

Preliminary Design of the ITER Magnetic Diagnostic Integrators

André Gonçalves Torres

Thesis to obtain the Master of Science Degree in

Engineering Physics

Supervisors: Prof. Dr. Horácio João Matos Fernandes Dr. André Cabrita Neto

Examination Committee

Chairperson: Prof. Dr. João Pedro Saraiva Bizarro Supervisor: Prof. Dr. Horácio João Matos Fernandes Members of the Committee: Prof. Dr. Bernardo Brotas de Carvalho Prof. Dr. Pedro Manuel Brito da Silva Girão

November 2017

ii

Aos meus avós, Maria Rosa e Manuel António Gonçalves

Acknowledgments

First and foremost I would like to thank both my supervisors, for distinct reasons. To Professor Horácio Fernandes for the support, not only on this thesis but in the path that led to it and the opportunities made available to me. And to Dr. André Neto for the close support and guidance through this project, that allowed me to build a skill set that goes beyond what is patent on this dissertation. I also thank Llorenç Capellà, whose work and explanations were instrumental to the understanding of the magnetic diagnostic integrators.

Institutionally, I would like to acknowledge Fusion For Energy and the European Union for promoting and financing the opportunity for the traineeship, a milestone in my professional, academical and personal path. This acknowledgement is extended to the people on the CODAC group and in F4E from whom I got a lesson on professionalism, dedication and mission.

On a personal level, I would like to thank my closest family on not only the support but also the trust. The trust that allowed me to carry on with a less conventional path with full support and based only on my best judgment. And to my girlfriend, for putting up with me and this thesis.

Resumo

Esta dissertação explora o integrador para o diagnostico magnético do ITER, a cargo da *Fusion For Energy* (F4E) e a ser desenvolvido pelo Instituto de Plasmas e Fusão Nuclear (IPFN) e o *Culham Centre for Fusion Energy* (CCFE). Baseado no trabalho conduzido no grupo de *Control, Data Access and Communication* (CODAC) na F4E no âmbito de um estágio, esta dissertação foca-se em dois aspectos fundamentais do diagnóstico: a arquitectura da placa do integrador; e a distribuição dos dados adquiridos através da rede de tempo-real do ITER – *Synchronous Databus Network* (SDN). Os principais desafios técnicos e de projecto são apresentados, bem como as principais fases de desenvolvimento e a analise de dados que valida as escolhas de projecto para as placas.

Palavras-chave: ITER, Diagnostico Magnético, Integrador, Distribuição de Dados, Tempo-Real, Tokamaks

Abstract

This thesis reviews the magnetics integrator for the ITER magnetic diagnostics,to be delivered by Fusion For Energy (F4E) and being delivered by *Instituto de Plasmas e Fusão Nuclear* (IPFN) and the Culham Centre for Fusion Energy (CCFE). Based on the work conducted at the Control, Data Access and Communication (CODAC) group at F4E in the scope of a traineeship, this thesis focuses on two key aspects of the diagnostic: the integrator board architecture; and the real-time distribution of the acquired data using the ITER Synchronous Databus Network (SDN). The main technical and design challenges and development phases are presented, as well as the data analysis that validates the design choices for the integrator board.

Keywords: ITER, Magnetic Diagnostics, Integrator, Real-Time, Data Distribution, Tokamak

Contents

	Ackr	nowledgments	۷
	Res	sumo	vii
	Abst	stract	ix
	List	of Tables	xiii
	List	of Figures	xv
	List	of Abbreviations	xix
1	ITEF	R Magnetic Diagnostic and Data Distribution	1
	1.1	ITER	1
	1.2	Magnetic Diagnostics	2
		1.2.1 Physical Principle	2
		1.2.2 Measured Quantities and Applications	3
	1.3	Magnetic Diagnostic Integrator	4
		1.3.1 Current Development on Integrators	5
	1.4	Magnetic Diagnostic in the ITER Network Architecture	7
		1.4.1 Distributed Networks on Tokamaks	7
	1.5	Outline	8
2	Mag	gnetic Diagnostic Integrator Design Description	11
	2.1	A Bespoke Integrator for ITER	11
	2.2	Square Modulation Technique	13
		2.2.1 Modulation Method Shortcomings	13
		2.2.2 Design Challenges	13
	2.3	High Frequency Signal Recovery	16
	2.4	Magnetic Diagnostic Subsystems Description	16
		2.4.1 Transmission Lines and Port Cell Resistors	16
		2.4.2 Digital Integrator Boards – Analogue Signal Conditioning	17
	2.5	Differences Between Modules	21
	2.6	Prototype Boards Testing	25
		2.6.1 Real-time Integration Software	26
		2.6.2 Test Plan	27

		2.6.3	Chopping Frequency Dependence Tests	29	
3	Rea	l-Time	Network Performance	37	
	3.1	Purpo	se of the Experiment	37	
	3.2	Hardw	vare and Test Rig	38	
		3.2.1	ODROIDs	38	
		3.2.2	ITER Fast Controller	39	
		3.2.3	Physical Installation	40	
	3.3	Test N	lethodology and Execution	40	
		3.3.1	Test Plan	42	
		3.3.2	Preparation of the Computers for the Experiment	46	
	3.4	Test Ir	nplementation Tools	48	
		3.4.1	SDN Software	48	
		3.4.2	Test Programs	50	
	3.5	Test R	esults and Analysis	57	
		3.5.1	Test 0 – Control	58	
		3.5.2	Test 1 – No Delay	60	
		3.5.3	Test 2 – miniCODAC Over Concentrated Traffic	63	
		3.5.4	Test 3 – miniCODAC Over Distributed Traffic	65	
	3.6	Conclu	usions and Possible Improvements to the Test Setup	68	
4	Con	clusio	าร	73	
Bi	Bibliography 75				
A	Driv	er Con	npilation Instructions	79	

List of Tables

2.1	Requirements (nominal and features) for the magnetic integrator boards	12
2.2	Summary of the comparison between the WO and EO relative to its source, insertion	
	point, behavior. While the WO has a smaller magnitude, it is not removed by the modula-	
	tion and demodulation	15
2.3	Identification under the ITER Plant Breakdown Structure (PBS) of a selection of coils with	
	their location.	17
2.4	Identification and classification of the eight integrator designs prototyped. The designs	
	were developed under two consecutive development phases and indexed from 1 to 8. To	
	facilitate communication during the development, each design was given an alias	18
2.5	Input stage parameters: static gain achieved as an attenuation of the signal by the resistor	
	divider, and cut-off frequency set by the passive RC filter.	19
2.6	Summary of the differences between all prototyped modules of all designs. Components	
	described in the text, SAR ADC has 18 bits and $\Delta\Sigma$ 23	24
2.7	Description of the tests done to the magnetic integrators in IPFN, CCFE and F4E	28
2.8	Round one of testing, with 10 short circuit pulses with half hour duration. Mean (μ) and	
	standard deviation () of the absolute drift presented with all results expressed in $\mu V\cdot s.$.	30
2.9	Pulses at (CP 1024, 16 kHz) with and without linear interpolation of 50 samples. Mean	
	(μ) and standard deviation (σ) of the absolute drift for 10 short circuit pulses with half hour	
	duration. All results expressed in $\mu V \cdot s$.	32
2.10	Summary of the module differences regarding Chopper and Snubbers. Identification of	
	the chopper models coherent with Table 2.6.	33
2.11	Round two of testing, with 100 short circuit pulses with half hour duration. Mean (μ) and	
	standard deviation (σ) of the absolute drift presented with all results expressed in $\mu V \cdot s.$.	33
3.1	Selection of specifications for the ODROID C2 used in the experiment. Hardware specifi-	
	cations according to the manufacturer [23].	39
3.2	Selection of specifications for the ITER supplied fast controller computers (miniCODAC1	
	and miniCODAC2) used in the experiment.	40
3.3	Configurations for the control test (test 0). Tests 0-1 to 0-4 have one million iterations at	
	an increasing publishing frequency, test 0-5 repeats the 1 kHz test with ten times more	
	iterations.	42

