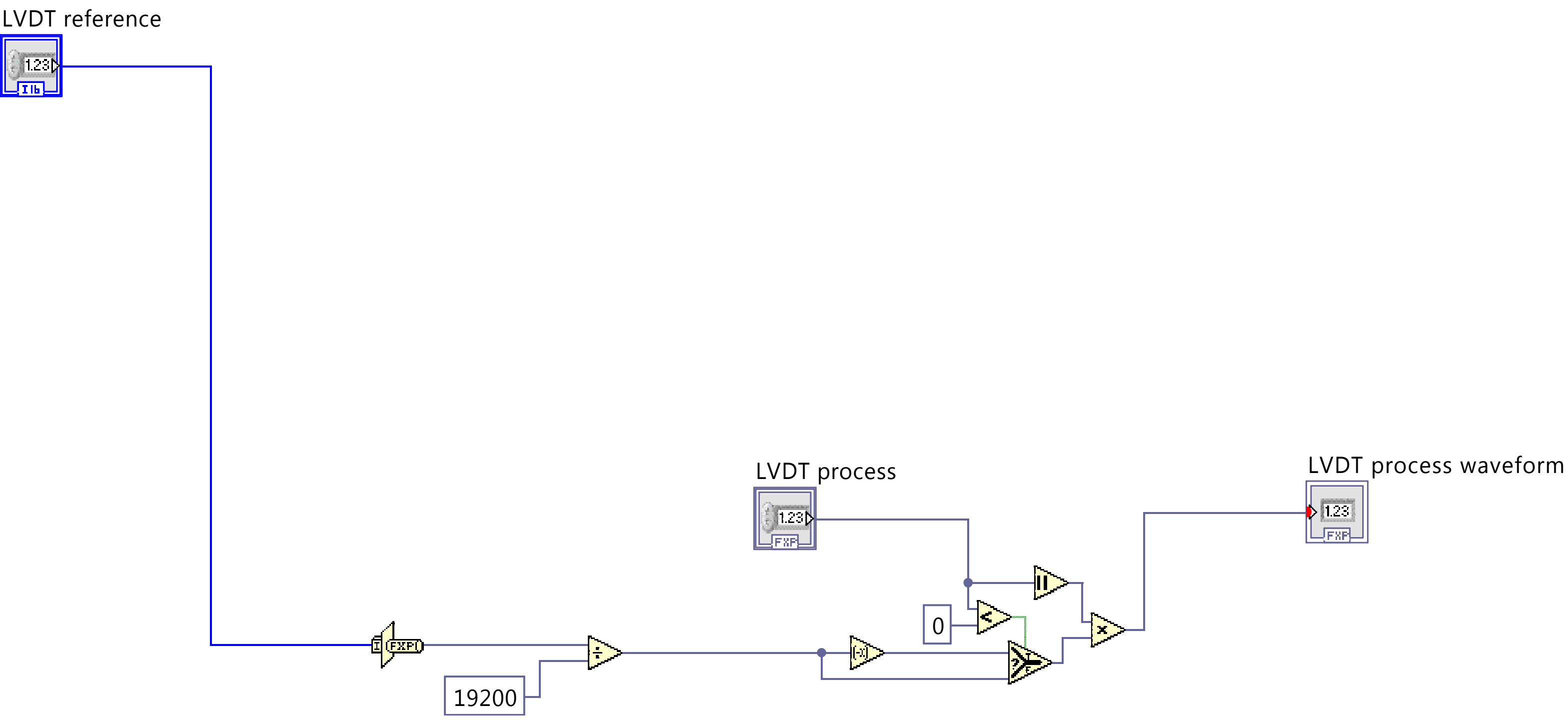
LVDT scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\LVDT scale.vi
Last modified on 24.04.2015 at 13:48
Printed on 05.05.2015 at 11:06

Front Panel



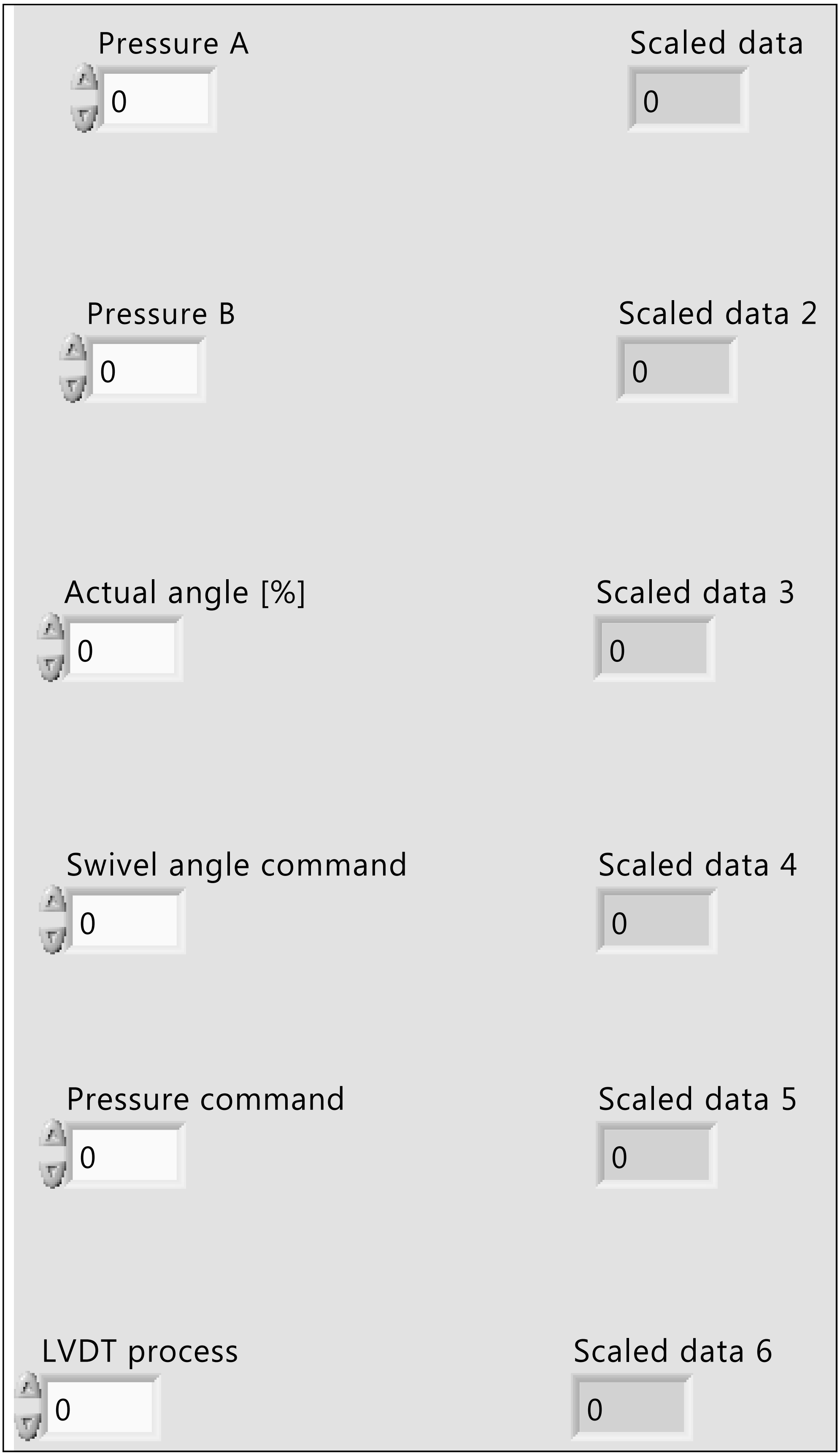
LVDT scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\LVDT scale.vi
Last modified on 24.04.2015 at 13:48
Printed on 05.05.2015 at 11:06

