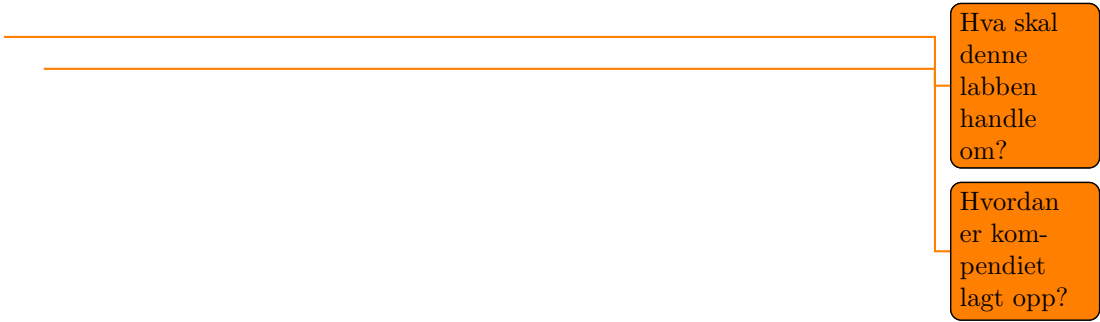


Contents

| | | |
|------------|--|-----------|
| 1 | Intro | 2 |
| I | Hardware | 3 |
| 2 | MC-Lab | 3 |
| 2.1 | Towing carriage | 3 |
| 2.2 | Wave generator | 3 |
| 2.2.1 | First order Stoke waves | 3 |
| 2.2.2 | Irregular waves | 3 |
| 2.3 | Qualisys positioning system | 3 |
| 2.4 | Ethernet communication network | 4 |
| 3 | CSE1 - C/S Enterprise I | 6 |
| 3.1 | Control system | 6 |
| 3.1.1 | Real time hardware | 6 |
| 3.1.2 | Real time software | 6 |
| 3.1.3 | Input/Output signals | 7 |
| 3.1.4 | NI Labview user interface | 7 |
| 3.1.5 | Startup procedure | 7 |
| 3.1.6 | Template: DP control system | 8 |
| 3.2 | HIL Simulation Setup | 8 |
| 3.2.1 | Hardware topology | 8 |
| 3.2.2 | Software C/S E1 | 8 |
| 3.2.3 | Startup procedure | 9 |
| 3.3 | Development procedure | 9 |
| 3.3.1 | Development for new algorithms in modules in Simulink . | 9 |
| 3.3.2 | Development with structural changes in I/O or Labview UI | 9 |
| 4 | Software | 9 |
| II | Theory | 10 |
| III | Lab exercises and expected results | 11 |
| 5 | Misc | 11 |
| 5.1 | Contributers and points of contact | 11 |
| 5.2 | VSP | 11 |
| 5.3 | YouTube demonstration | 12 |
| 5.4 | To do list | 12 |
| 5.5 | VeriStand FPGA programming | 12 |
| 5.6 | Software needed | 12 |

1 Intro



Hva skal
denne
labben
handle
om?

Hvordan
er kom-
pendiet
lagt opp?

Part I

Hardware

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 on-board the vessel runs the QNX real-time operating system while experimental results are presented in real-time on a host PC using Labview.