3.4	Configurations for the no delay test (test 1). Tests 1-1 sets a reference with the miniCO-	
	DAC and the 15 ODROIDs replying with no added delay and a regular payload size. Test	
	1-2 doubles the publishing frequency, test 1-3increases the payload size and test 1-4 has	
	fewer slaves.	43
3.5	Configurations for the test with concentrated traffic (test 2). The delay value is the sleep	
	time on that machine between the reception of the package from the master and sending	
	the reply. These tests aim at timing the miniCODAC reply to occur simultaneously, before	
	and after the bulk of the ODROID traffic window.	44
3.6	Configurations for the test with distributed traffic (test 3). The delay value is the sleep time	
	on that machine between the reception of the package from the master and sending the	
	reply. n stands for the index of the ODROID ($n \in [1, 15]$). Test 3-2 acts as a control	45
3.7	Description of the terminal arguments for sdn-ping.	51
3.8	Description of the terminal arguments for sdn-pong.	54
3.9	Description of the bash scripts created to automate the ODROIDs usage.	55
3.10	Description of the supporting programs for the experimental data analysis	56
3.11	Test 0 statistical results for the round-trip measurements. Tests 0-1 trough 0-4 have a	
	increasing publishing frequency, while test 0-5 is a repetition of the test at $1 \ kHz$ (0-3)	
	with 10 times more statistical elements.	58
3.12	Test 1 statistical results for the round-trip measurements. Results shown for the miniCO-	
	DAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15).	
	Tests 1-1 is the control test with 15 ODROIDs, $100 \ Hz$ publishing frequency, and small	
	payload size. Test 1-2 has a $200 \ Hz$ publishing frequency, test 1-3 a larger payload size	
	and test 1-4 only 9 ODROIDs online	60
3.13	Test 2 statistical results for the round-trip measurements. Results shown for the miniCO-	
	DAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15).	
	Tests 2-1, 2-2 and 2-3 have the miniCODAC response delayed by $420,\ 350$ and $520\ \mu s$	
	respectively	63
3.14	Comparison of the latencies obtained in test 2 with the expected. The expected delay	
	is computed as the sum of the mean round trip delay for the test 1-1 (123.8 μs) with the	
	intentionally added delay (420, 350 and 520 μs respectively)	63
3.15	Test 3 statistical results for the round-trip measurements. Results shown for the mini-	
	CODAC (replier 0) and the average of the results for the individual ODROIDs (repliers	
	1-15)	65
3.16	Test 4 statistical results for the round-trip measurements. Tests 4-1 trough 4-3 are repe-	
	titions of tests 1-1, 2-1, and 3-1 respectively with a different network switch and 500000	
	packets instead of one million. Results shown for the miniCODAC (replier 0) and the	
	average of the results for the individual ODROIDs (repliers 1-15)	68

List of Figures

1.1	Progress of the fusion "triple product" of plasma ion density, ion temperature and energy	
	confinement time. Copyright © EUROfusion 2014 - 2018	3
1.2	Illustration of the most common magnetic coils used in tokamak magnetic diagnostics [4].	4
1.3	Examples of different integrator layouts. a) analog, b) hybrid, c) digital [5]	4
1.4	Simplified schematic representation of the differential structure of the Tore Supra's mag-	
	netics integrator [6].	5
1.5	Simplified schematic representation of the hybrid integrator for long pulses [11]	6
1.6	Schematic representation of the analog part of the digital chopper integrator for W7-X [12].	7
1.7	Magnetic diagnostic in the ITER network architecture. Highlighted in yellow, the scope of	
	this thesis. Adapted from [1].	8
2.1	Scope of this chapter on the magnetic diagnostic.	11
2.2	Simplified scheme of the magnetic diagnostic integrator board with the two noise types	
	and their insertion points on the system.	14
2.3	Demonstration of the effect of the EO and WO after modulation. Left: on time domain,	
	right: on frequency domain (as function of the frequency, F_c denotes the chopping fre-	
	quency). One can observe that the WO is modulated, with the signal, while the EO is	
	added after the chopping and appears as a DC offset.	14
2.4	Demonstration of the effect of the EO and WO after demodulation. Left: on time domain,	
	right: on frequency domain (as function of the frequency, F_c denotes the chopping fre-	
	quency). One can observe that the WO is demodulated, with the signal, while the EO is	
	appears now modulated	15
2.5	Demonstration of the effect of the EO and WO after integration on time domain. One can	
	observe that the EO is integrated as perturbations around 0 while the WO is appears now	
	as a drift.	15
2.6	Block diagram of the two channels showing on top the modulating path, developed in this	
	thesis and on the bottom the high frequency path.	16
2.7	Schematic representation of the electronic stages of the analog path for the three config-	
	urations. Each configuration labeled with the designs that use it.	18
2.8	Left: Topology of the input stage, showing a Thévenin voltage divider and a passive low-	
	pass filter in a symmetrical architecture. Right: Frequency response of the filter	19

2.9	Representation of the chopped signal in the frequency domain. The original signal is reproduced every odd multiple of the chopping frequency.	20
2.10	Left: Topology of the Multi-Feedback anti-aliasing filter in a symmetrical architecture. Right: Frequency response of the filter.	21
2.11	Left: Topology of the Sallen-Key anti-aliasing filter, in a symmetrical architecture. Right: Fre- quency response of the filter.	23
2.12	Example of short circuit test performed in the premises of F4E for two boards with the design D4 Rome : D4M3 in blue and D4M4 in red. Two phenomena are visible: the maximum drift was not obtained in the end of the experiment for the signal in blue and an inversion of the drift.	25
2.13	Mean values for the first round of the testing, with 10 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.8.	31
2.14	Standard deviation values for the first round of the testing, with 10 short circuit pulses with half hour duration.	31
2.15	Mean values for the first round of the testing, with 10 short circuit pulses with half hour duration. Same as the plot in Figure 2.13 without the outlier D1M4 . Absolute drift versus chopping frequency, as in Table 2.8.	32
2.16	Mean values for the second round of the testing, with 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.	33
2.17	Mean values for the modules with the chopper A. 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.	34
2.18	Mean values for the modules with the chopper C (D4 Rome design). 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.	35
3.1	ODROID-C2 ARM 64 bit 1.5 GHz quad core single board computer used for the experi- ment. Source: [23]	39
3.2	Structure with the ODROIDs as part of the test rig. In this picture one can identify the PC case, the 4 independent power supplies and wiring, the 3 stacks of 5 ODROID single board computers each and the CAT 6 Ethernet cables going out of the case. The pieces of paper visible on the photo identify the ODROIDs by their IP addresses.	41
3.3	Illustration of the basic principle behind the experiment. Left: In the simulated scenario, the master represents the ECPC and the slaves gyrothrons or other auxiliaries of the ECRH system. The master transmits to every slave a message, using UDP multicast. This message is structured as a SDN topic, of which the master is the publisher and all the slaves subscribe to. Right: in the test rig, the master is miniCODAC1 and the slaves are miniCODAC2 and the array of ODROIDs. Having received the message the slaves	
	process and reply directly to the master using UDP unicast.	41

- 3.4 Illustration of the basic principle behind test 2. Left: The master transmits to every slave a message, using UDP multicast, structured as a SDN topic. Center: the miniCODAC slave performs a busy sleep for a set time, while the ODROIDs naturally take more time processing and transmuting. Right: all slaves reply directly to the master using UDP unicast. 44
- 3.5 Illustration of the basic principle behind test 3. Left: The master transmits to every slave a message, using UDP multicast, structured as a SDN topic. Center: the miniCODAC slave performs a busy sleep for a set time, while each ODROIDs waits for incrementally longer time. Right: all slaves reply to the master using UDP unicast.
- 3.7 Round-trip delays in test 0-1 (100 Hz) over time, smoothed by a moving average to improve readability. Figure 3.6 shows (in red) the equivalent histogram. The measured round-tip values distribute themselves around two distinguishable values: around 118 and 123 μs, also visible on the histograms.
- 3.9 Histogram of the round-trip delay values of all the slaves for two different publishing frequencies: 100 Hz (test 1-1, red) and 200 Hz (test 1-2, green). The traffic coming from the miniCODAC and from the ODROIDs appear in different bands, being the first distinguishable from the rest. A slight dependence of the round-trip delay with the publishing frequency is visible.
- 3.10 Histogram of the round-trip delay values of all the slaves for two different publishing payload sizes: 200 B (test 1-1, red) and 1192 B (test 1-3, green). Both tests have a publishing frequency of 100Hz. The traffic coming from the miniCODAC and from the ODROIDs appear in different bands, being the first distinguishable from the rest. A clear influence of the payload size on the delay values is observed.

3.12	Round-trip delays in test 2-1 of selected slaves over time, smoothed by a moving average	
	to improve readability. In red the miniCODAC response, in green, blue and magenta, the	
	replies of ODROIDs 5, 10 and 15, respectively. Figure 3.11 shows (in red) the equivalent	
	histogram	64
3.13	Histogram of the round-trip delay values of all the slaves with the miniCODAC replies in	
	red and the rest of the slaves in green (test 3-1). The miniCODAC is delayed by $420~\mu s$	
	while the ODROID index n has a delay given by $10 \ \mu s \times n$.	66
3.14	Histogram of the round-trip delay values for the miniCODAC traffic only. Comparison of	
	the performance with concentrated ODROID traffic, test 2-1, in purple, achieved by a delay	
	of $420 \ \mu s$ on the miniCODAC; with distributed ODROID traffic, test 3-1 in green, achieved	
	by a delay of $420 \ \mu s$ on the miniCODAC and $10 \ \mu s \times n$ delay on the n^{th} ODROID; and in	
	a control test with no ODROID traffic and a delay of the same $420 \ \mu s$ on the miniCODAC,	
	test 3-2 in cyan	66
3.15	Round-trip delays in test 3-1 of selected slaves over time, smoothed by a moving average	
	to improve readability. In red the miniCODAC response, in green, blue and magenta,	
	the replies of ODROIDs 5, 10 and 15, respectively. Figure 3.13 shows the equivalent	
	histograms	67
3.16	Histogram of the round-trip delay values of all the slaves with no imposed delays (test	
	4-1). In red the miniCODAC replies and in blue the ODROID replies	69
3.17	Histogram of the round-trip delay values of all the slaves. Imposed delay of $520~\mu s$ on the	
	miniCODAC (test 4-2). In red the miniCODAC replies and in blue the ODROID replies	69
3.18	Histogram of the round-trip delay values of all the slaves. Imposed delay of $520~\mu s$ on the	
	miniCODAC and $10 \times n \ \mu s$ on the ODROID n (test 4-3). In red the miniCODAC replies	
	and in blue the ODROID replies.	70
3.19	Illustration of the basic principle behind test the GIPO proposed test. Left: The master	
	transmits to every slave a message, using UDP multicast, structured as a SDN topic.	
	Center: the miniCODAC and a 'master' ODROID perform a busy sleep for a set time,	
	while the remain ODROIDs wait for their GPIO input to toggle state. Right: the ODROID	
	master' toggles the state of the line, sending a go signal for all ODOIDS to reply to the	
	master using UDP unicast.	70

