HIL and MC lab handbook

TMR4243

January 10, 2015

Contents

Eirik: Skriv ordentlig om labview/XML delen av FPGA programmeringen

Nomenclature

cRIO National Instruments compact reconfigurable input/output real-time embedded industrial controller

RPi Raspberry Pi single-board computer

VI virtual instrument, a LabVIEW program

1 Introduction

Hva skal denne labben handle om?

1.1 Structure

Part I describes the laboratory facilities and equipment. A general overview of hardware and software.

Part II gives the necessary theoretical background for the lab work.

Part III is a user guide intended for students of the course. Step-by-step instructions for development and deployment of programs to the real-time controller are given. Lower level details, intended for laboratory assistants, are given in Part V.

Part I
Laboratory description/equipment

2 MC-Lab

The Marine Cybernetics Laboratory is the newest test basin at the Marine Technology Centre. It is located in what was originally a storage tank for ship models made of paraffin wax.

As the name indicates, the facility is especially suited for tests of marine control systems, due to the relatively small size and advanced instrumentation package. It is also suitable for more specialised hydrodynamic tests, mainly due to the advanced towing carriage , which has capability for precise movement of models in 6 degrees of freedom.

The MCLab is operated by the Department of Marine Technology, and has been a Marie Curie EU Training Site (2002-2008). It is mainly used by Master and PhD-students, but it is also available for MARINTEK and external users.

The software in use was developed using rapid prototyping techniques and automatic code generation under Matlab/Simulink and Opal. The target PC onboard the vessel runs the QNX real-time operating system while experimental results are presented in real-time on a host PC using Labview.

Sakset rett fra nettsiden http://www.ntnu.edu/imt cybernetics-lab

2.1 Towing carriage

Carriage : towing speed 2 m/s, 5 (6) DOFs forced motions Current generation: 0-0.15m/s

2.2 Wave generator

The wave genertor is located at the end of the tank and is operated from its own copmuter. It has the capability to create first order Stoke waves or irregular recreate different wave spectras such as JONSWAP or PM spectras.
Significant wave height $H_s = 0.3$ [m] with period T between 0.6 [s] and 1.5 [s]

2.2.1 First order Stoke waves

First order stoke waves are regular linear waves. Very nice to do calculations with, but not so representative for real life conditions. Described by potential theory

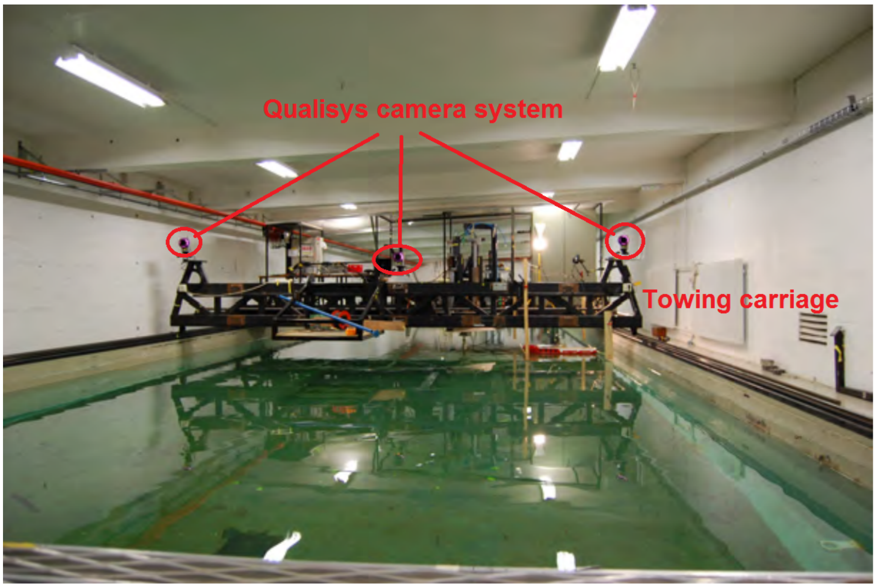

Figure 1: Towing carriage

2.2.2 Irregular waves

2.3 Qualisys positioning system

The positioning system works by tracking reflectors placed on the ship with the use of high speed cameras.
Qualisys consists of three systems

- Qualisys Oqus: The cameras used to register/see the IR markers
- Qualisys Motion Capture Systems: is the system that process the data from Oqus
- Qualisys Track Manager: The userinterface to interact with Motion Capture System

2.4 Ethernet communication network

For communication with the ship, the wireless network HILlab is used

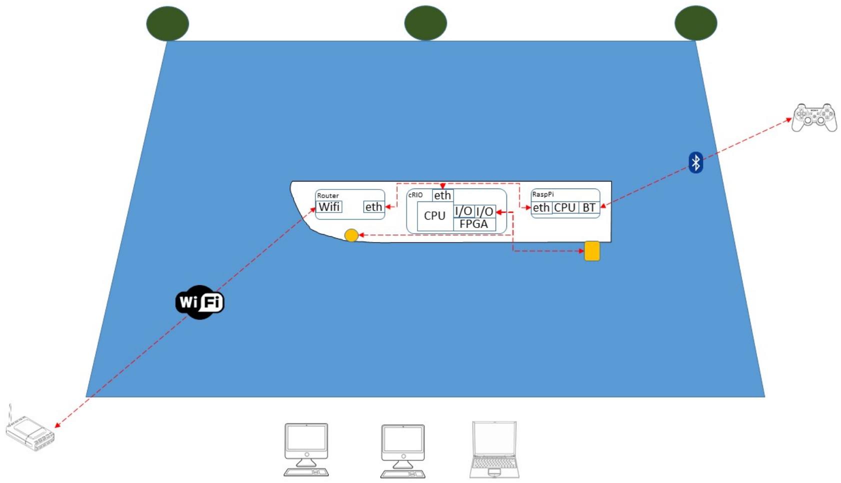

Figure 2: Towing carriage

3 CSE1 - CS Enterprise I

Figure 3: C/S Enterprise I

3.1 Control system

3.1.1 Real time hardware

CompactRIO

CompactRIO is a reconfigurable embedded control and acquisition system. The CompactRIO system's rugged hardware architecture includes I/O modules, a reconfigurable FPGA chassis, and an embedded controller. Additionally, CompactRIO is programmed with NI LabVIEW graphical programming tools and can be used in a variety of embedded control and monitoring applications.
For more info vistit the producer website