Block Diagram



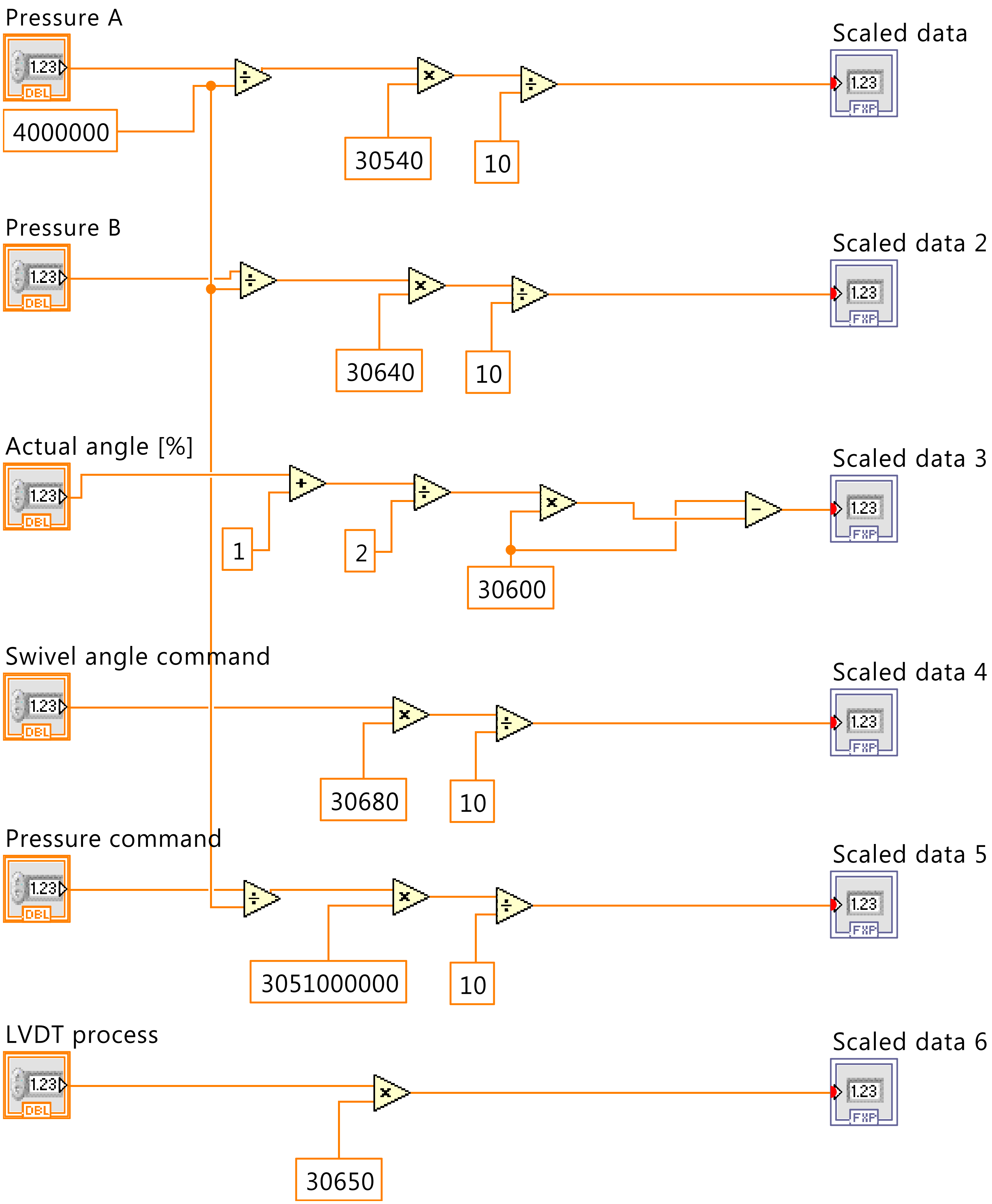
Output scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Output scale.vi
Last modified on 04.05.2015 at 10:03
Printed on 05.05.2015 at 11:06

Front Panel



Output scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\HIL system\Output scale.vi
Last modified on 04.05.2015 at 10:03
Printed on 05.05.2015 at 11:06