List of Abbreviations

ADC Analog to Digital Converter. BIOS Basic Input/Output System. **BJT** Bipolar Junction Transistors. CCS CODAC Core System. **CODAC** Control Data Access and Communication. **CP** Chopper Parameter. CPU Central Processing Unit. **DA** Domestic Agencies. **DAN** Data Archiving Network. **ECPC** Electron Cyclotron Plant Control. ECRH Electron Cyclotron Resonance Heating. **ENOB** Effective Number of Bits. **EO** Electronics Offset. F4E Fusion For Energy. FET Field Effect Transistors. FPGA Field Programmable Gate Array. GPIO General Purpose Input and Output. HF High Frequency. HIF High Impedance Follower. HRT High Resolution Timer.

IO ITER Organization.

- IP Internet Protocol.
- LSB Least Significant Bit.
- MARTe Multithread Application Real-Time executor.
- MFT Multi-Feedback Topology.
- MHD Magneto Hydro Dynamics.
- **MRG** Messaging Real-Time Grid.
- **MSPS** Mega Samples Per Second.
- **OpAmp** Operational Amplifier.
- PBS Plant Breakdown Structure.
- PCS Plasma Control System.
- **PON** Plant Operation Network.
- **PS I&C** Plant System Instrumentation and Control.
- **PSC** Plant System Controller.
- **R&D** Research and Development.
- SAR Successive Approximation Register.
- **SDN** Synchronous Databus Network.
- TCN Time Communication Network.
- **TSC** Time Stamp Collector.
- WO wiring Offset.

Chapter 1

ITER Magnetic Diagnostic and Data Distribution

Since the first tokamaks, the magnetic diagnostics have proven to be a reliable and important part of the data acquisition and control systems in fusion research and engineering. For once, it is completely passive, adding no energy to the plasma (i.e., electro-magnetic wave or particle insertion) in order to operate. This makes the diagnostic more robust, as it has a small interaction with the plasma.

Requiring little processing to compute the sensor physical value and having no explicit dependency on other diagnostics, the majority of the magnetics diagnostic measurements have real-time application in control, namely plasma shape and position, encompassing a large amount of coils of different geometries and locations (~1600 in 23 different groups for ITER [1]) and measuring different quantities. Furthermore the magnetics is also a complementary and/or auxiliary diagnostic to others.

Imposing a single network protocol to manage the interchange of real-time data between plant systems on a large scale project can be beneficial for the design, integration and commissioning of such systems. The ITER Synchronous Databus Network (SDN) will distribute the diagnostic measurements and handle data transmission for all the systems that contribute to the real-time control of the ITER. It is then paramount to guarantee that the SDN interface is indeed adequate to be also used inside the diagnostic, i.e. that it meets the low-latency (< $100 \ \mu s$) and robustness requirements (no loss of packets during the experiment).

This thesis provides an investigation into the innovative ITER magnetic diagnostic integrator and report on results of development phase tests. The Real-Time performance of the ITER SDN is also tested using an innovative test bed.

1.1 ITER

ITER, originally standing for International Thermonuclear Experimental Reactor, is a tokamak currently under construction, which will be the world's largest tokamak. Located in Cadarache, France, this magnetic confinement fusion device aims at proving the feasibility of fusion power as a viable energy solution.

Being the largest tokamak ever devised, ITER is the result of a collaboration between 35 nations. The project is managed by the ITER Organization (IO), however IO is not responsible for the development of the components themselves. For that purpose, Domestic Agencies (DA) were created for each of the participant parties (China, the European Union, India, Japan, Korea, Russia and the United States). The DAs are responsible for the development and delivery of the components on-site. Fusion for Energy (F4E) is the European DA, responsible for the preparation and coordination of the design, research and development (R&D) and fabrication of most of the high-technology components on ITER.

Fusion as a source of energy needs no proof: the Sun's irradiated energy comes from fusion reactions. On earth though, the pressures resulting from the Sun's massiveness can not be achieved, hence gravitational confinement is out of our reach.

Fusion is dependent on three physical quantities: density n_i , confinement time τ_E and temperature T_i . The product of this quantities, the so called "triple product", or Lawson's criterion, gives a theoretical threshold for fusion. Figure 1.1 shows a mapping of milestone results of several facilities regarding the fusion triple product. In the figure is also displayed the predictable ITER placement, right in the threshold for ignition, at $n\tau_E T_i \ge 5.10 \cdot 10^{21} m^{-3} s \, keV$. Reaching this value, the fusion reaction is self-sustained, the "burning plasma", where the heating of the plasma by the nuclei is greater than the heat loss. Note that this condition is different from the "breakeven", the point where the fusion energy equals the external heating. In a deuterium-tritium (D-T) plasma, the most efficient and the one used in ITER, most of the energy is carried in the neutrons, not contributing directly to the ignition condition. Denoting Q as the ratio of the fusion energy to that used to heat the plasma, breakeven sits at Q = 1 while in a fully ignited burning plasma $Q \to \infty$, meaning there is no need for external heating. While a breakeven was never achieved, JET stands as the record holder, with a ratio of 0.62 in 1997 [2]. ITER aims at $Q \ge 10$ with an inductive burn duration not shorter than 300s [3].

1.2 Magnetic Diagnostics

The magnetics diagnostic provides passive, real-time, integral measurements on different plasma parameters, depending on the type and positioning of the magnetic coils.

1.2.1 Physical Principle

This diagnostic uses a signal generated by a simple and well known physical phenomenon: electromagnetic induction – The magnetic field crossing the coil's section generates a signal proportional to the flux rate of change:

$$V = -\frac{d\phi}{dt}.$$
(1.1)

This means, however, that in order to get the nominal value of the field/flux from the induced electromotive force ($\epsilon(t)$) an integration has to take place:

$$\phi(t_1) - \phi(t_0) = -\int_{t_0}^{t_1} \epsilon(t) \cdot dt.$$
(1.2)



Figure 1.1: Progress of the fusion "triple product" of plasma ion density, ion temperature and energy confinement time. Copyright © EUROfusion 2014 - 2018.

Considering an uniform field normal to an equivalent coil surface S of n turns \times area, the inducing field can be recovered as

$$B(t) = \frac{\phi(t)}{S}.$$
(1.3)

1.2.2 Measured Quantities and Applications

The described principle can be employed to a vast range of coils, with different designs and placements, allowing the measurement of different plasma parameters. Three of the most common and widespread pick-up coils used in tokamaks are illustrated in Figure 1.2. In addition, the voltage loop (toroidal loop, parallel to the plasma) is also present in most tokamaks, as it can provide useful diagnostics of the plasma resistance and the Ohmic heating. By looping around itself and returning along its axis, the Rogowsky coils encircle the plasma without enclosing any net flux parallel to the current, allowing a measurement of the plasma current. The coils can be placed externally (to measure the toroidal field outside the plasma, for instance), along a surface, such as the vacuum vessel, or at very localized regions of interest, such as the divertor.

Besides the data for physics analysis, this diagnostics provide real time data with minimal processing. Therefore, it is crucial for the plant control as well as interlock/safety control. Plasma shape and position, MHD and other plasma modes, current and magnetic fields are derived from either the raw or integrated flux data and fed to the relevant plant systems.

The fact that the diagnostic data is used for control brings a whole set of requirements and engineering challenges for the data acquisition system – low and constant latency, minimal uncertainty and performance reliability.



Figure 1.2: Illustration of the most common magnetic coils used in tokamak magnetic diagnostics [4].

1.3 Magnetic Diagnostic Integrator

The integration can be performed by three different strategies: analog, hybrid or digital. The analog integration is, by far, the most widely used. A schematic representation of each architecture is shown in Figure 1.3.



Figure 1.3: Examples of different integrator layouts. a) analog, b) hybrid, c) digital [5].

Analog integration relies on robust electronics, with little to no processing needed as it outputs has a continuous signal, proportional to the flux, without latency in the low-frequency range.

The vast majority of magnetic confinement fusion devices are tokamaks, with pulse durations ranging from milliseconds to a few seconds in the larger machines. Therefore, high frequency acquisitions are prioritized, and for that, the no latency analog integrator is a good solution. For ITER, though, not only the amplitude of the fields measured is higher, the acquisitions can last up to one hour on the first phase [3]. Because of that, the most important aspect to consider developing the integrator is the minimization of the drift. With analog integrators, while there are some designs that try to mitigate this effect, the problem always comes down to the electronic components' sensibility. In particular, the thermo-electric

voltages in the input stage of the operational amplifiers are a source of time-variant drift that cannot be compensated.