Sakset fra ni.com/compactrio

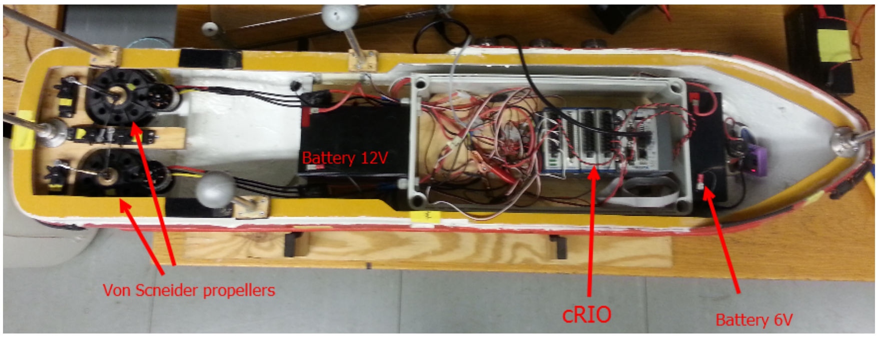

Figure 4: CSE1 - Hardware

3.1.2 Real time software

LabVIEW - Real time module
National Instruments real-time technology offers reliable, deterministic performance for your time-critical applications. Use the LabVIEW Real-Time Module to develop and deploy complex real-time systems quickly and efficiently to the CompactRIO microprocessor.
Veristand

3.1.3 Input/Output signals

PWM

Digital output

3.1.4 NI Labview user interface

3.1.5 Startup procedure

Connecting to Cybership Enterprise 1 (RT CompactRIO - NI-cRIO9024-CSE1 (192.168.0.77))

Nam sin prosedyre fra readme.txt

1. Place "Main battery" (large fat one) beneath wireless anntenna, adjacent to waterproof box, between the wires, with battery terminals furthest away from it.

2. Place "Servo battery" (small slim one) at bow between tunnel thruster and waterproof box, with battery terminals closes to the waterproof box.

3. Postive battery terminal ("RED-port") at portside and negative battery terminal ("BLACK-port") at starboard side
Connect wire with red isolation ("RED-wire") to "RED-port" and wire with black isolation ("BLACK-wire") to "BLACK-port"

4. Connect first the "RED-wire" before the "BLACK-wire" to the batteries.

5. The "Main battery" (large fat one) should be connected first then wait a few sec (5s) before connecting the "Servo battery" (small slim one).

6. Note: it should not matter in which order it is done, but from experience connectiong “RED-wire” before “BLACK-wire” gives a much higher probability for communication with the CompactRIO on Cybership Enterprise 1 (99-100%’ish) than connecting the “BLACK-wire” before the “RED-wire” (25%ish), and it is a habit to connect main before the servo, since main powers “CompactRIO” while servo powers “D-Link wireless bridge”
7. There should be 3 red lights lighting up, one at bow in a purple box for indicatiing power to tunnel thuster two close to “Main battery”, one on each side for each Voith Schneider propeller
8. The indicators on “ACT/LiNK” port 1 should light up (green) to indicate communication with “HILlab”
9. Test communication:
   - Open “Command Promt”
   - write: ping 192.168.0.77

   A successfull ping should return somthing like

   ```
   C:\Documents and Settings\mcl>ping 102.168.0.77

   Pinging 192.168.0.77: bytes=32 time = 5ms TTL=64
   Pinging 192.168.0.77: bytes=32 time = 5ms TTL=64
   Pinging 192.168.0.77: bytes=32 time = 5ms TTL=64
   Pinging 192.168.0.77: bytes=32 time = 2ms TTL=64

   Ping statistics for 192.168.0.77:
   Packets: Sent = 4, Received = 4, Lost = 0 <0% loss>,
   Approximate round trip times in milli-seconds:
   Minimum =2ms, Maximum = 5ms, Average = 4ms
   ```
10. The most imprtant thing is that you receive packets in return, the time might vary but the important thing is that it responds to the ping.
11. If Lost = 100% meaning no repons means either “Laptop” or “CompactRIO” is unable to communicate with “HILlab”.
12. Check Laptop is connected to wireless network “HILlab”, if not connect to it “HILlab”
13. Check ACT/LiNK” port 1 are showing activity e.g. are lit, blinking, if not check ethernet cable is connected to “ACT/LiNK” port 1 and to the “D-Link Wireless Bridge” if not connect to those Battery gives power to “CompactRIO” and “D-Link”, lights/indicators are lit/blinking if not check wiring
14. Check battery voltages, “Main battery” should be 10 Volt or more, maximum around 13 Volt, regular 11 to 12 Volt, low 10 Volt “Servo battery” should be in 5 Volt or more, max around 6.4 Volt, regular around 6 Volt
15. Note: Black wire should always be the last to be connected, and “Main Battery” first

3.1.6 Template: DP control system

3.2 HIL Simulation Setup

3.2.1 Hardware topology

3.2.2 Software C/S E1

PWM tick = FPGA clock pulse

$tickinseconds = 1/frequency = 1/40MHz = 1/(40 * 10^6) = 25 * 10^-9 = 25ns$

output at 50 Hz demands output every $40MHz/50Hz = (40 * 10^6)/50 = 800000tick$

3.2.3 Startup procedure

3.3 Development procedure

3.3.1 Development for new algorithms in modules in Simulink

3.3.2 Development with structural changes in I/O or Labview UI

4 Software

MATLAB	Mathworks	
LabVIEW	National Instruments	
VeriStand	National Instruments	Interfacing
NI MAX	National Instruments	Measurement & Automation Explorer: configurering av cRIO
Qualisys?		

Table 1: Software

Hva brukes det enkelte programmet til?

Part II
Theory

5 HIL

DNV, Hardware in the Loop Testing (HIL)

Johansen, Sørensen, Experiences with HIL Simulator Testing of Power Management Systems

Smogeli, Introduction to third- party HIL testing

Johansen, Fossen, Vik, Hardware-in-the-loop Testing of DP systems

Pivano, Experiences from seven years of DP software testing

DNV, Rules for Classification of Ships (Part 6, Ch 22)

Ambrosovskaya, Approach for Advanced Testing of DP Control System

Selvam, System Verification Helps Validate Complex Integrated Systems

A. Veksler prøveforelesning:

- Increased complexity marine vessels increases the need for testing and verification.
- A reasonably new approach to this is Hardware-In-the-Loop testing, or HIL.
- Widely used in the automotive industry
- Can be seen as something in between simulation testing and full scale testing
  - More realistic than a simulation, less realistic than a full-scale testing
  - Mathematical models of the systems that are not included as hardware.
- A real-time simulator, constructed by hardware and software, that is
  - configured for the control system under consideration
  - embedded in external hardware
  - and interfaced to the target system or component through appropriate I/O
- Advantages:
  - Another layer of independent verification
  - Allows testing emergency procedures that would be too dangerous on a real vessel
- Disadvantages:
  - Initial investment to set up a HIL simulator for a particular system. The resources could be spent on simulation testing or full scale testing.

  - Supplements, but does not replace, proper software design techniques
    * As with all software testing, it typically executes only a fraction of the control system code.