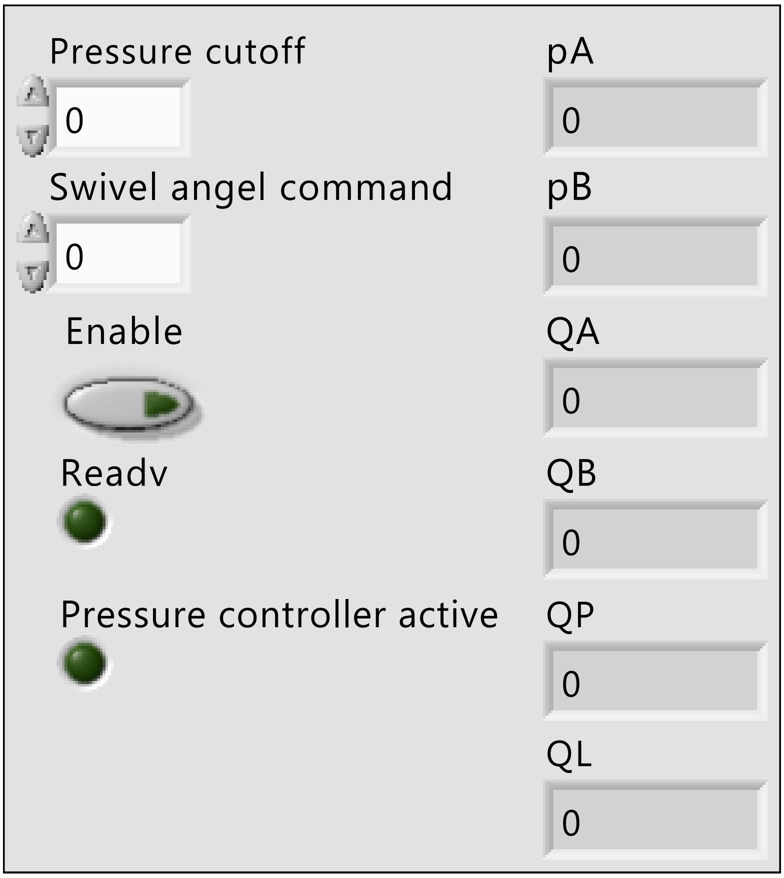
Block Diagram

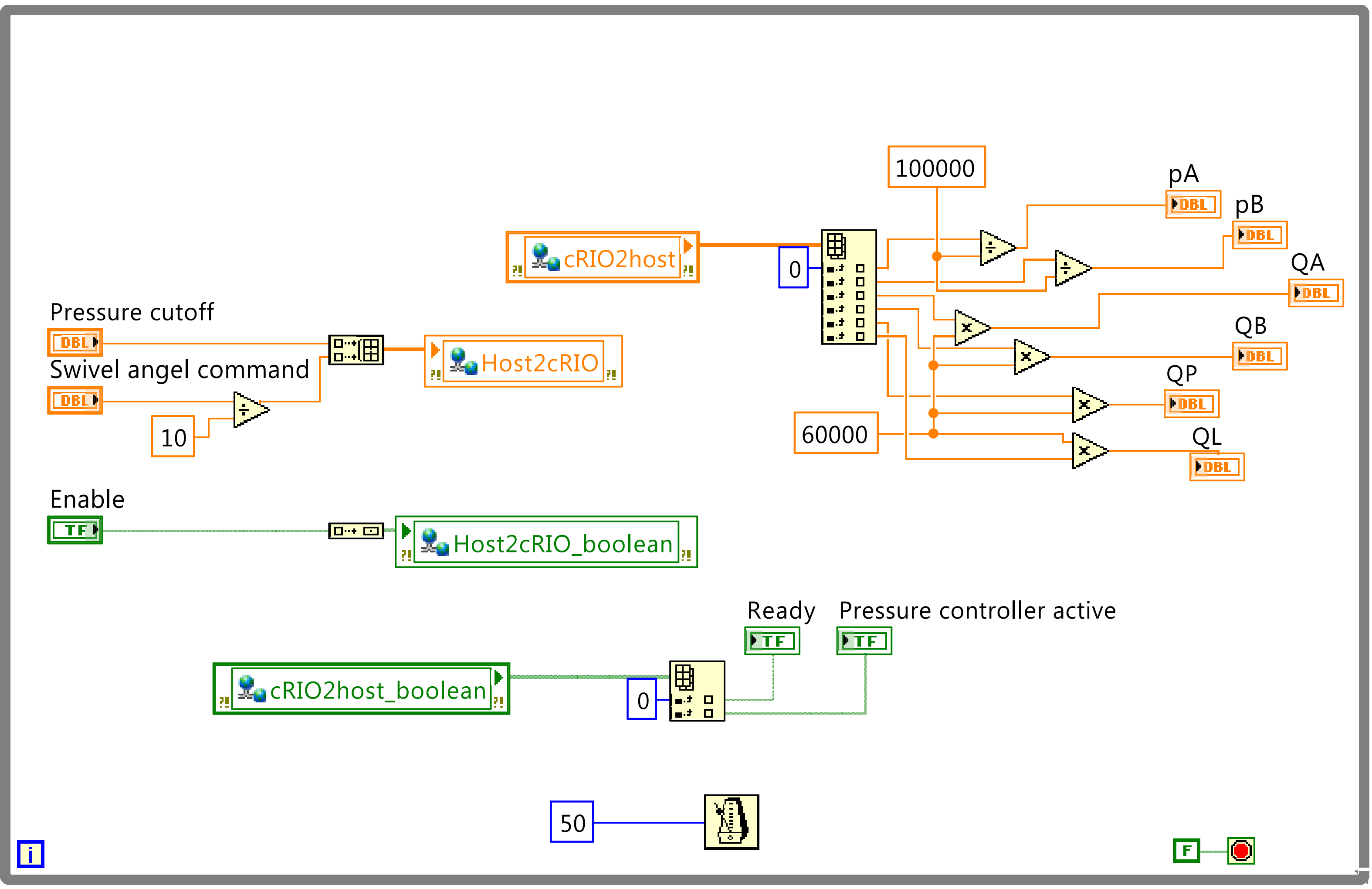


Appendix D

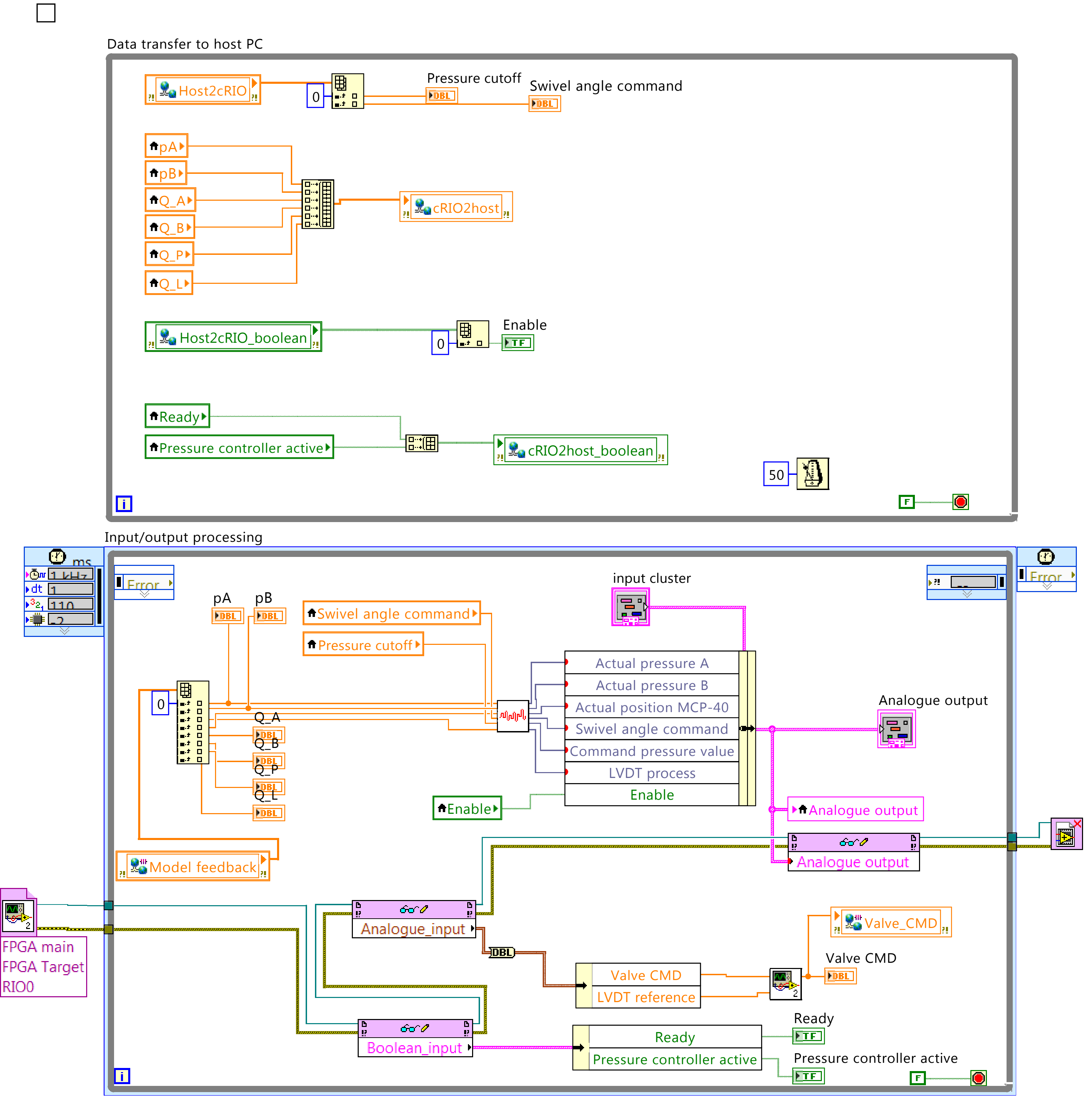
LabVIEW Program for External RT Target

Host main.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\Host main.vi
Last modified on 05.05.2015 at 11:18
Printed on 05.05.2015 at 11:25