On the other hand, a digital integration, even though not so accurate for high frequency transients has two key benefits when dealing with long integrations and slow varying signals: (i) it has a higher dynamic range, determined by the input saturation level (and not the output as in the analog case), signal characteristic times and sampling rate; (ii) since it relies on a processing element, a number of strategies can be employed, relying on digital feedback to the electronics (clock signal, conditional signaling, synchronous events, predictive control, etc.). This, of course, comes with the drawback of increasing latency and decreasing reliability with the complexity of the processing.

1.3.1 Current Development on Integrators

In this section, the state-of-the-art acquisition systems for each integration strategy are presented. These are also representative of the strategies considered for the ITER magnetic diagnostic integrator.

Tore Supra – Analog Integration

Tore Supra uses an analog integrator for the magnetics which, after the last improvement, sets the state of the art for such of integrators. This particular integrator has a differential structure (see Figure 1.4), meaning the signal is integrated through two cells, based on a auto-compensated Operational Amplifier (OpAmp). In order to minimize the drift, the OpAmps used are temperature stable with only $\pm 0.01 \ \mu V/^{\circ}C$ average input offset drift. With the same objective, polypropylene capacitors were also chosen due to their reduced leakage current. The most recent improvements on this system [6] show an overall 74% reduction in the maximum average drift, siting at $405 \ \mu V \cdot s$ for a $1000 \ s$ integration. These results are, however, well above the ITER maximum drift requirement of $500 \ \mu V \cdot s$ for a 1 hour pulse.





More recent tokamaks still make use of the analog integration. KSTAR [7, 8] has improved its long time integration capabilities by introducing temperature control. The EAST [9] supper-conductive tokamak switches between two sister integrators, while the HL-2A [10] a corrective term, resulting from the



Figure 1.5: Simplified schematic representation of the hybrid integrator for long pulses [11].

integration of a short-circuit in a 'sister' integrator, is constantly subtracted to the signal. The performance of these integrators is however inferior (larger drifts) to the Tore Supra.

DIII-D – Hybrid Integration

In an effort to integrate ITER range pulses $(10^3 s)$ an experimental hybrid (digital/analog) integrator was devised and tested in the DIII-D tokamak [11]. The goal of this approach is to have the benefits of an analog integration at high frequency transient operation and have a digital integration for slowly varying signals, where the input signal is comparable in magnitude to the noise. The flux is calculated digitally according to

$$\Phi = \int V_0 dt = \int V_1 dt + RCV_1 \approx \sum V_i \Delta t + RCV_i$$
(1.4)

(see Figure 1.5, V_i represents the voltage at the input of the amplifier). In this equation, the first term on the right hand side corresponds to the digital integration and the second to the analog.

With an high frequency signal, the digital integration term acts as a correction to the analog integration.

There is also a corrective factor obtained by switching the input signal at the digital stage from the output of the passive RC integrator to a dummy resistive load. This factor is subtracted to V_i in equation (1.4), and is represented by the feedback on Figure 1.5. There is also another corrective term, accounting for the noise the switching introduces in the system. However this factor is measured before the pulse and not in real-time.

W7-X – Digital Integration

In the W7-X stellarator, with predicted experiments up to 30 minutes in duration, the approach for the integration for the magnetic diagnostics was a digital integrator [12].

The key component in this architecture is the chopping of the signal (see Figure 1.6). The chopper consists of CMOS switches used to invert the polarity of the signal at the clock frequency and an associated low-pass filter. The filter ensures that there is no high frequency transient during the switch by spreading the energy in a larger time interval, keeping the integral unaltered. The signal is thereafter modulated by a square waveform. Later, in the digital part of the system, when performing the integration, the signal is demodulated.



Figure 1.6: Schematic representation of the analog part of the digital chopper integrator for W7-X [12].

1.4 Magnetic Diagnostic in the ITER Network Architecture

Due to its usage for plasma/interlock control, the magnetics integration with the networks architecture needs to be fully optimized. ITER's data must be streamed through the Control Data Access and Communication (CODAC) networks (see Figure 1.7). Four separate networks are put in place, segregated and with different protocols, according to their function:

- TCN The Time Communication Network ensures clock synchronization among all ITER systems.
- DAN The Data Archiving Network is the channel by which all the raw data is streamed with the
 objective of being stored for offline analysis. This network has to cope with the fast sampling rates
 of the acquisition systems.
- PON The Plant Operation Network is used for the slow monitoring and configuration of the plant systems.
- **SDN** The Synchronous Databus Network interconnects real time systems and provides the data for control algorithms, both plasma control and interlock. The SDN also connects complementary diagnostics, that are dependent on real time data acquired elsewhere.

As mentioned before, the plant control relies on the magnetics real-time data, distributed by the SDN. This makes the interfacing between the magnetics and the CODAC architecture crucial, on ITER [1] as it was on other tokamaks before [13]. This interface is highlighted in yellow in Figure 1.7, showing the full scope of this thesis.

The magnetic diagnostic will use the SDN technology to interconnect in real-time all the magnetic sensors which are widely distributed in many sectors of the machine. This will allow the centralization of the computation of the magnetics parameters (e.g. plasma shape), before distributing them to the ITER SDN, so that they can be used by the plasma control system.

1.4.1 Distributed Networks on Tokamaks

In any large scientific installation the interconnection between all the plant systems is crucial to a successful operation of the device. In many fusion experiments, digital real-time networks are used to



Figure 1.7: Magnetic diagnostic in the ITER network architecture. Highlighted in yellow, the scope of this thesis. Adapted from [1].

connect the diagnostic data with real-time controllers and subsequently with plasma actuators (e.g. the poloidal field coils control system). This is the case in ASDEX-upgrade [14], DIII-D [15] and JT-60 [16] and JET.

Being the largest fusion device in activity, and being operated by a large international research community, the JET distributed network infrastructure [17, 18, 13] is, in some aspects a model for ITER. This real-time network architecture have proven itself to be robust (meeting sudden demand peeks), scalable, and modular enough to sustain development and improvement of new online real-time applications, without compromising plasma control and safety systems [2].

The Multithread Application Real-Time executor (MARTe) framework [19] proved to be successful in its performance and in line with computer architecture's technological development and market availability trends. Its integration in ITER is still not decided.

1.5 Outline

This thesis describes the development, testing and integration on the CODAC architecture of one key diagnostic for ITER – the magnetic diagnostic. Besides the detailed overview of the system, two components of the diagnostic will be detailed and the testing of such components presented – the integrator board and the integration of the diagnostic on the ITER CODAC architecture.

Chapter 2, aims at closely following the design and testing of the magnetic diagnostic integrator board, developed at IPFN. Furthermore, it details a test battery on the prototype integrator boards conducted in the premises of F4E in Barcelona, Spain with the objective of: (i) increasing the statistical significance of the shortcut test performed in IPFN; and (ii) study the influence of the chopper frequency on the boards performance.

Section 1.4 established the importance of the distribution of the magnetic diagnostic data, processed in real-time, to critical systems through the SDN. In order to demonstrate the reliability of the SDN, that

will distribute the real-time data from the magnetic diagnostic to the other plant systems, a series of prototype activities were performed in F4E and are presented in Chapter 3. The test rig was designed in a way that simulates a ITER plant system using a combination of low cost components and ITER supplied hardware. The employment of low cost components opens the possibility of, with a low budget, demonstrating the reliability and performance of the SDN before implementation of any plant system. This way, while on development phase, other plant systems can chose implement the SDN protocol.

Chapter 2

Magnetic Diagnostic Integrator Design Description

From the whole system of the magnetic diagnostic, this thesis provides a description of the bespoke electronics. This subsystem interfaces with the magnetic diagnostic coils upstream and with the Plant System Controller (PSC) downstream. In between, the scope of this thesis encompasses the port cell electronics (which are basically passive resistors), the transmission line from the port cell to the diagnostic hall and the digital integration board (Figure 2.1). While not including directly the interface with the PSC, since the understanding of the software algorithm is fundamental to the architecture of the digital integrator board, a dedicated section is devoted to this subject.



Figure 2.1: Scope of this chapter on the magnetic diagnostic.

2.1 A Bespoke Integrator for ITER

The analogue part of the integration board can be divided in 5 main stages:

- Input stage resistor divider and first order filter;
- Chopper square modulation, for the low frequency integration;
- Post-chopper Electronics differs between designs and shall be detailed later on;
- Anti-aliasing Filter operational amplifier with a second order low-pass filter;

• **ADC** – Analog to digital converter.

The main goal of the analogue path is to keep the offset as low as possible and its subsequent drift after integration in order to meet the 500 $\mu V \cdot s$ drift requirement. Table 2.1 shows the requirements taken in consideration for the development of the boards.