Asgeir

- HIL testing is accomplished by connecting a simulation PC in the system's communication network.
- Inputs to the equipment under test are simulated.
- The controllers respond as they would in a dynamic environment.
- Simulator responds to output from the controllers as the dynamic system would
- Software (core SW and/or configuration) errors are exposed.

5.1 Real-time computing

5.2 SIL

6 Control theory

Part III
Laboratory user guide

7 Simulink model compilation

7.1 Modelling

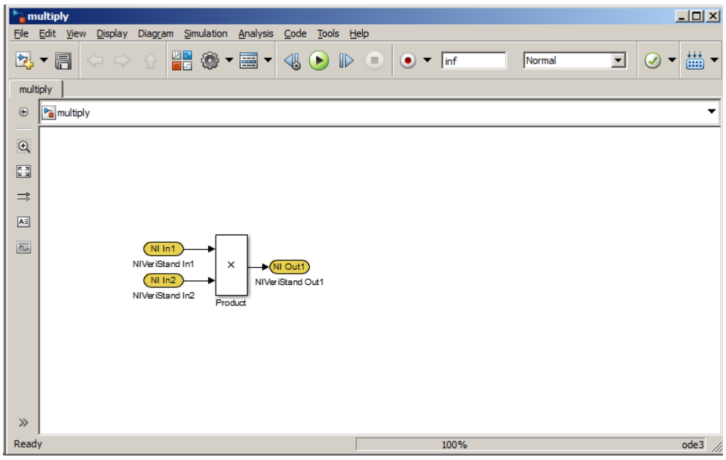

Figure 5: Simulink model for VeriStand

The Simulink model's relevant inputs and outputs should have special VeriStand ports, as seen in yellow in Figure 5. The figure shows an example used throughout this section: two inputs are multiplied to form an output.

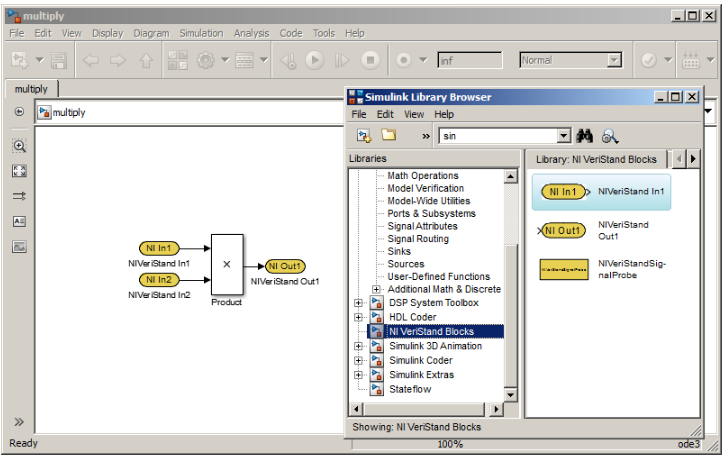

Figure 6: Simulink library

The special ports are available in the Simulink Library Browser under NI VeriStand Blocks, see Figure 6.

The model should be saved [model name].mdl

7.2 Model configuration

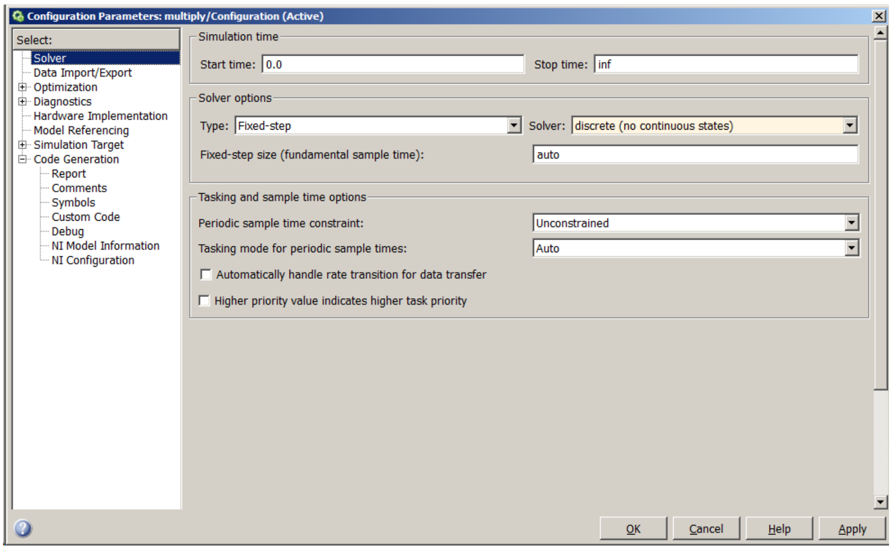

Figure 7: Simulink model configuration - solver

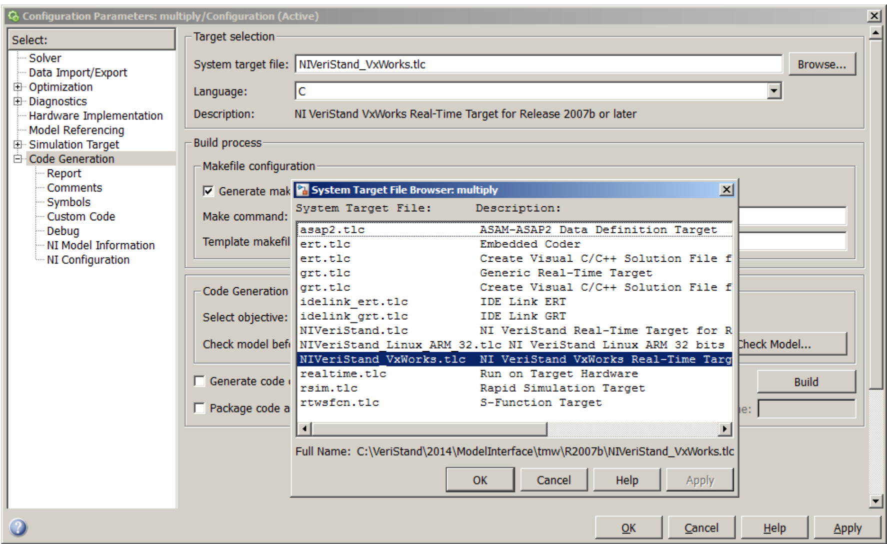

Figure 8: Simulink model configuration - target selection

1. Open the Configuration parameters window by _____
2. Configure solver as in Figure 7.
   (a) Under “Simulation time”, “Stop time” must be “inf” for infinite, since the model should not stop by itself