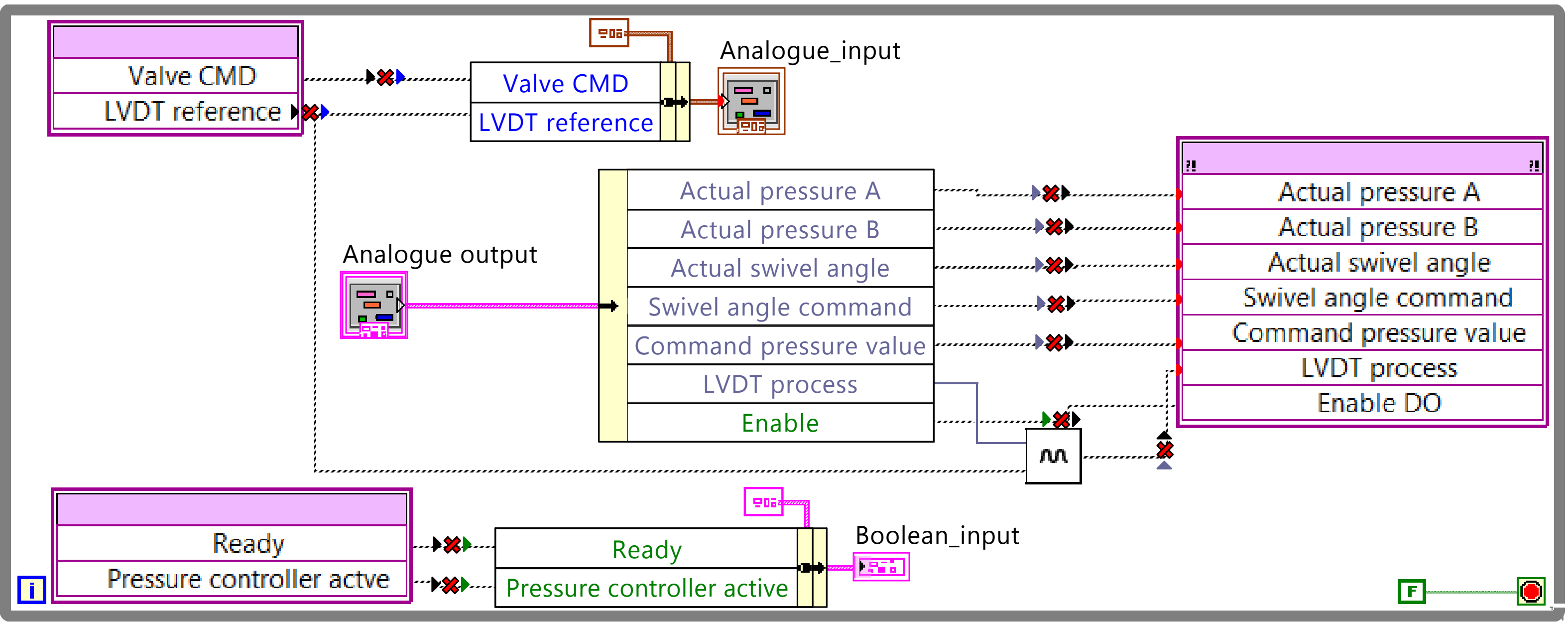




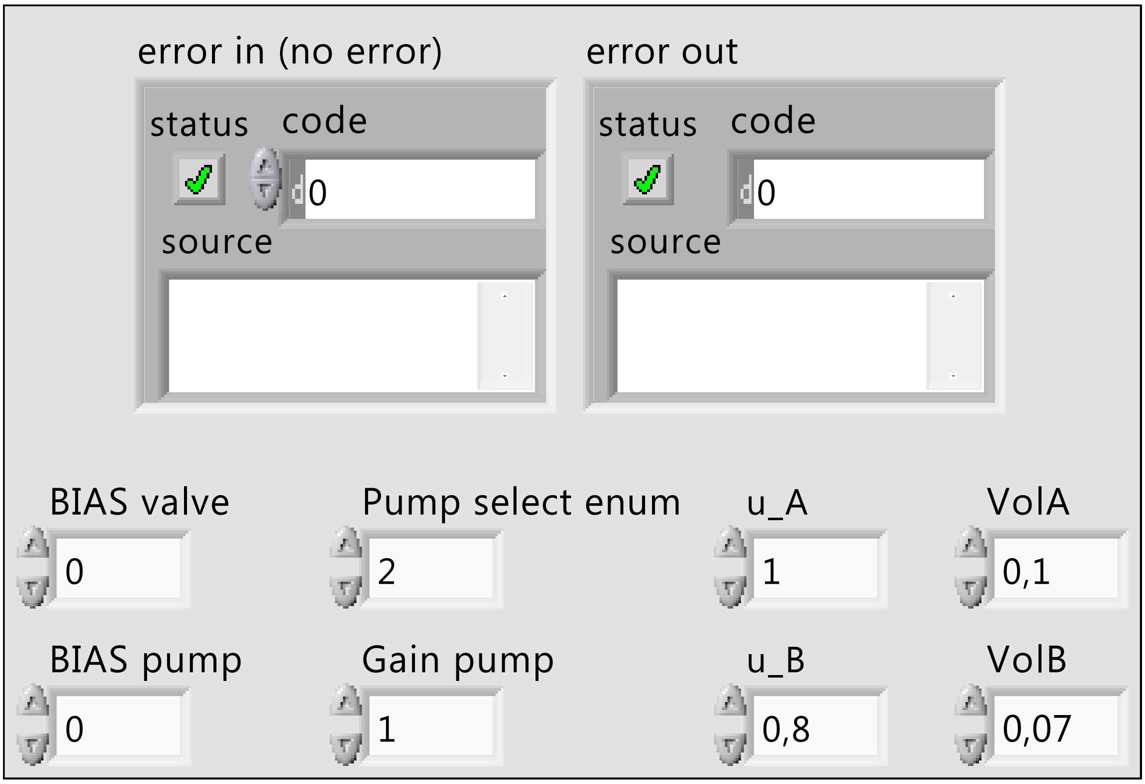
cRIO MAIN.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\cRIO MAIN.vi
Last modified on 05.05.2015 at 11:19
Printed on 05.05.2015 at 11:25