Description	F4E specification guidelines	
Input dynamic range	\pm 2 V, \pm 5 V, \pm 10 V, \pm 20 V, \pm 50 V, \pm 100 V, \pm 500 V, \pm 1000 V	
Input impedance	$\geq 100 \ k\Omega$	
ADC sampling rate	2 MSPS (maximum)	
ADC resolution	22 bits (maximum)	
Galvanic isolation	$\geq 500 \; V dc$	
Power consumption	<1 W (excluding ADC)	
High input voltage transient	$\leq 1 \ kV$	
High input voltage oscillation	$\leq 1 \ kV$	
Common mode current	$\leq 1 \ \mu A$	
Integrator flux	$\geq 150 \ V \cdot s \pm 500 \ \mu V \cdot s$	
Drift	$\leq 500 \ \mu V \cdot s/hour$	
Integrator output bandwidth (region where 1/f is valid)	$\geq 10 \ kHz$	
Integrator digital saturation flag	Yes	
Integrator digital out-of-range input voltage flag	Yes	
Environmental static magnetic field during operation	10 mT (any direction)	
Environmental magnetic field variation during oper- ation	50 mT/s	
Gain value - setup and readout	Yes	
Filter frequency - setup and readout	Yes	

Table 2.1: Requirements (nominal and features) for the magnetic integrator boards.

In order to have an exhaustive study of digital integration performance, eight different board designs were developed, in two consecutive phases. All the designs are based on a square-wave modulation technique, with changes at the component level. For each design, four modules were manufactured. From these four modules, two of them had small modifications with respect to the original design, while the other two were keep unchanged. The idea behind this strategy was to increase the variety of concepts being tested, while demonstrating repeatability of the results (by using two identical boards).

2.2 Square Modulation Technique

2.2.1 Modulation Method Shortcomings

Even though modulation is a good way of eliminating the electronics noise from the signal [12], the modulation and demodulation process has its shortcomings. The usage of the chopper (an analogue switch) provides a very sharp, and therefore precise, modulation. However, some problems arise in the demodulation process. The square wave generated by the chopper (x(t)) with a frequency f can be seen as a sum of sine waves given its Fourier series:

$$x(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin\left(2\pi(2k-1)ft\right)}{2k-1} .$$
(2.1)

In the frequency domain, the square wave is composed of an infinite number of odd-integer harmonic frequencies, however the pass band of the acquisition is limited from DC to the Nyquist frequency. This means it is impossible to perfectly recover the original signal, and that an error is always introduced in this process.

2.2.2 Design Challenges

While similar systems are already extensively used in fusion reactors, ITER brings a new set of challenges. The main difficulties arise from two factors in which ITER's operation will be different from the previous experiments: hour-long continuous pulses with stringent drift variation requirements; and burning plasmas, where neutron exposure has to be taken in account.

Integration Time

The magnetic coils signal is degraded by various noise sources, thermal and radiation induced voltages. While this is true for all data acquisition systems, the fact that the coil's signal is being integrated over time propagates even the smallest offset, less than an ADC least significant bit (LSB), to a considerable drift. Since the integrator has to operate continuously for at least one hour in the first phase of ITER, a maximum measured drift of $500 \ \mu V \cdot s$ was set as a requirement. To tackle this problem, the input low frequency noise was grouped in two main categories accordingly to its source and the stage it enters into the system: the Wiring Offset (WO) and Electronics Offset (EO). The Wiring Offset is the general term for all the noise components that are added to the analogue input before the chopper. This means that the WO is also driven by sources that are external to the module (e.g. cable junctions). The WO shows some variation over time. The EO, on the other hand, has a high absolute value (reaching hundreds LSB) and shows some variations over the course of hours. Considered downstream from the chopped signal, the EO is generated in the active components, where, in addition to the thermal voltages, there are power supply imbalances, and op-amp and ADC circuit asymmetries.

Figure 2.2 shows at what stage the offsets are added to the signal. While the EO is much larger than the WO, the modulation and demodulation technique has proven itself to be efficient in the mitigation of



Figure 2.2: Simplified scheme of the magnetic diagnostic integrator board with the two noise types and their insertion points on the system.

the EO. Figure 2.3 through Figure 2.5 show the effect that this technique has on the offsets. After the signal modulation, when digitized, the EO appears as a DC component and the WO is modulated by the chopper (Figure 2.3). After the signal demodulation, the WO shows itself as a small DC component, when compared to the EO. The EO, at this stage, is now concentrated at the chopping frequency, averaging 0 (Figure 2.4). After integration, WO originates a drift, with perturbations caused by the EO (Figure 2.5). Indistinguishable from the signal of the probes, the WO is therefore the main concern in terms of signal conditioning in this architecture. A successful mitigation of the WO will result in a low drift over hour long acquisitions. Table 2.2 shows a quick review of the WO/EO differences.



Figure 2.3: Demonstration of the effect of the EO and WO after modulation. Left: on time domain, right: on frequency domain (as function of the frequency, F_c denotes the chopping frequency). One can observe that the WO is modulated, with the signal, while the EO is added after the chopping and appears as a DC offset.



Figure 2.4: Demonstration of the effect of the EO and WO after demodulation. Left: on time domain, right: on frequency domain (as function of the frequency, F_c denotes the chopping frequency). One can observe that the WO is demodulated, with the signal, while the EO is appears now modulated.



Figure 2.5: Demonstration of the effect of the EO and WO after integration on time domain. One can observe that the EO is integrated as perturbations around 0 while the WO is appears now as a drift.

Table 2.2: Summary of the comparison between the WO and EO relative to its source, insertion point, behavior. While the WO has a smaller magnitude, it is not removed by the modulation and demodulation.

	WO	EO
Source	Internal/External	Internal
Insertion, relative to the chopper	Upstream	Downstream
Magnitude	Low	High
Variation over time	High	Low
Eliminated by the (de)modulation technique	No	Yes

2.3 High Frequency Signal Recovery

The integrator board designs described in this thesis are designed for low frequency signals. The first step of the integrator is even a low-pass filter. For control, all phenomena evolved are slow (tens of millisecond scale): actuators, vessel response and flux chances in the pickup coils. It is also only at very low frequencies (close to DC) that there is integration drift and non-zero mean phenomena, and hence the (de)modulation technique employed. Nevertheless, high frequency integrated data is also of interest, namely for MHD studies. The strategy for the wider frequency input range is to add, in parallel with the narrow-band integrator module, another acquisition board with a high input cut-off frequency and to have the chopper disabled. Both signals are then combined to deliver an accurate measurement. This way, the diagnostic has two channels performing digital integration: one with the low-drift focused signal conditioning for the low frequencies and another capturing the high frequency components of the signal. Figure 2.6 shows a diagram of the parallel channels.

The recombination of these two signal is done digitally and will be subject to some empirical calibration. However, this will only be put in place once the development of the integrators is complete and will not be investigated further in this thesis.



Figure 2.6: Block diagram of the two channels showing on top the modulating path, developed in this thesis and on the bottom the high frequency path.

2.4 Magnetic Diagnostic Subsystems Description

2.4.1 Transmission Lines and Port Cell Resistors

Outside of the integrator boards, but also an important part of the diagnostic, the transmission lines have to be accounted in the development of the boards. The magnetic diagnostic integrators will receive signals from 25 different coil types [5]. Connecting the coils to the instrumentations cabinets with the integrators there are going to be transmission lines of different lengths. It was stipulated that there should be only one integrator model and therefore it has to be able to function with any signal produced by any coil. To overcome that, the input gain on the digitalization stage can be configured, allowing a correction factor. Nevertheless, simulations have revealed that the signals from the AA, AB, AC and AJ coils (see Table 2.3) resonate with long transmission lines, thus amplifying the voltage and making the system inviable due to the high voltages and power dissipation that are reflected back into the coils.

Any solution to further mitigate this problem has to comply with the following relevant specifications:
Table 2.3: Identification under the ITER Plant Breakdown Structure (PBS) of a selection of coils with their location.

PBS	Name	Location
AA	Tangential Coils (Inner)	Inner vessel, behind the blanket and under the
AB	Normal Coils (Inner)	divertor on 6 vessel sectors
AC	Toroidal Coils	Top of inner vessel, behind the blanket on 9 vessel sectors
AJ	HF (High Frequency) Sensors	Inner surface of the vacuum vessel and behind the blanket

- the impedance seen by the coil shall be at least 100 kΩ (so that the measurement is fairly insensitive to coil resistance variations);
- the maximum power dissipation of the coil in the worst conditions shall be less than 10 W;
- electronics in the port cell are not allowed (except resistors).

After analyzing worst case configurations in computer simulations it was decided to place resistor networks in the port cell. This way, the required impedance to the system is ensured while at the same time attenuating the resonance between the coil and the long transmission line.

2.4.2 Digital Integrator Boards – Analogue Signal Conditioning

The integrator boards were developed in two consecutive development phases. In each development phase, four designs were devised, each design with four modules in a grand total of 16 prototype boards assembled. All these designs share the same fundamental modulated digital integration working principle¹ and electronic stages. Each prototype board is labeled as $D_x M_y$, where *x* stands for the Design number (1-4) and *y* for the module number (1-4). The different modules of the same design may include some minor variations from the original design. This allows the investigation of the impact of a given component in the performance of the design. In general, two of the modules were kept as the original design and the other two were allowed to have some small modifications at the component level.