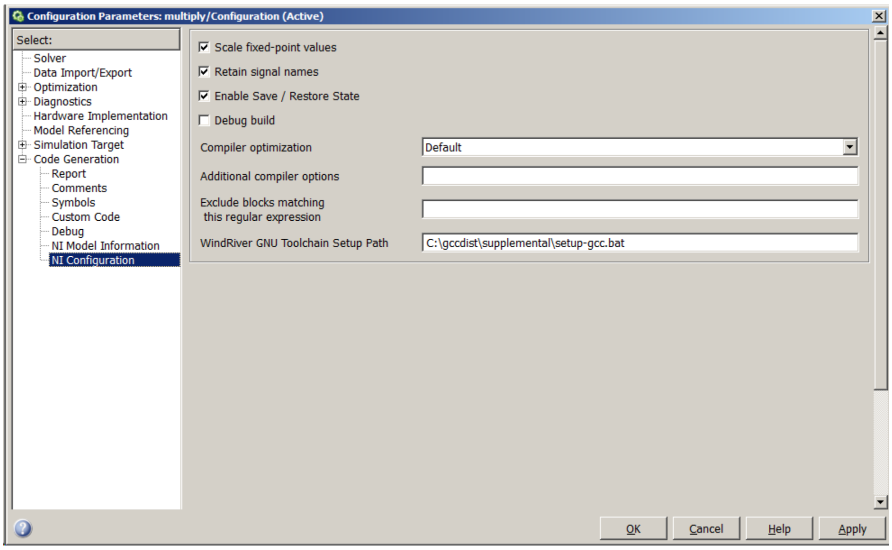

Figure 9: Simulink model configuration - NI configuration

   (b) Under "Solver options", "Type" should be set to "Fixed-step" and "Solver" to "discrete (no continuous states)"

3. Configure target, 8

4. Make sure that the WindRiver GNU Toolchain Setup Path is correct, Figure 9

7.3 Build

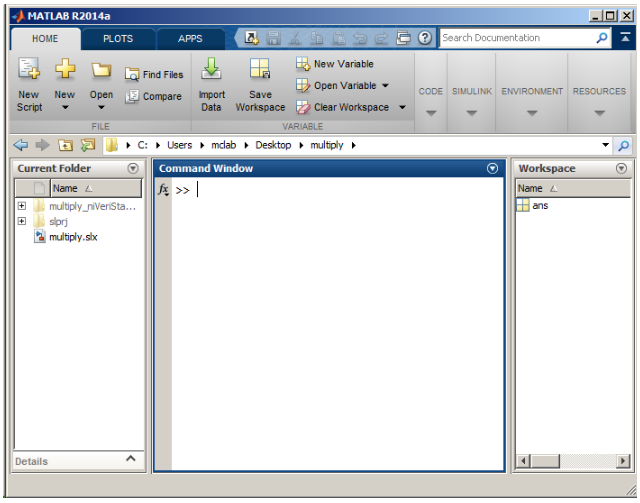

Figure 10: Matlab console

Creates a build folder, named NAME__niVeriStand_VxWorks_rtw, in the MATLAB Current Folder 10, need to change to the desired directory first

In Simulink: Code ¿ C/C++ Code ¿ Build model or Ctrl+B or button

Model file [model name].out

8 Model deployment

Models are deployed and interfaced through Veristand.