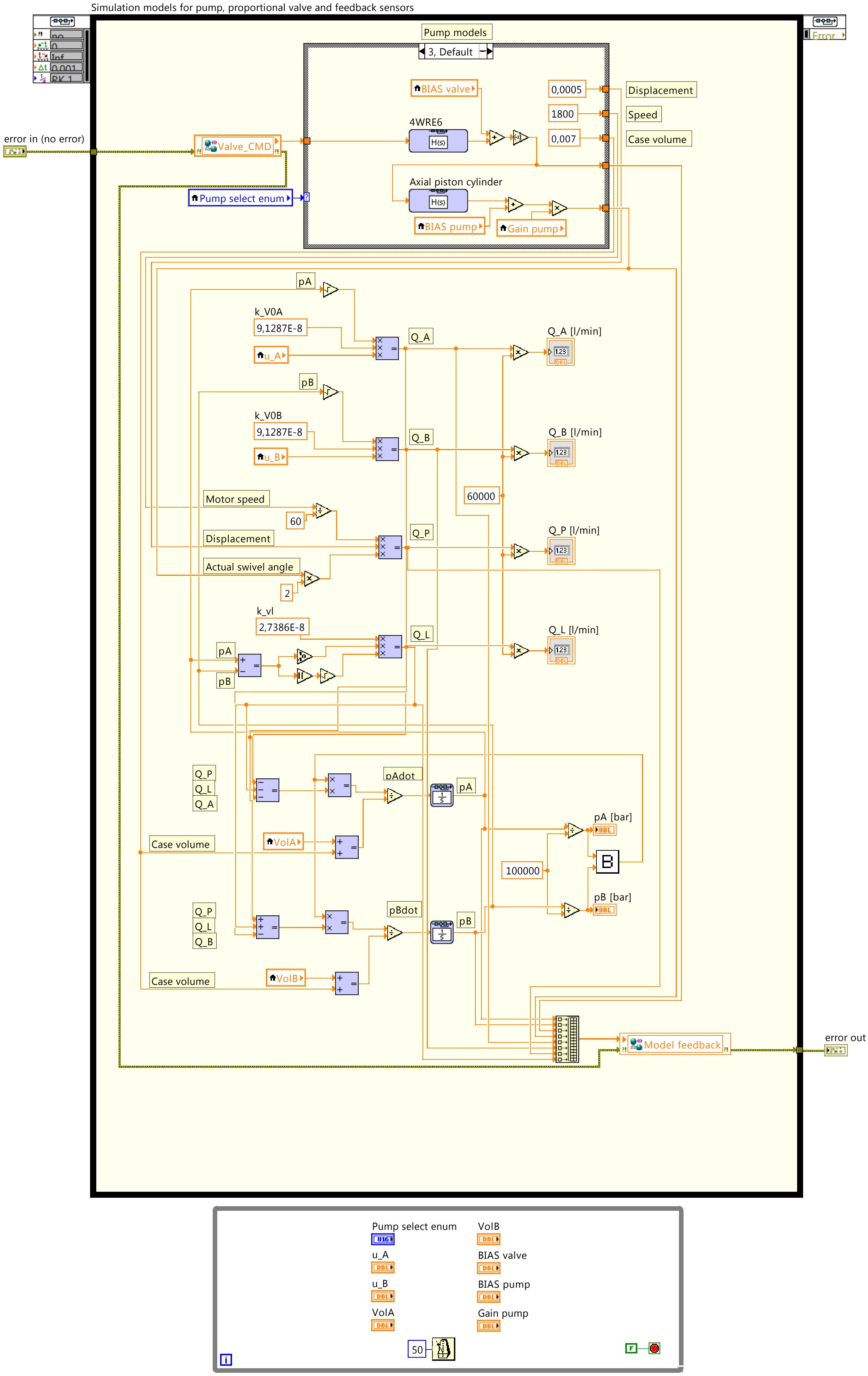
FPGA main.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\FPGA main.vi
Last modified on 05.05.2015 at 08:42
Printed on 05.05.2015 at 11:25



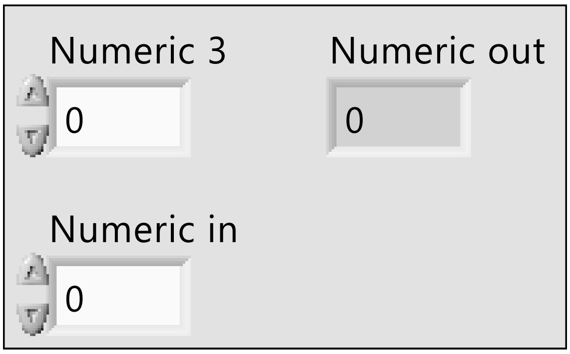
RT main.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\RT main.vi
Last modified on 05.05.2015 at 11:23
Printed on 05.05.2015 at 11:25

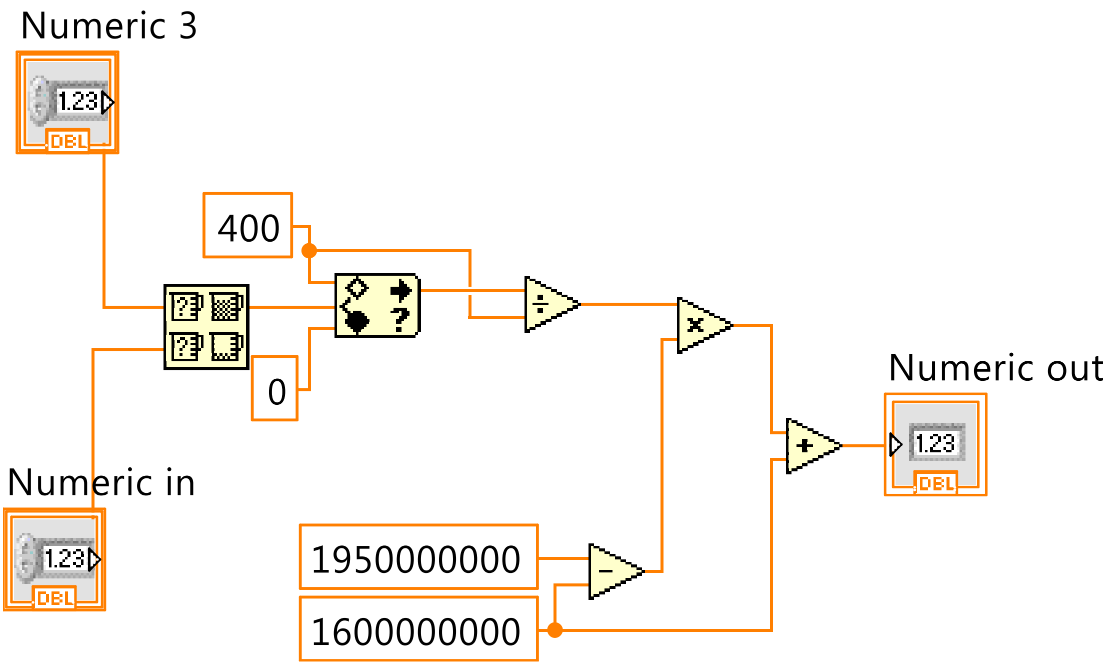


RT main.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\RT main.vi
Last modified on 05.05.2015 at 11:23
Printed on 05.05.2015 at 11:25



Beta calc.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\Beta calc.vi
Last modified on 24.04.2015 at 15:01
Printed on 05.05.2015 at 11:25