As already shown in figure 2.2, the analogue path of the integrator boards is divided in 5 stages. The first stage is formed by a differential Thévenin voltage divider and a passive low-pass filter. After this stage, the signal goes through a differential analogue switch that performs the actual chopping, inverting the signal periodically. Depending on the design, the signal then passes through buffers and operational amplifiers implementing a second order anti-aliasing low-pass filter. Two of the designs have modifications in the analogue path – **D2 Barcelona** has ceramic filters (band-pass filters) after the buffers, and in **D7 Mestre** the buffers are replaced by capacitors in series. Figure 2.7 shows the architectural differences among the designs.

¹Design **D2 Barcelona** has a modified working principle, see 2.4.2 for details.

Table 2.4: Identification and classification of the eight integrator designs prototyped. The designs were developed under two consecutive development phases and indexed from 1 to 8. To facilitate communication during the development, each design was given an alias.

Development Phase	Design	Alias
	1	Lisbon
	2	Barcelona
	3	Oxford
	4	Rome
	5	Menorca
	6	Lagos
	7	Mestre
	8	Edinburgh



Figure 2.7: Schematic representation of the electronic stages of the analog path for the three configurations. Each configuration labeled with the designs that use it.

Input Stage

The main objective of this stage is to attenuate the large input signals so they fit within the ADC input range. By adding a capacitor to this resistor network, the divider performs also as a 1st order low-pass filter for pre-filtering high input signal frequencies. This filter is critical to to avoid the artificially DC voltage generation when the input is a multiple of the chopper frequency.

A schematic representation of this stage is shown in Figure 2.8.

The values of the passive components were chosen having two considerations: attenuation according to the ADC chosen for each design, and a cut-off frequency of around 10 Hz, as shown in table 2.5.



Figure 2.8: Left: Topology of the input stage, showing a Thévenin voltage divider and a passive low-pass filter in a symmetrical architecture. Right: Frequency response of the filter.

Table 2.5: Input stage parameters: static gain achieved as an attenuation of the signal by the resistor divider, and cut-off frequency set by the passive RC filter.

De	esign	Static gain	Cut-off frequency
1	Lisbon	1/6	10 Hz
2	Barcelona	1/6	10 Hz
3	Oxford	1/11	10 Hz
4	Rome	1/11	10 Hz
5	Menorca	1/6	10 Hz
6	Lagos	1/6	10 Hz
7	Mestre	1/6	10 Hz
8	Edinburgh	1/11	10 Hz

Chopping Stage

This stage is the key component of the digital integrator and provides a conceptual barrier between the input stage and the following electronics. The modulation stage consists of an analogue chopper which inverts the signal periodically with a 50% duty cycle.

Being a square modulation, the signal is spread around the chopper frequency and its odd harmonics (see Figure 2.9). Consequently, the demodulation process can be done by picking one harmonic (using a sinusoidal demodulation signal) or picking all the harmonics recombining them (using a square demodulation signal).

On the design **D2 Barcelona** filters are placed to pick only the third harmonic $(3 f_c)$. The demodulation is then achieved using a sinusoidal demodulation signal on the FPGA (see 2.6.1).



Figure 2.9: Representation of the chopped signal in the frequency domain. The original signal is reproduced every odd multiple of the chopping frequency.

Electronics Post Chopper

In this stage the different designs use different techniques to condition the signal: Most of the designs (**D1 Lisbon**, **D3 Oxford**, **D4 Rome**, **D5 Menorca**, **D6 Lagos** and **D8 Edinburgh**) only add buffers, in order to increase the impedance seen by the chopper aiming at reducing the currents to a minimum and avoiding the asymmetric voltages that these currents might generate. The **D2 Barcelona** design also has buffers to increase the impedance seen by the chopper, however, it includes a set of narrow band-pass ceramic filters in order to pick only the third harmonic. The **D7 Mestre** design has capacitors in series which provide a voltage separation between the chopper and the ADC. This also allows the shifting of the common mode voltage to the middle of the ADC input range without using a fully differential operational amplifier with common mode input and thus greatly simplifying the circuit (which benefits both the integrator reliability and cost).

Anti-aliasing Filter

This stage works as a driver for the ADC and, at the same time, provides filtering to prevent aliasing. This is achieved by operational amplifiers which act as an active low pass filter. The filter implemented is a multi-feedback second order filter of the Butterworth response type. This topology (see Figure 2.10) was chosen as it provides a good compromise between a good step response and good attenuation at high frequencies. It was implemented with a fully differential operational amplifier with a common mode input to adapt the voltage level to the ADC. This configuration provides enough attenuation at high frequencies, avoiding aliasing.

It was decided to use a cut-off frequency (-3 dB) around 100 kHz. In **D2 Barcelona** a much higher modulation frequency is used (approx. $151 \ kHz$) and thus the cut-off of the anti-aliasing filter is set to $1 \ MHz$. The gain is unitary in all designs.



Figure 2.10: Left: Topology of the Multi-Feedback anti-aliasing filter in a symmetrical architecture. Right: Frequency response of the filter.

Analog to Digital Conversion

The ADC is one of the most important components of the board. Therefore two models were tested across the designs. The signal is digitalized at the rate of 2 MSPS in both cases and aiming to obtain the maximum possible Effective Number Of Bits (ENOB).

The first ADC is a charge redistribution Successive Approximation Register (SAR) ADC. The topology of this ADC is fully differential and it is a 18 bit ADC with a maximum acquisition rate of 5 MSPS (set at 2 MSPS). In the first design phase, this ADC was installed in the designs **D1 Lisbon** and **D2 Barcelona** while for **D3 Oxford** and **D4 Rome** a higher resolution delta-sigma ($\Delta\Sigma$) ADC was installed. The 24-bit ADC chosen has 23 bits when operating at 2 MSPS.

Galvanic Isolation

In addition to the essential components to implement this integration strategy, the architecture chosen has a 1kV galvanic isolation [20]: the communications of the FPGA with the outside is done via capacitive coupling, while the power supplies are isolated via magnetic coupling. The full isolation of the electronics adds another degree of flexibility and at the same time, avoids ground loops and noise from the outside.

2.5 Differences Between Modules

The integrator board is based on the W7X design. That was the basis for the **D1 Lisbon** design and the modifications the other designs exhibit will be presented as relative to this design and summarized in table 2.6.

As previously mentioned, design **D2 Barcelona** has a different operation than **D1 Lisbon**, as ceramic filters are added after the chopper to select only the third harmonic. This has three implications: (i) the chopping frequency is chosen as 151672 Hz, so that the third harmonic (455016 Hz) falls in the passband of the filter ($455 \pm 6 kHz$); (ii) as the high chopping frequency would conflict with the anti-aliasing

filter, the cut-off frequency of this filter is changed to 1 MHz; (iii) in the selection of the resistors and capacitors, the output impedance of the anti-aliasing filter has to be taken into account, in order to match the output impedance of the ceramic filters.

Design **D3 Oxford** goes back to the original square modulation-demodulation process, introducing a new ADC. Furthermore, this design is focused at the possible impact of the charge injection of the chopper to the final drift. For that reason snubbers are introduced before and after the chopper. These snubbers aim at suppressing the voltage spikes caused by the circuit's inductance when the fast switchings (of the chopper) occur. As previously mentioned, this design introduces a $\Delta\Sigma$ ADC that provides a higher resolution without compromising the sampling rate. The introduction of this ADC forced small modifications in the input attenuation value. This model also implements new Operational Amplifiers (OpAmp) in the anti-aliasing filtering stage. This model, represents a trade-of, having higher input voltage noise but less total harmonic distortion.

Continuing the efforts to reduce the asymmetric charge injected by the chopper, design **D4 Rome** uses a different chopper with low charge injection (represented as C in table 2.6). This design is based on the previous, however, two of the modules do not have the snubbers (before and after the chopper).

These four designs constitute Phase I and were all produced and tested simultaneously. The following designs were developed after the testing (see 2.6.2) and analysis of the first phase. The focus of the new design changes are the reduction of the noise levels and the thermo-electrical voltages of the input stage. For this reason, a box is added around the input stage of the boards in order to implement temperature control. The chopper for the new designs is the low charge injection chopper used in **D4 Rome**. With the objective of reducing the noise on the high impedance buffer stage, the OpAmps used ion the second phase are Bipolar Junction Transistors (BJT) instead of Field Effect Transistors (FET) as in the first phase. However this change comes with a reduction of the impedance. As for the Anti-aliasing filter, the phase II modules have the same topology but different response type. Instead of Butterworth, a Bessel response family filter is chosen in order to avoid over-voltages during the chopper transitions.

D5 Menorca and **D6 Lagos** are very similar, however, in **D6 Lagos**, a common mode voltage is applied at the input stage in order to accommodate the voltage to the middle point of the input stage of the ADC from the very beginning.