1. Start VeriStand 11

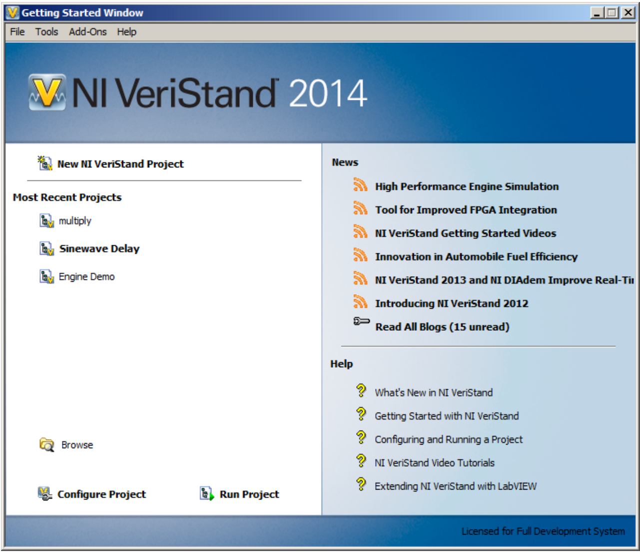

Figure 11: VeriStand

2. In the project explorer

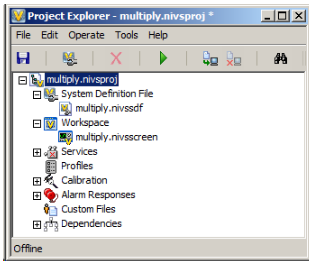

Figure 12: VeriStand Project Explorer

3. In System Explorer

Veristand osv

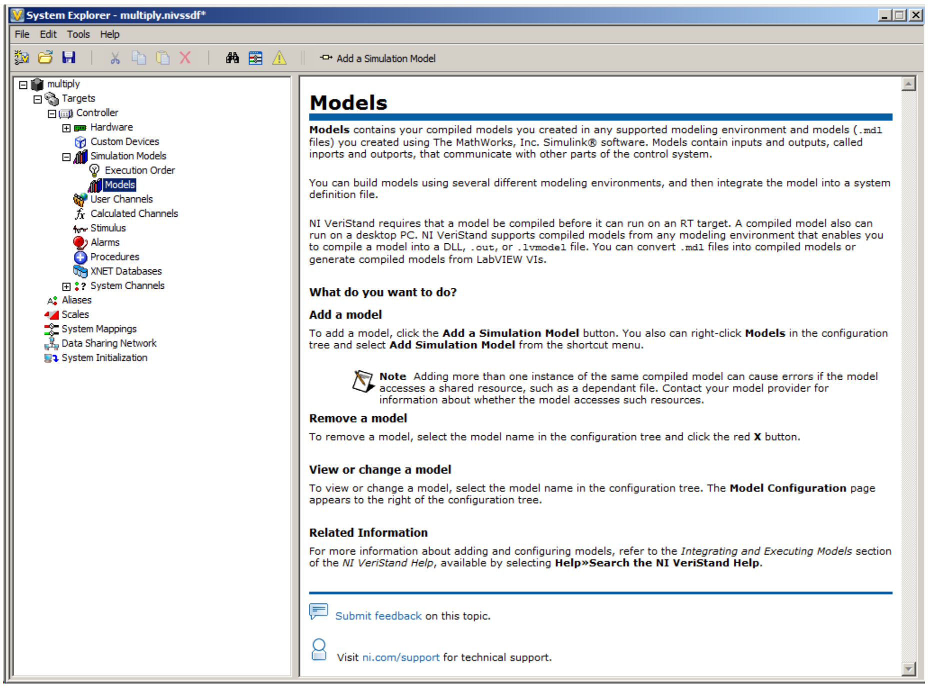

Figure 13: VeriStand - System Explorer

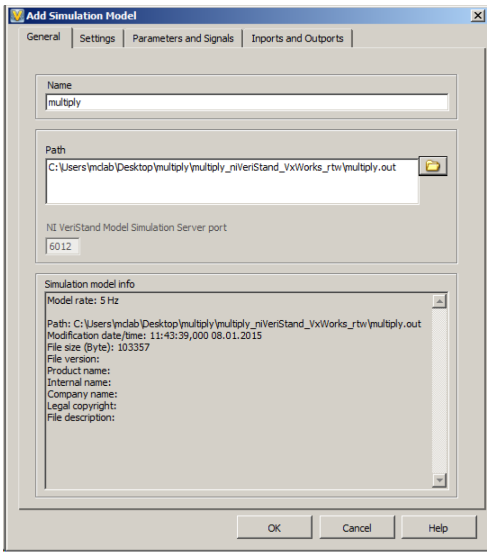

Figure 14: VeriStand - System Explorer Model

9 System interfacing

work space

10 Deploying basic VI

1. Open LabVIEW
2. Create project
3. Blank project
4. In the left pane tree, right-click Project: XXX, New -¿ Targets and devices
5. Keep setting Existing target or device, Discover an existing target(s) or device(s)
6. In the tree, expand Real-Time CompactRIO, wait for search and select the cRIO
7. In the Project Explorer, drag and drop the VI to the device
8. Right-click the VI, Run

11 Troubleshooting

typiske feil
batteri
nettverk

Part IV

Lab exercises and expected results

Forslag fra Roger

- Estimere pivot point
- Estimere bølgefrekvens og bølgehøyde??
- Tuning PID parametere
- Analysere agressivitet ved ulik tuning
- Analysere og teste notch effekt i bølgefilter
- Thrust allocation
- Lyapunov function. UGES $\rightarrow$ estimate convergence rate $\rightarrow$ simulate convergence rate $\rightarrow$ test convergence rate
- Bandwidth!

From TMR 4243 and 4515

- Robust nonlinear control methods (different backstepping methods, ISS designs, sliding-mode designs, passivity-based control, nonlinear PID and integral control).
- Adaptive control designs for nonlinear systems (adaptive backstepping, gradient methods, L1 adaptive control, etc.)
- Maneuvering control theory and path-following control designs for marine vessels (path parameterization, path generation, guidance theories, and feedback control laws).
- Fault-diagnosis and fault tolerant control (failure-mode detection, fault isolation, and control redesign to detect and handle failures in equipment and processes).

Part V
Equipment setup and configuration

A cRIO

A.1 Ethernet ports

A.1.1 Primary

Set fixed IP, set fixed IP on HIL-computers

A.1.2 Secondary ethernet port

Enabling the port

1. Start *NI MAX*
2. In the left pane tree, select the cRIO under *Remote Systems*
3. Open the *Network Settings* tab (located at the bottom of the window)
4. Set *Adapter Mode* to *TCP/IP Network*
5. Set *Configure IPv4 Address* to *Static*

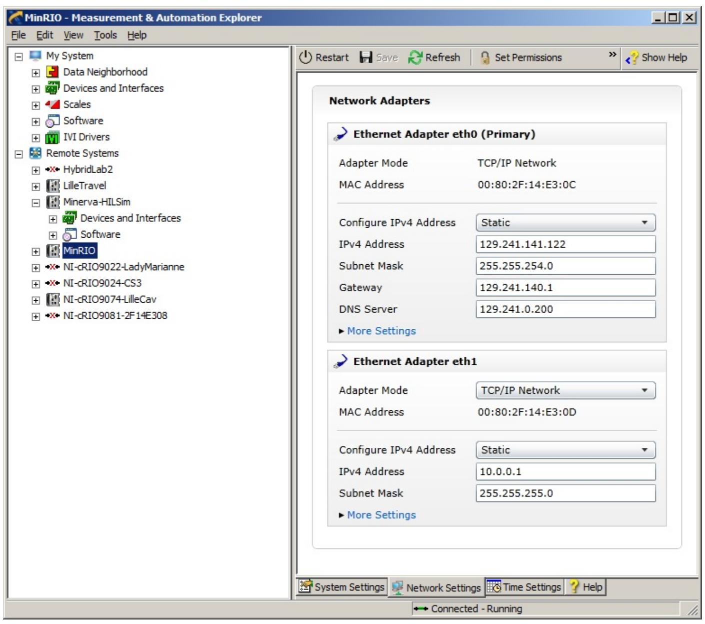

Figure 15: NI MAX - Network Settings

A.2 Install NI Veristand Engine

A.3 Installing custom device driver

In order to use a RPi to send joystick commands to the cRIO it is necessary to build a custom device driver. In our case Torgeir Wahl has built a driver, and this guide will show how to install the driver.

The first step is to copy the whole directory (folder named WL_Joystick) of the custom device driver into the correct directory on your computer:

C:\Users\Public\Documents\National Instruments\NI VeriStand 2014\Custom Devices

The directory should now contain something like Figure 16.

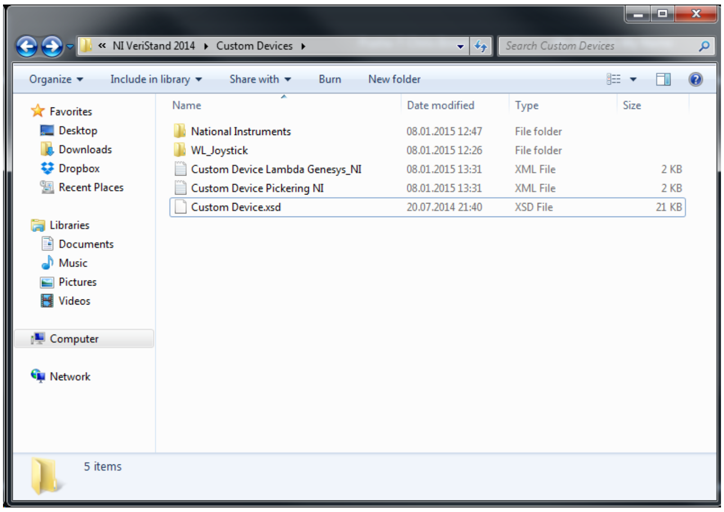

Figure 16: Custom device folder

The next step is to add custom device to your project. This is done in the system explorer, which is found as seen in Figure 17.

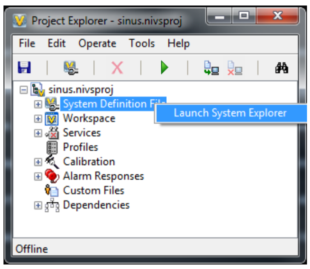

Figure 17: VeriStand launch system explorer

When in the system explorer, adding the custom device should be as simple as right clicking the custom device pane and choosing WL_Joystick, as in Figure 18. If you do not find the custom device WL_Joystick, the most likely problem is that the placement of the custom device folder from step 1 is wrong.