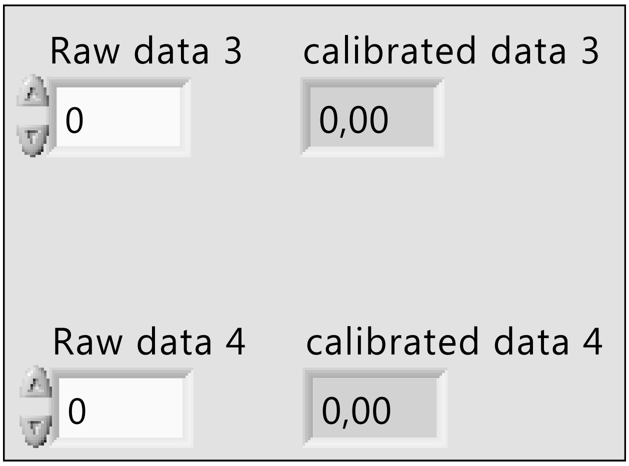


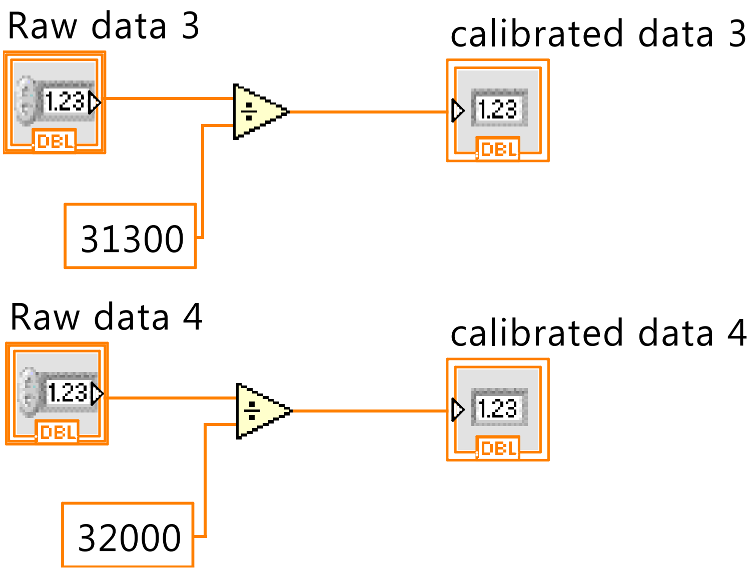
Input scale.vi

C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\Input scale.vi

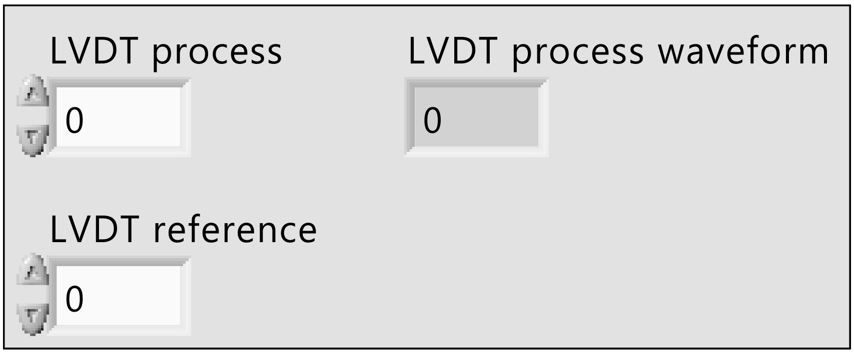
Last modified on 05.05.2015 at 11:23

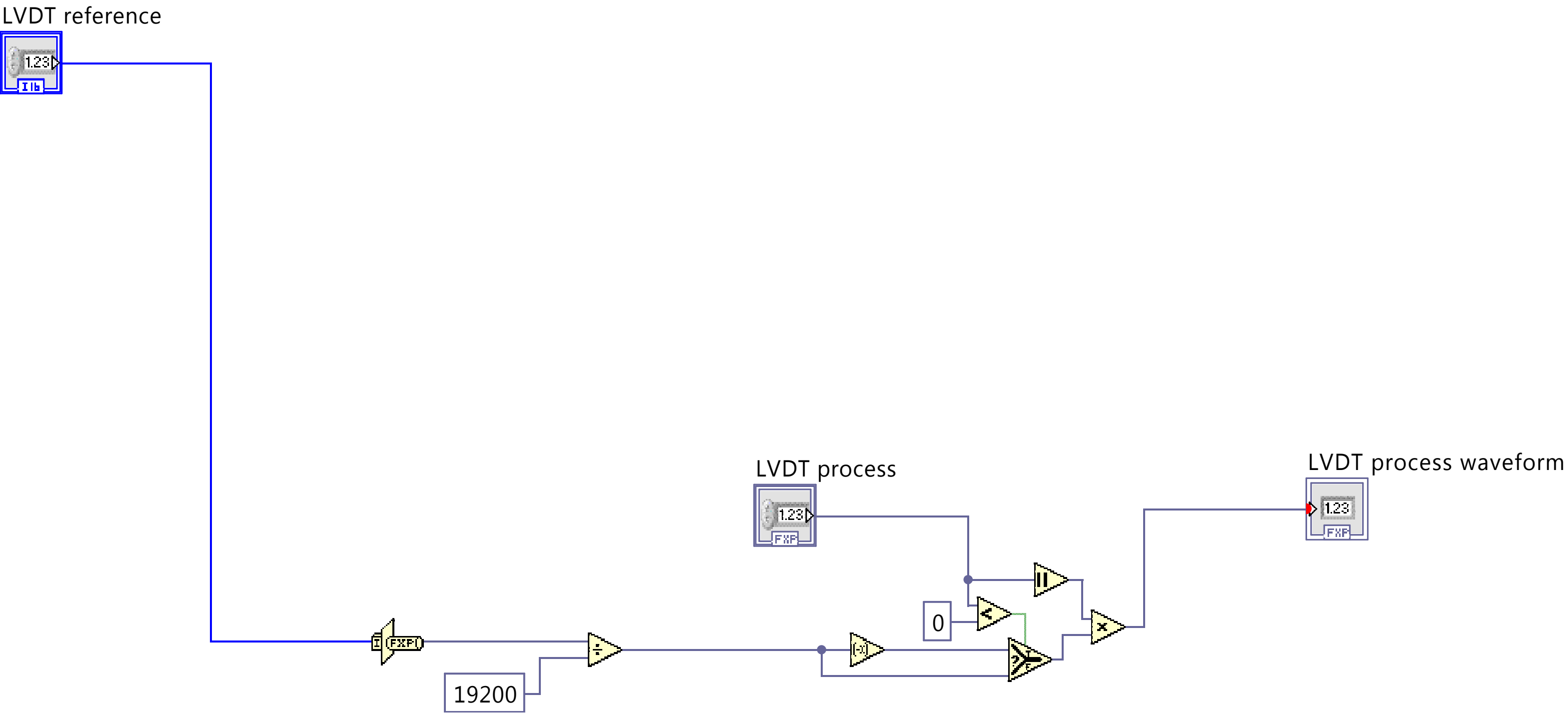
Printed on 05.05.2015 at 11:25



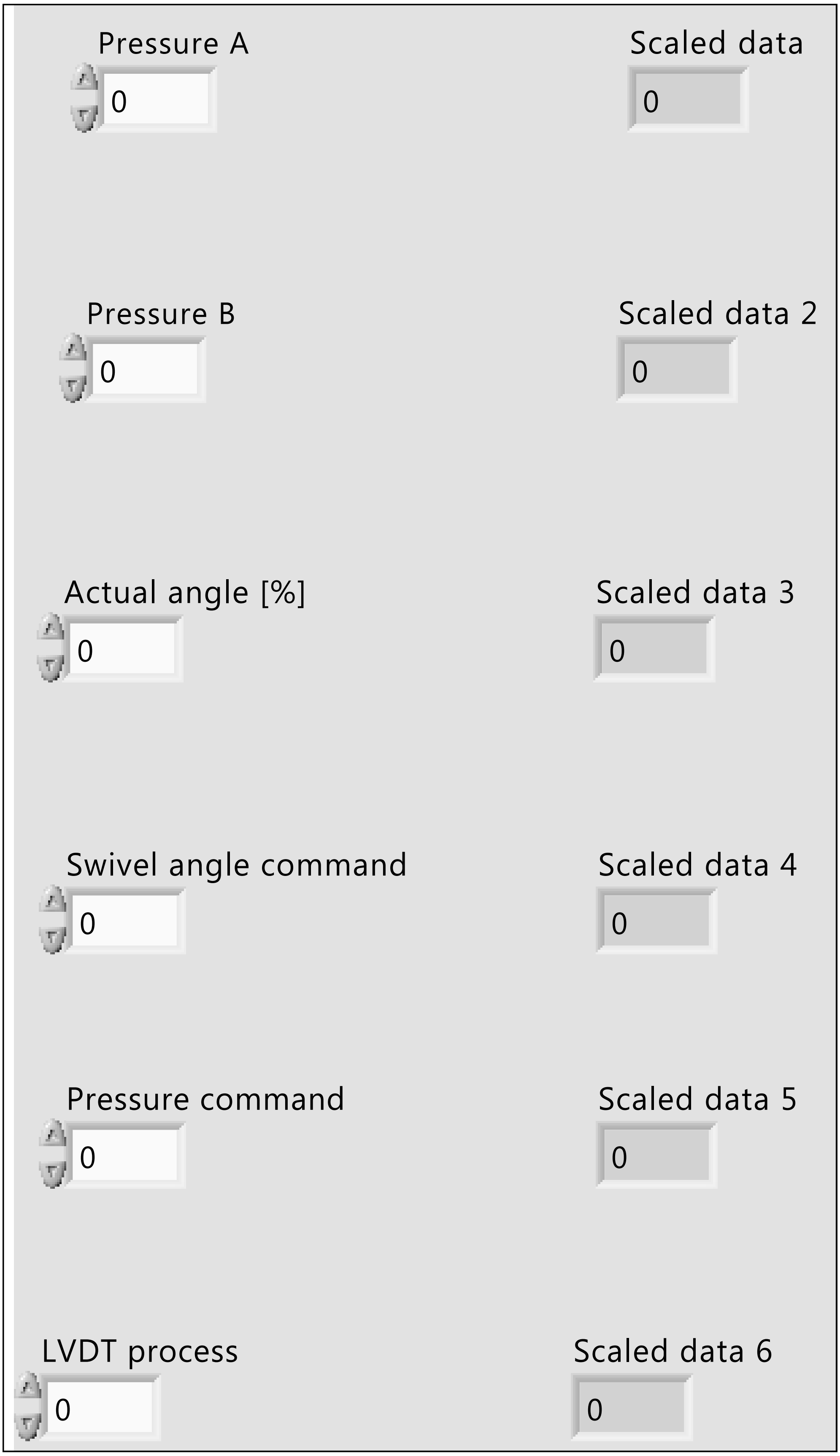


LVDT scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\LVDT scale.vi
Last modified on 05.05.2015 at 08:42
Printed on 05.05.2015 at 11:25

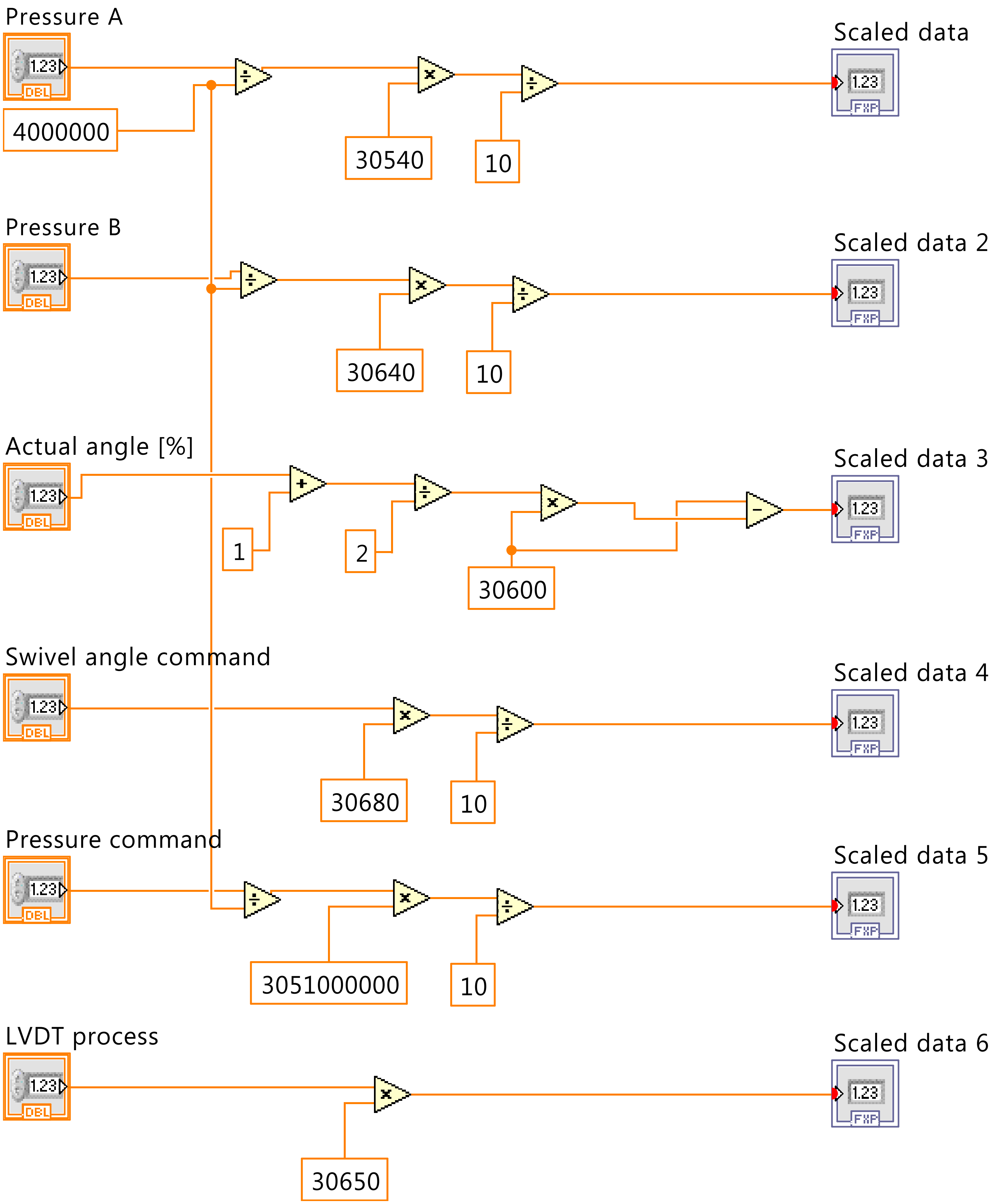




Output scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\Output scale.vi
Last modified on 05.05.2015 at 11:23
Printed on 05.05.2015 at 11:25



Output scale.vi
C:\Users\BoO2\Dropbox\Master2015\LabVIEW\External RT\Output scale.vi
Last modified on 05.05.2015 at 11:23
Printed on 05.05.2015 at 11:25



Appendix E

HIL System User Manual

User manual

HIL Simulation for Pump Control

RIG/PLANT		
ADDITIONAL CODE	SDRL CODE	TOTAL PGS 8
REMARKS		
MAIN TAG NUMBER		DISCIPLINE
CLIENT PO NUMBER		
CLIENT DOCUMENT NUMBER		

REFERENCE	REFERENCE DESCRIPTION
	Crane Hydraulics Kristiansand

National Oilwell Varco Norway AS
Postbox 401, Lundsiden
N-4604 Kristiansand
Norway

Phone +47 38 19 20 00
Fax +47 38 19 26 04

DOCUMENT NUMBER	REV
	0

NATIONAL OILWELL VARCO

www.nov.com

REVISION HISTORY

0	19.5.2015	Internal	BoO		
Rev	**Date (dd.mm.yyyy)**	**Reason for issue**	**Prepared**	**Checked**	**Approved**

CHANGE DESCRIPTION

Revision	**Change description**
0	

TABLE OF CONTENTS

NB! This document shall under no circumstances be sent to the customer, subcontractor or similar.