With the **D7 Mestre** design, the general idea was to reduce the number of active components (OpAmps) with the goal of reducing the circuit complexity and thus the noise levels and the cost, whilst aiming at increasing the manufacturing yield and the circuit reliability. Instead of buffers with OpAmps, capacitors are placed in series, right after the chopper, providing voltage separation between the input stage and the ADC while allowing to reference the voltage to the middle point of the ADC. These capacitors act as a high-pass filter, which is not a problem, since the low frequency signal is at this stage modulated to high frequency and on the low frequency band only noise can exist (EO). With no buffers, the impedance seen by the chopper is now set by the anti-aliasing filter. Therefore, the topology of this filter is a Sallen-Key (see Figure 2.11), with higher impedance for a equivalent cut-off frequency. On the other side of the trade-off, the frequency response o the Sallen-Key is not as sharp as with the Multi-Feedback.

22



Figure 2.11: Left: Topology of the Sallen-Key anti-aliasing filter, in a symmetrical architecture. Right: Frequency response of the filter.

D8 Edinburgh is similar to **D5 Menorca**, introducing a new approach to control the noise at the input stage (WO). The four resistors in the input stage (Figure 2.8) are now part of a single chip. The idea is that the temperature on the chip is more homogeneous than having the independent resistors separated. The more homogeneous setup is expected to prevent asymmetries that can lead to WO and therefore drift. The introduction of this component forced the usage of a different gain, having the nominal value of the capacitor been adjusted to assure the 10 Hz Cut-off frequency.

Table 2.6: Summary of the differences between all prototyped modules of all designs. Components described in the text, SAR ADC has 18 bits and $\Delta\Sigma$ 23.

Design	Module	Temperature control	Snubbers before chopper	Chopper	Snubbers after chopper	Electronics post chopper	Anti-aliasing filter	ADC
D1 Lisbon	1,3	No	No	A	No	High Impedance Follower (HIF)	Multi-Feedback Topology (MFT)	SAR
	2,4	No	No	В	No	HIF	MFB	SAR
D2 Barcelona	1,3	No	No	A	No	HIF and Ceramic Filters	MFB with 1 MHz cut-off frequency	SAR
	2,4	No	No	В	No	HIF and Ceramic Filters	MFB with 1 MHz cut-off frequency	SAR
D3	1,3	No	Yes	A	Yes	HIF	MFB	$\Delta\Sigma$
Oxford	2,4	No	Yes	В	Yes	HIF	MFB	$\Delta\Sigma$
D4	1,3	No	Yes	с	Yes	HIF	MFB	$\Delta\Sigma$
Rome	2,4	No	No	с	No	HIF	MFB	$\Delta\Sigma$
	1	Yes	Yes	с	No	Different OpAmp	MFB	SAR
D5 Menorca	2	Yes	Yes	с	Yes	Different OpAmp	MFB	SAR
	3	No	Yes	С	No	Different OpAmp	MFB	SAR
	4	No	Yes	с	Yes	Different OpAmp	MFB	SAR
	1	Yes	Yes	с	No	Different OpAmp	Sallen-Key	SAR
D6	2	Yes	Yes	с	Yes	Different OpAmp	Sallen-Key	SAR
Lagos	3	No	Yes	с	No	Different OpAmp	Sallen-Key	SAR
	4	No	Yes	с	Yes	Different OpAmp	Sallen-Key	SAR
	1	No	Yes	с	No	Capacitors	MFB	SAR
D7	2	No	Yes	с	Yes	Capacitors	MFB	SAR
Mestre	3	Yes	Yes	с	No	Capacitors	MFB	SAR
	4	Yes	Yes	с	Yes	Capacitors	MFB	SAR
	1	No	Yes	с	No	HIF	MFB	SAR
D8 Ed- inburgh	2	No	Yes	с	Yes	Different OpAmp	MFB	SAR
	3	Yes	Yes	с	No	Different OpAmp	MFB	SAR
	4	Yes	Yes	с	Yes	Different OpAmp	MFB	SAR

2.6 Prototype Boards Testing

In order to assess the performance of the boards, several tests were performed under different conditions. Two tests per board are systematically done in order to increase reliability. Given that the hardest performance specification to meet is "the drift after one hour must be below $500 \ \mu V \cdot s$ ", on most of the tests (see 2.6.2), the criterion to decide if a test succeeds or not is to check the average of the final drift and compare it against the specification. However, due to the amount of data generated (> 28.8 GB per channel per hour), the tests are executed with only half an hour effective experiment time. By doing so, the assumption is being made that the drift is somewhat linear, and that the threshold can be consequently adapted ($|drift| \leq 250 \ \mu V \cdot s$ per hour). The value for the drift is obtained as the integrated signal at the experiment end time. Figure 2.12 shows an example of a short circuit test for two boards. The signal in red represents the normal behavior for these tests, with the measured value corresponding to the maximum absolute error. In the signal in blue, however, this does not happen: the measured value is not the maximum absolute error and the behavior not so linear. If the experiment were to have ended in the 1300 s mark and then extrapolated for one hour, **D4M4** would show a way better performance, while if on the 1500 s mark, both models would have registered similar values.



Figure 2.12: Example of short circuit test performed in the premises of F4E for two boards with the design **D4 Rome**: **D4M3** in blue and **D4M4** in red. Two phenomena are visible: the maximum drift was not obtained in the end of the experiment for the signal in blue and an inversion of the drift.

Each acquisition is composed by three stages, followed in succession:

- Warm-up Power on the board and configure the chopper frequency and the ADC sampling rate. No voltage should be applied to the analogue input board. The chopper must be active and the ADC acquiring data (it is not necessary to save it). During operation, this stage is not expected to be performed frequently as the boards will likely be continuously operating and providing data to the central I&C systems.
- 2. Calibration When the temperature of the board is stable, the calibration can start. It is extremely important that the analogue input voltage applied is 0 V and no interferences from other devices are present. If there are interferences during the calibration, the error will be accumulated over time, leading to a large error at the end of the experiment. During this phase, two different calibrations are done: (i) EO calibration before demodulating the signal, the average of the signal is recorded for later subtraction from the signal. (ii) WO calibration when the EO correction value is obtained, the WO calibration can start. It is based on the drift measurement of the integral signal (after demodulation) when no voltage is applied. This value is recorded and then subtracted from the signal. For testing, the WO calibration time should be of at least 10 minutes, this value will have to be revisited given the ITER conditions after the system is installed.
- 3. **Pulse** When an experiment starts, the integral value is set to 0 and the signal is acquired. The voltage should never exceed the maximum expected value, otherwise the signal saturates resulting in large errors and increasing the risk of causing permanent damaging to the boards.

2.6.1 Real-time Integration Software

According to the concept described, besides the integration, the acquired data must be compensated (to remove the calibration drift) and demodulated. The real-time integration software, that shall run on the boards FPGAs during operation² performs the following operations to integrate the signal:

- EO compensation The EO (electrical offset) compensation tries to compensate for the offset from the electronics between the analogue chopper and the ADC (included). First, during the calibration phase and before demodulating the signal, the offset value of the signal is computed averaging 30 seconds of data. Onwards, this value is subtracted from the data. The EO value computed is a rounding of the average, expressed as a 32 bit integer.
- 2. **Demodulation** The demodulation process inverts the signal accordingly (and synchronously) to the chopper signal.
- 3. Interpolation The interpolation is an extra feature which allows to reconstruct the transition of the chopper using one of the following alternatives: (i) Linear interpolation replaces the value of a given number of pre-configured samples by a linear interpolation of the first and the last sample considered during the interpolation period. The configurable parameter is the number of samples to interpolate. (ii) Hold last value replaces the samples value with the last value before the

²During testing phase the boards do not have an embedded FPGA but rather an acquisition board that implements the described algorithm.

chopper transition. The configurable parameter is the number of samples to hold. (iii) Set 0 - sets to 0 the values of all the samples during the chopper transition. The configurable parameter is the number of samples set to 0.

- 4. Integration Performs a trapezoidal integration in integers (ADC units). Afterwards, however, the integral is converted into a floating point to be compensated and saved in physic units $(V \cdot s)$.
- 5. WO compensation The WO compensation is computed by performing an integration without any signal in the input during the calibration phase. In the pulse phase, the WO value is integrated and subtracted from the integrated data, however, it could also be subtracted before the integration and then integrated afterwards.

The integration software requires a calibration start trigger which indicates when to start calibration. The algorithm manages the EO and the WO triggering times, against pre-configured time-windows. At the end of the process the algorithm triggers a flag indicating that it is ready to integrate and waits for the start integration trigger. When the algorithm receives a start integration trigger starts integrating using the pre-calculated calibration parameters while stop integration trigger is not received.

The experiment is controlled using MARTe [19] and the data archived using MDSplus [21]. To access the data, a Java tool jScope is used. Additionally, a set of python scripts were conceived to generate plots of the magnetic tests. This set of scripts allows saving the data in smaller, decimated binary files for later custom plotting. Figure 2.12 shows a plot outputted by this Python interface with a single command line instruction, since the tools are modular and with a command line arguments interface.

2.6.2 Test Plan

The tests on the boards were performed in three different locations: IPFN, CCFE and F4E. Table 2.7 shows a summary of the tests conducted. The boards were developed in IPFN and therefore, the tests conducted there were mostly concerned with the electronic specifications showed in Table 2.1. Being the main testing location, with a custom test rig, and having developed the digital interface for the testing, CCFE tests are mostly concerned with the performance and specifically, with the 500 $\mu V \cdot s$ drift requirement. The CCFE test rig consists of a permanent magnet that is moved in (and later out) of a probe coil, connected to the input of the integrators. Additionally, a stimulation coil with a signal generator was placed close to the probe, as to cause perturbations.