If the installation is successful you should be able to see WL_Joystick folder under custom devices as seen in the red box in Figure 19. Here you will also see the different inputs from the custom device, in this case it is joystick axis.

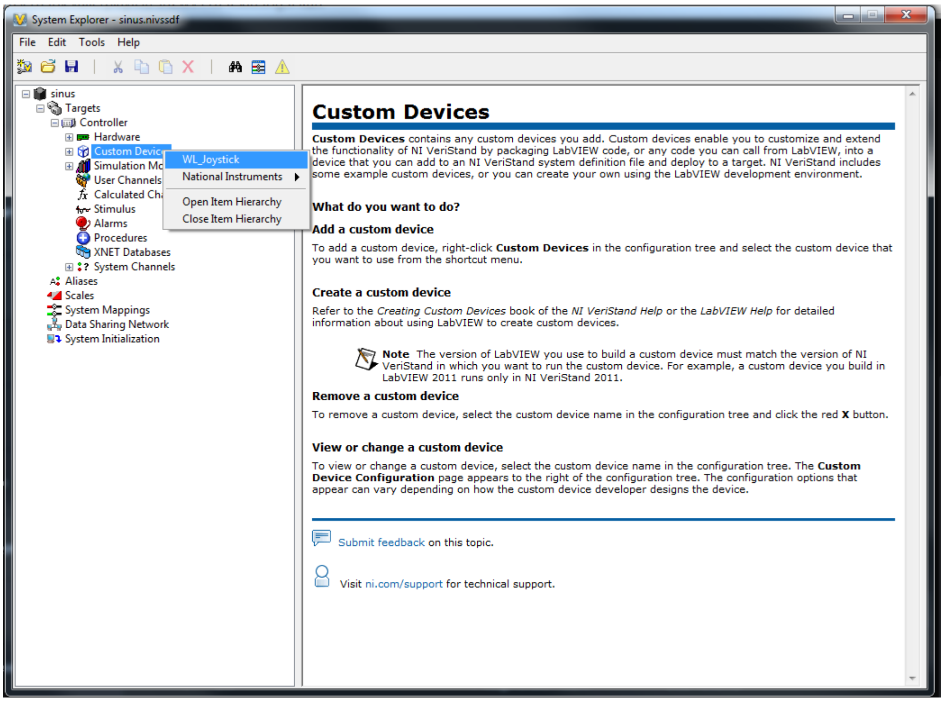

Figure 18: Custom device selection

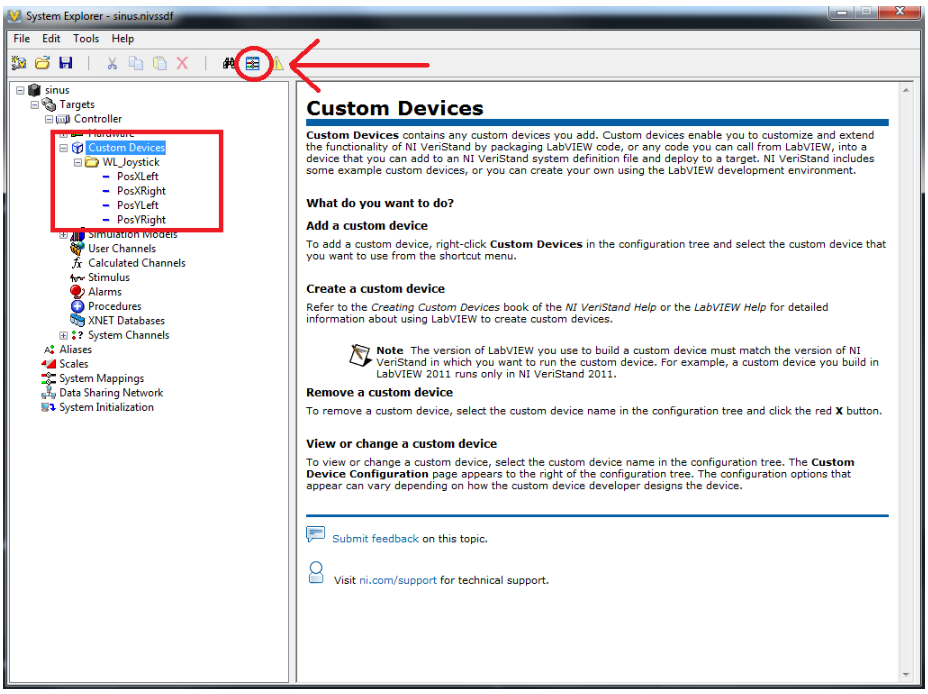

Figure 19: VeriStand

To connect the joystick to the input ports of the Simulink model. You open the system configuration mappings (click the button marked by the arrow in Figure 19).

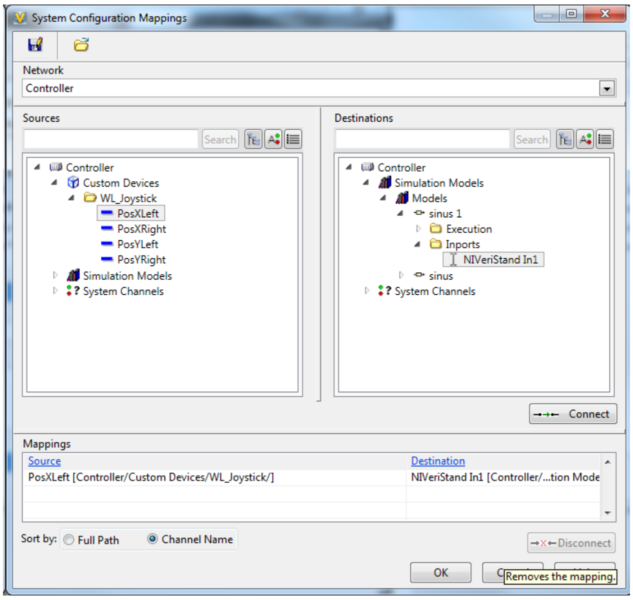

Figure 20: VeriStand System Configuration Mappings

You the simply find the ports you would like to connect, mark them and click the connect button. Figure 20 a joystick output is connected to a input port on the Simulink model.

A.4 Creating custom device driver

PWM, analog inn, analog ut

A.4.1 PWM output

VeriStand FPGA programming LabView -¿ Create project -¿ All -¿ NI VeriStand FPGA project -¿ Compact RIO -¿ Discover existing system -¿ Velge eget utstyr -¿ Vente på discovering -¿ I Project explorer *.vi (er bitfilen) *.fpgaconfig (egentlig XML) Endre på *.vi Fjerne overføldige pakker Oppdatere antall pakker i XML-filen og fjerne pakker som ikke er aktuelle, oppdatere tall på beholdte pakker. Kompiler

Kopier bit-file ut i samme mappe som *.fpgaconfig I System exlporer, FPGA -¿ Add FPGA target -¿ Finne *.fpgaconfig

A.4.2 Analog input

Eirik: FPGA-greier osv

A.5 Veristand FPGA programmering

Eirik: Skriv ordentlig om labview/XML delen av FPGA programmeringen

In order to access the analogue and digital I/O modules on our cRIO from Veristand, it is necessary to create a FPGA target in Labview with Labview and you will have to write a custom XML file.