1 GENERAL

This user manual is applicable for the following equipment:
Pump Controller Card: VT – VPCD Digital Electronic Controller
Real-time controller: NI CompactRIO-9075
Cable: Clas Ohlson art. No. 32-5853. Crossed network cable
Software: Bodac, LabVIEW
http://www.boschrexroth.com/business_units/bri/en/downloads/index.jsp

1.1 Contact persons:

NOV , **Norway:**
Ørjan Bø: orjan.bo@nov.com

1.2 Firmware update

Latest firmware edition: Tools/utilities/send firmware

FW update VPCD with L1 Bootstrap.ms

VT-VPCD-1-1x_V0_1-0-1_L04-B23-2.95-1.

2 SETTING UP THE HIL SIMULATION

1) Install the real-time software application on the instructor PC. An EXE file start the application.
2) Connect the instructor PC to the CompactRIO with via crossed network cable.
3) Install Bodac on the PC used for education. This might be the same PC as the instructor PC.
4) Connect the PC used for education to VT-VPCD pump card via RS232 cable.
5) Power up both the CompactRIO and pump card.

3 CONFIGURING SIMULATION MODEL

The simulation model has various configuration parameters. Figure 1 shows the hydraulic circuit used for setting up the dynamic simulation model.

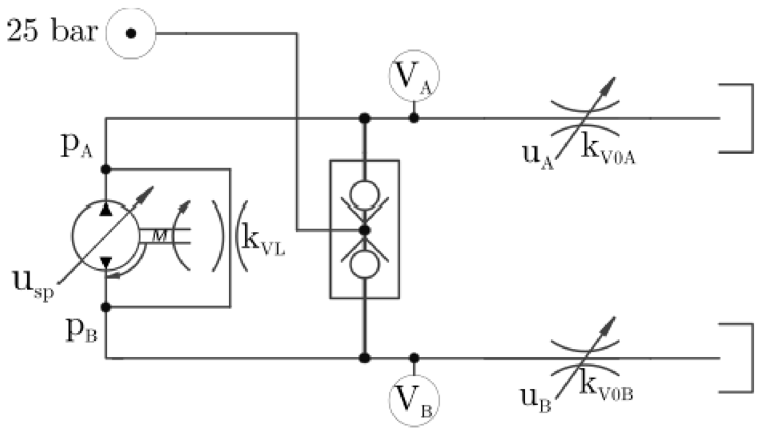


Figure 1: Hydraulic circuit for the dynamic simulation model

From the settings panel, shown in Figure 2, configuration parameters related to the hydraulic control circuit and the hydraulic leakage circuit can be operated. The pump type can also be changed from this panel. The instructor configures the simulation model to wanted configuration.

Start the EXE file. The application sends and receives data from the real-time target where the simulation model is executed.

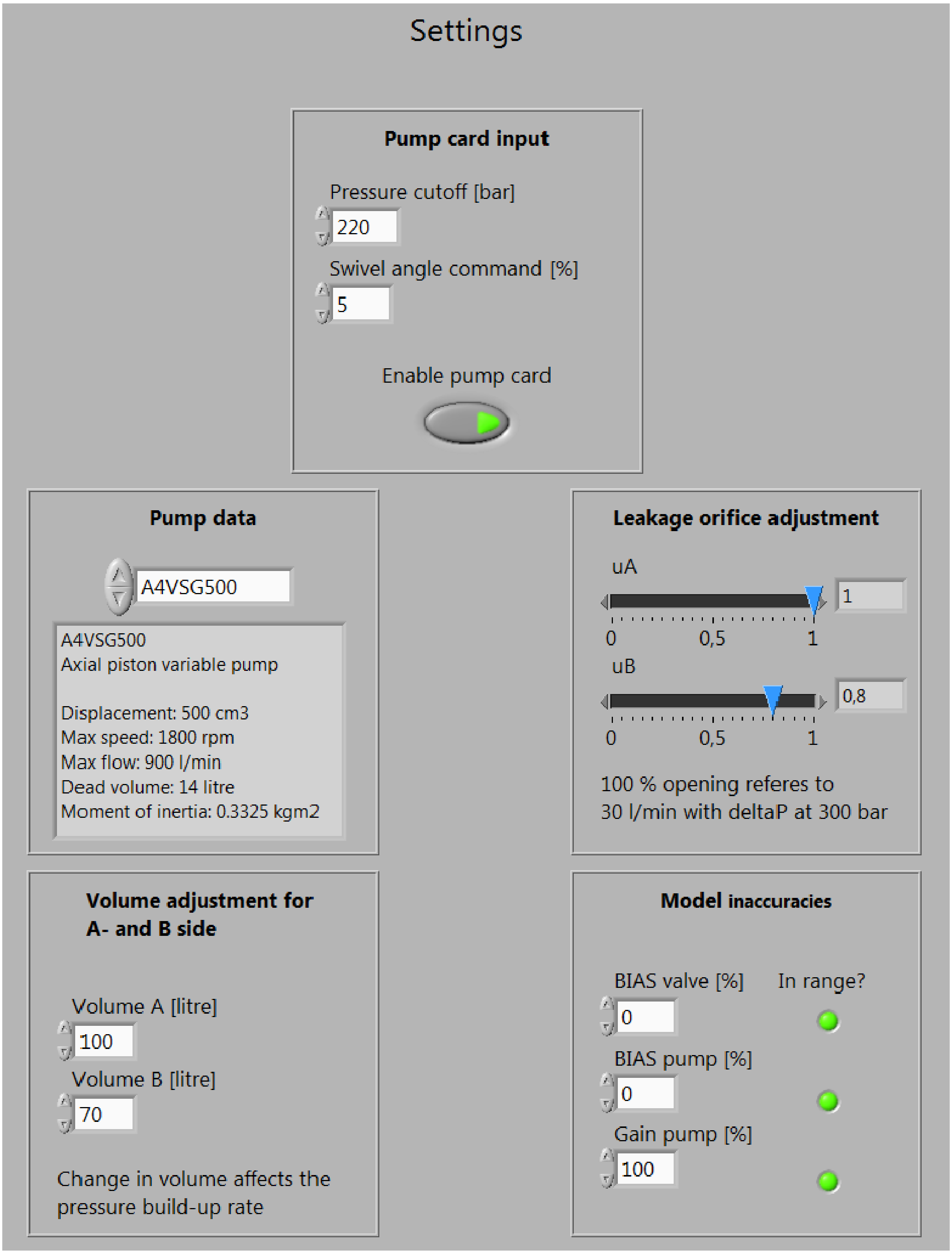


Figure 2: Settings panel on the real-time applications graphical user interface

4 TUNING PUMP CARD

1) Open Bodac. Bodac automaticly connects to the pump card. If not, check communication port: Start\Devices and Printers in Windows. Tools\Prefrences\Com in Bodac. The communication port must be the same in Windows and Bodac.
2) Update firmware to latest version.
3) Follow internal procedure for pump card tuning of HIL simulation.