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

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

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

http://www.ntnu.edu/imt

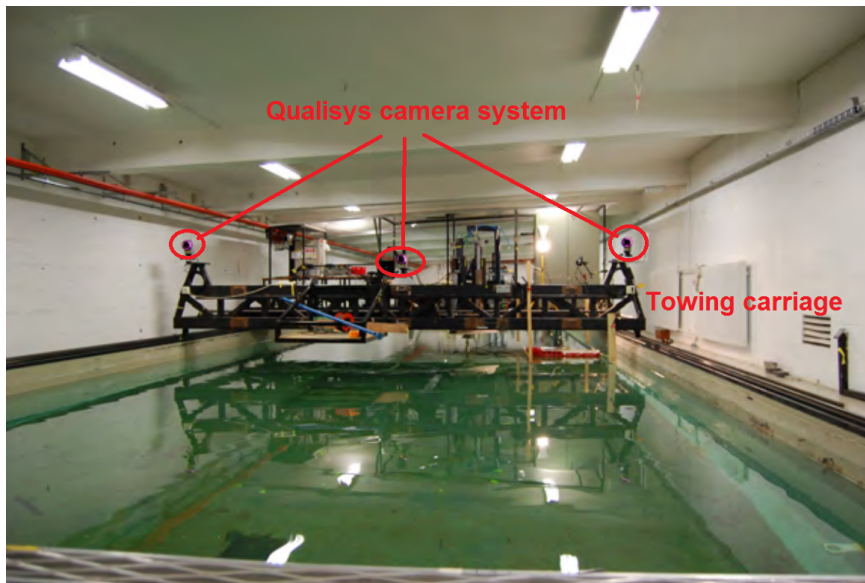


Figure 1: Towing carriage

- 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

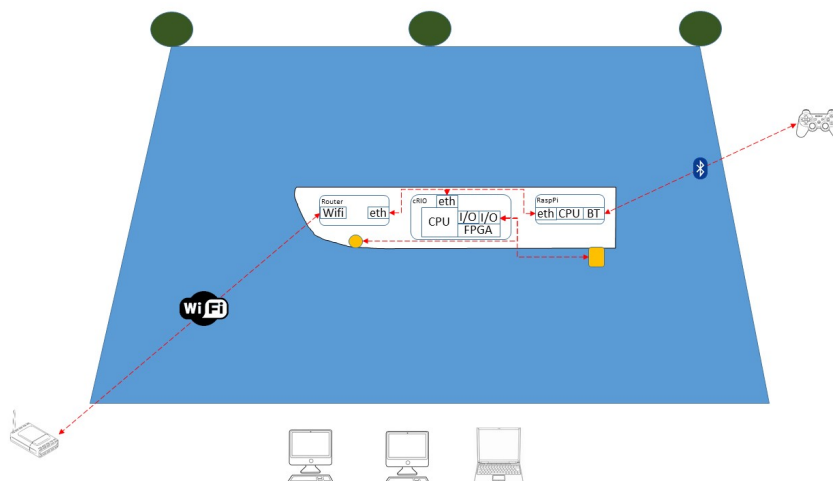


Figure 2: Towing carriage

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

3 CSE1 - C/S Enterprise I



Figure 3: C/S Enterprise I

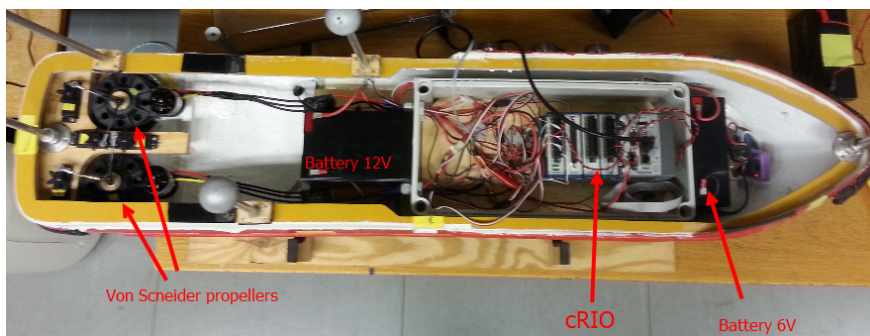


Figure 4: CSE1 - Hardware

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 visit the producer website

Sakset fra
ni.com/compactrio

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))

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

Nam sin
prosedyre
fra
readme.txt

```
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 important 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 response 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

Part II

Theory

Part III

Lab exercises and expected results

Forslag fra Roger

- Estimere pivot point
- Estimere bølgefrequens og bølgehøyde??
- Tuning PID parametere
- Analysere aggressivitet ved ulik tuning
- Analysere og teste notch effekt i bølgefilter
- Thrust allocation
- Lyapunov function. UGES → estimate convergence rate → simulate convergence rate → 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).

5 Misc

5.1 Contributors and points of contact

See Table 5.1

5.2 VSP

Thrusters were ordered at www.cornwallmodelboats.co.uk/acatalog/voith_schottel.html. Per 2014, availability is variable.

| | |
|-----------------------------|--------------------------------------|
| Hakon Nødset Skåtun | Hakon.Nodset.Skatun@km.kongsberg.com |
| Øivind Kåre Kjerstad | |
| Dinh Nam Tran | |
| Andreas Orsten | |
| R kanajus | |
| Torgeir Wahl | |
| Eirik Valle | |
| Andreas Reason Dahl | |

Table 1: POCs

5.3 YouTube demonstration

<http://www.youtube.com/watch?v=MIESJsIZ004>

5.4 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.

5.5 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 overfløydige pakker Oppdatere antall pakker i XML-filen og fjern pakker som ikke er aktuelle, oppdatere tall på beholdte pakker. Kompiler

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

5.6 Software needed

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