A.5.1 Create Labview FPGA target and XML

The first step is

`https://decibel.ni.com/content/docs/DOC-13815`

A.5.2 Install in veristand

The Veristand software does not recognize the physical I/O components of the cRIO. It is necessarry to write a specific FPGA mapping for the specific setup. This results in a XML file that maps the ports.

To add this file to your Veristand project, enter the system explorer and find the FPGA pane under *targets\controller\hardware\chassis*, as seen in figure 21.

The next step is to find your XML file. In this case called cRIO-9113 Ex, it is very important that the XML file is placed on level above the FPGA bitfile folder in the directory system, as the files are really being used are the FPGA bitfiles.

The menu in should now look something like Figure 22, here you can see the analogue input signals and the digital output PWM signals. These can again be linked to other signals as seen in Figure20.

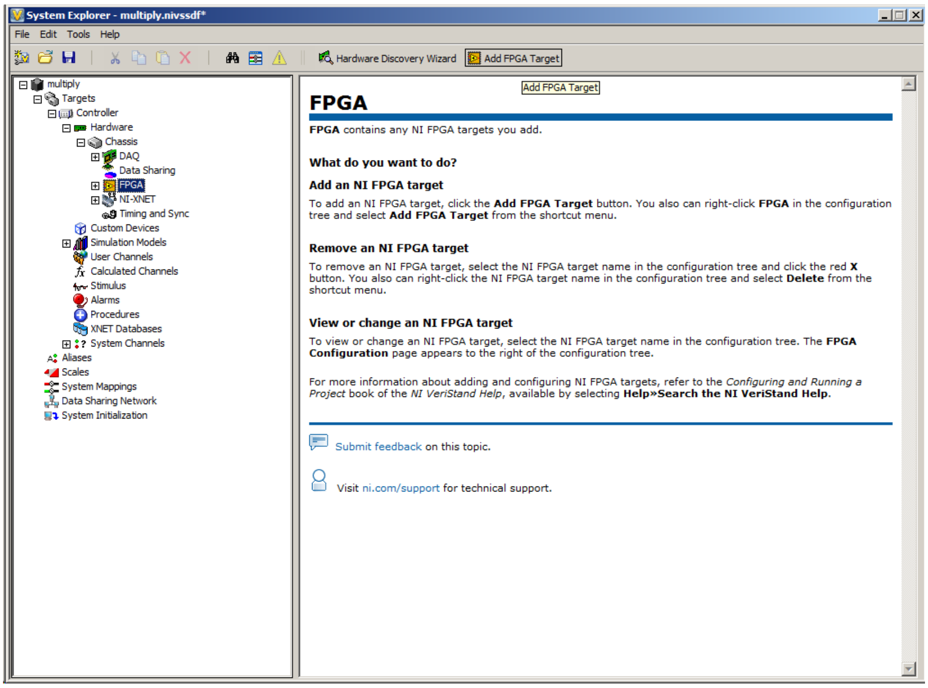

Figure 21: FPGA1

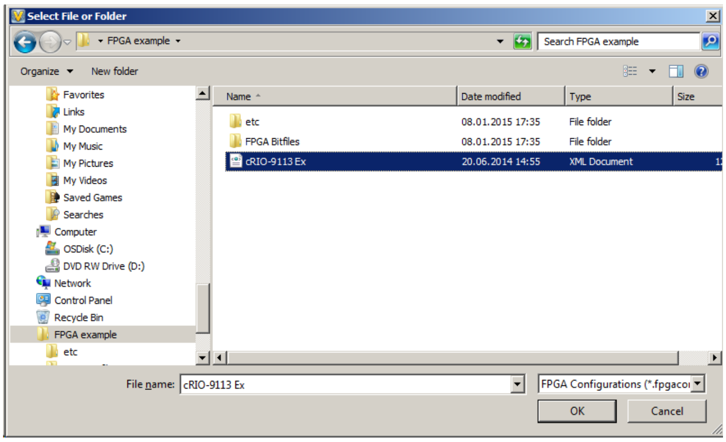

Figure 22: FPGA2

B Raspberry Pi

B.1 OS installation

Description using a Windows desktop, but also possible in Mac and Linux. See table B.1.

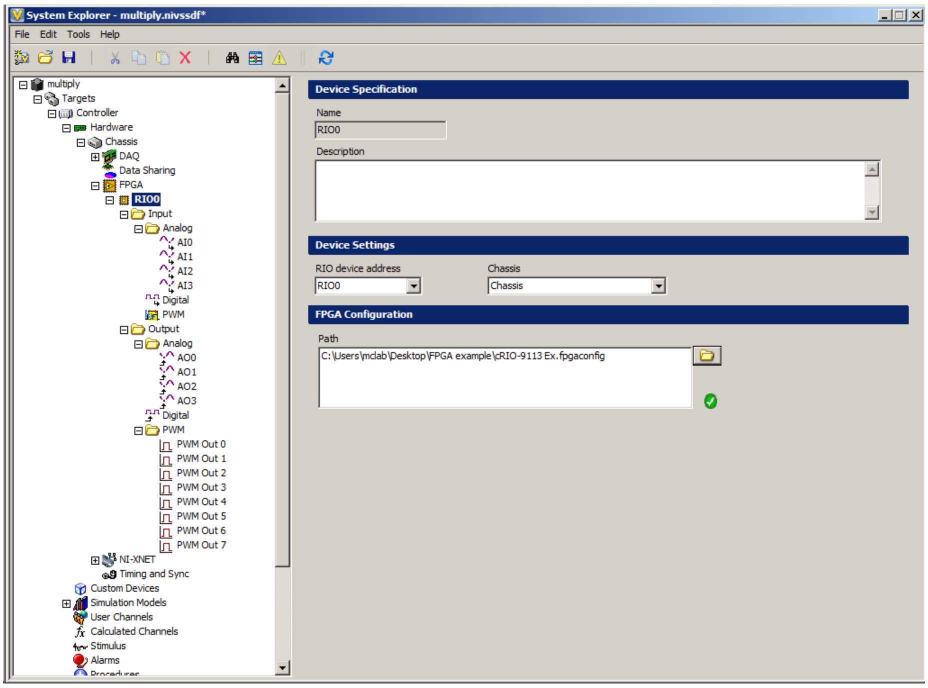

Figure 23: FPGA3

Windows	Linux, OSX
Win32 Disk Imager	dd
Advanced IP scanner	nmap
Putty	ssh

Table 2: RPi utilities

B.1.1 Download

- newest raspbian image for instance on http://www.raspberrypi.org/, currently 2014-09-09-wheezy-raspbian.img. Extract if zip.
- Win32 Disk Imager for instance http://sourceforge.net/projects/win32diskimager/
- Advanced IP scanner http://www.advanced-ip-scanner.com/.
- Putty http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

B.1.2 Write image to SD card

Run Win32 Disk Imager as administrator
Select the image file
Make sure you have the right drive
Push write
Figure 24.

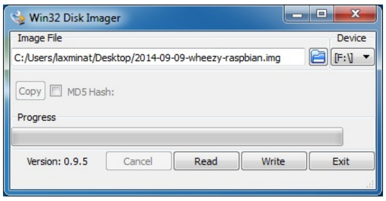

Figure 24: Disk Imager

B.1.3 Terminal access

Description using a Windows desktop, but also possible in Mac and Linux. Then image writer is XXX and putty is XXX.

Determining RPi IP address This step can be skipped if the IP is known.

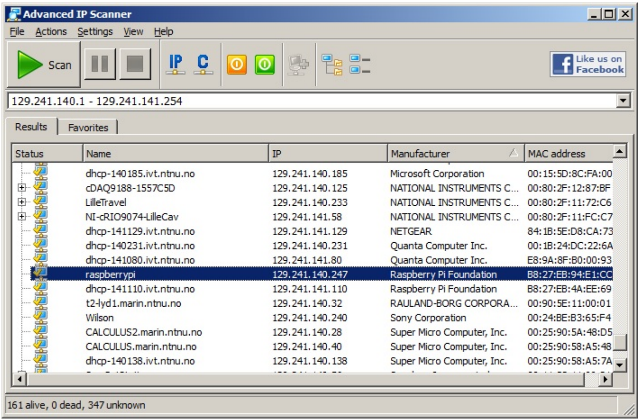

Figure 25: Advanced IP Scanner

By default the RPi has name *raspberrypi*. The IP is given in the next column, as in Figure 25.

Connect via SSH The RPi may be accessed through the network, i.e. without having to directly connect a monitor and keyboard.

Figure 26

Figure 27

Putty http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Default RPi are

```
Username: pi
Password: raspberry
```

However, the password is changed to the MC-lab standard.

B.1.4 Set fixed IP

```
sudo nano /etc/network/interfaces
```

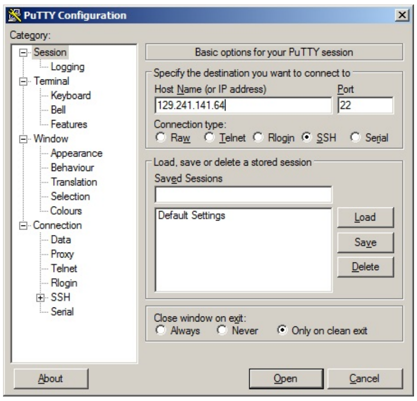

Figure 26: Putty settings

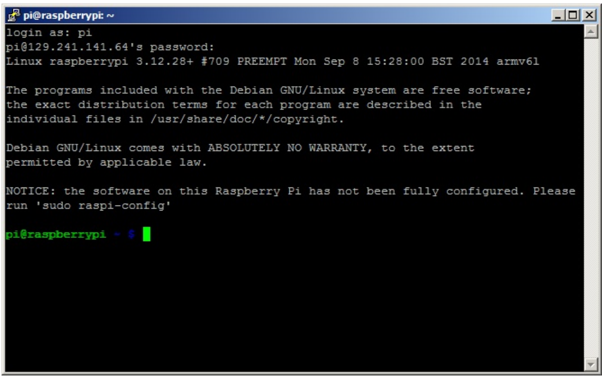

Figure 27: SSH connection

```
auto eth0
iface eth0 inet static
   address 192.168.1.22
   netmask 255.255.255.0
```

B.1.5 Finalize configuration

Figure 28
sudo raspi-config

Update configuration tool 8 Advanced Options
A9 Update

Expand filesystem 1 Expand Filesystem
Finish
Yes to reboot now
You will need to reconnect through SSH.

Change password 2 Change User Password

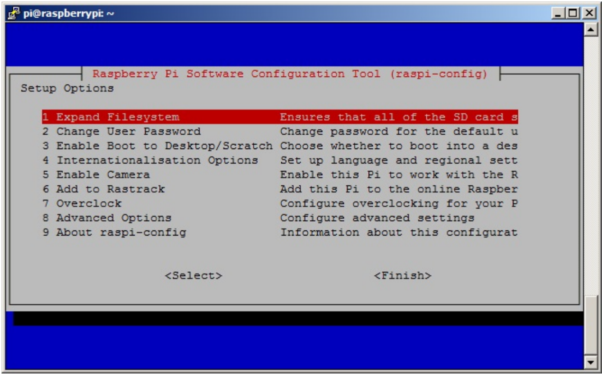

Figure 28: RPi configuration tool

Update and upgrade sudo apt-get update
sudo apt-get upgrade

B.2 PS3 Controller installation and configuration

Eirik: driver, python-Joystick osv

C Qualisys

calibration

Part VI
Misc

D Contributers and points of contact

Håkon Nødset Skåtun	Hakon.Nodset.Skatun@km.kongsberg.com
Øivind Kåre Kjerstad	bygde om skroget
Dinh Nam Tran	oppryddingsarbeid
Andreas Orsten	brukt mye, skrevet artikel om sleping av isberg
Robert kanajus	rkajanus@gmail.com brukt HIL-lab og Minerva
Torgeir Wahl	fulgt siden starten
Eirik Valle	
Andreas Reason Dahl	andreas.r.dahl@ntnu.no

Table 3: POCs

See Table D

E Suppliers

Laptops	Dell
cRIO	National Instruments
VSP	Thrusters were ordere at `www.cornwallmodelboats.co.uk/acatalog/voith_schottel.html`. Per 2014, availability is variable.

Table 4: Suppliers

F YouTube demonstration

`http://www.youtube.com/watch?v=MiESJsIZO04`

G To do list

- Etablere fargekoder for simulinkblokker (spesielt "ikke røre"-farge)
- Konsekvent notasjon: C/S Enterprise 1 eventuelt CSE1

- Forklaring av hva realtime betyr i HW og SW
- Troubleshooting-prosedyrer for de vanligste feilene
- Implementere "fail to zero" for når kommunikasjonen avbrytes.
- legge til IMU/gyro på båten

H Software needed

- Matlab
- Labview
- LabVIEW development system
- LabVIEW Real-Time Module
- LabVIEW FPGA Module (recommended)
- NI-RIO driver
- VeriStand