The results and discussion of the tests performed in both IPFN and CCFE fall outside of the scope of this thesis. Nevertheless, the F4E testing is detailed and discussed in the next section.

Table 2.7: Description of the tests done to the magnetic integrators in IPFN, CCFE and F4E.

Test	Premises	Description
ENOB	IPFN	The goal of this test is to measure the Effective Number Of Bits(ENOB) and check the correctness of the boards. For that, 4 second pulses were configured with no EO or WO compensation, no interpolation and without the chopper active. The input signal is sinusoidal with 20 Vpp and a frequency of 10, 100, 1 k, 10 k 100 k and 200 kHz. The input filter is removed, as the interest component is the ADC.
Short circuit	IPFN	The goal of this test is to integrate a short circuit for 30 minutes and then measure the offset (drift). A resistor with 0, 270 and 510 Ω connecting both ends of the input. In the boards configuration, a normal operation pulse in configured, with 30 s EO calculation time and 600 s WO calculation time.
Linearity	IPFN	The goal of this test is to measure the linearity of the system by plotting the voltage measured by a voltmeter and the inter- polated signal. The signal is a DC voltage generated with a battery and a resistor divider. No WO compensation is con- figured as the signal is not being integrated. a EO calculation time of 1s is used. The interpolation is turned on on the Hold Last Value mode with 50 samples. Each pulse has a duration of 30 s.
Chirp	IPFN	The goal of this test is to check the impact of the input sig- nal frequency on the drift. Using a signal generator, a sine wave signal with 4 V amplitude is fed to the integrator with a frequency sweeping from 300 Hz to 3 kHz. In this sweep, the critical frequency at twice the chopping frequency is applied for two seconds (approx.). The configuration is normal, with no interpolation.
Long test	IPFN	The goal of this test is to integrate a short circuit for a long period of time (50 h and 12 h) and measure the maximum drift variation during 1 h. As for configuration, no EO or WO correction enabled and the chopper is off.
Permanent magnet	CCFE	This tects is the base line test for the following tests, acting as control. The point of this test is is to generate a 0 mean signal for 20 minutes and measure the offset at the end. This is achieved by sliding a permanent magnet in and, 20 minutes after, out of the coil. configurations are 60 s EO calculation time and 600 s WO calculation time, no interpolation. The pulse lasts 30 minutes.
Frequency sweep	CCFE	Base line test conditions plus a 0 mean stimulation using an external coil. Critical frequency applied for one second.
Beating	CCFE	Base line test plus a beating signal (0 mean) applied with the external coil.
Chopping Frequency	F4E	See section 2.6.3.

2.6.3 Chopping Frequency Dependence Tests

As previously mentioned, the tests performed at IPFN and CCFE were performed twice for each board and with a chopper frequency of around 1 kHz ³. This procedure was chosen as: (i) the chopping frequency has no clear requirements and therefore a reasonable value of 1 kHz was chosen (comfortably bellow the anti-aliasing filters cut-off frequencies); (ii) the experiments are run only four boards at each time (16 total boards for each phase) and the experiments last more than 45 minutes, therefore it is time costly to run more than two tests on each board during development phase. Nevertheless, when phase II of the development initiated, some of the boards of the first phase were shipped to F4E for further testing.

The testing conducted has the following objectives:

- Study the chopping frequency influence on the final drift. Some electrical components on the integrators have regions of lower noise for higher frequencies. Would the performance be improved using a higher chopping frequency?
- Increase the statistical relevance of the short circuit tests. Comparisons were made between
 modules with very little results, performing these tests with more pulses will bring a higher degree
 of confidence to the result interpretations.

The tests were conducted in two rounds, allowing some preliminary analysis of the results and improvements to the test plan according to the conclusions taken. The first round of tests was performed under the following conditions:

- 10 x 30 min acquisitions, with:
 - 10 min warm-up,
 - 10 min WO calculation time,
 - 30 s EO calculation time;
- short circuit;
- acquisitions that show high temperature variations are removed from the pool;
- only decimated integrated and temperature data is saved;
- the drift is obtained by taking the mean of the last 100 values of the decimated integrated data;
- data processing relates to the absolute error, i.e. the absolute value of the drift described above.

On the configuration software, the chopping frequency is controlled by the chopper parameter (CP) and therefore the values chosen for the chopping frequency (in Hertz) are not exact numbers as the chopper is implemented in the FPGA using a counter. The frequency chosen for all previous testing was around 1 kHz, chopper parameter (CP) 64, in this tests the chopper frequency is consecutively doubled, starting at CP 32: [32, 64, 128, 256, 512, 1024]. Additional tests are conducted:

³Except for **D2 Barcelona**, whose chopping frequency is given by the ceramic filters.

- At the D2 Barcelona frequency (cp 9940, 152 kHz): This design has its chopping frequency limited by the passing band of the ceramic filters (455 ± 6 kHz), therefore the chopper frequency is predetermined in order for the filters to pick the third harmonic. To check whether the bad results of the D2 Barcelona design can be explained (partially, at least) by the high chopping frequency, rather than design flaws, an experiment will be conducted at this frequency on the other designs;
- At high frequency (cp 1024, 16 kHz) with linear interpolation, to check if the there is a performance improvement due to the interpolation, something not observed at the standard frequency testing.

In the second round of the testing, the following alterations were made to the test plan: After concluding 10 experiments for each frequency does not provide enough result consistency, it was decided to increase the statistics, running 90 more experiments for each frequency. Furthermore, one additional frequency was used, following the same exponential fashion: (cp 2048, 32 kHz). In order to run these many experiments, and given that the experiments are automated to run in succession, the 10 min warm-up was dropped, as it is not needed except for the first experiment of each batch. The board with the worsts results was also dropped, as the results have proven to be significantly worse than all the others.

First Round Results

Table 2.8 shows the result of the average and standard deviation of the 10 experiments for each board tested. This data is also represented graphically in the plots of Figures 2.13 and 2.15. The standard deviation is represented in a separate plot in order not to over saturate the first plot (Figure 2.14).

Table 2.8: Ro	und one of	testing, with	10 short	circuit p	oulses wit	h half ho	ur duration.	Mean (µ)	and
standard devia	ation (σ) of t	he absolute o	lrift prese	nted with	all result	s express	ed in $\mu V \cdot s$.		

Frequency		D1	M3	D1M4		D3M3		D4M3		D4M4	
CP	[Hz]	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
32	488.28125	33.4	24.2	51.3	56.1	58.1	32.0	61.9	46.6	63.7	51.2
64	976.5625	40.8	24.6	63.7	86.4	45.3	46.2	56.4	31.4	56.9	39.8
128	1953.125	43.1	31.6	94.6	93.4	33.5	21.3	18.4	12.3	34.7	26.7
256	3906.25	46.0	32.7	98.4	44.2	51.7	57.3	73.5	52.0	38.4	29.4
512	7812.5	60.1	53.7	261.6	222.1	37.6	23.2	41.2	24.5	30.6	33.3
1024	15625	114.5	65.0	464.2	535.4	40.8	25.4	20.1	10.4	44.2	42.2
9940	151672.36	81.0	52.5			18.7	18.4	15.3	11.0		1

For a more clear analysis, Figure 2.15 shows a repetition of the plot in Figure 2.13 with the **D1M4** data removed. This plot shows an overall reduction of the drift with the increasing frequency for the Designs 3 and 4 while the opposite seems to happen for the Design 1 (also on the **D1M4**, removed from this plot). It is worth noticing that there is an increase of the drift from the (**CP 128, 1,9 kHz**) to (**CP 256, 3.9 kHz**) points across all boards.



Figure 2.13: Mean values for the first round of the testing, with 10 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.8.



Figure 2.14: Standard deviation values for the first round of the testing, with 10 short circuit pulses with half hour duration.

For the experiment at **(CP 9940, 152 kHz)** (see last row of Table 2.8), it seems clear that the underperformance of Design 2 cannot be explained by the high chopping frequency. Note that **D1M3** and **D3M3** share the same chopper with **D2M3**. Also, this frequency cannot be a working frequency as most designs have a low pass filter tuned at 100 kHz, being the low absolute errors obtained expected.

Table 2.9 shows the results of the repetition of the acquisition at high frequency (CP 1024, 16 kHz)



Figure 2.15: Mean values for the first round of the testing, with 10 short circuit pulses with half hour duration. Same as the plot in Figure 2.13 without the outlier **D1M4**. Absolute drift versus chopping frequency, as in Table 2.8.

for the **M3** boards with linear interpolation of 50 samples. The results show an increase in drift of several standard deviations instead of a decrease.

Table 2.9: Pulses at **(CP 1024, 16 kHz)** with and without linear interpolation of 50 samples. Mean (μ) and standard deviation (σ) of the absolute drift for 10 short circuit pulses with half hour duration. All results expressed in $\mu V \cdot s$.

Interpolation	No Interpolation		50 Sa	mples	
Board	μ	σ	μ	σ	Improvement (%)
D1M3	114.5	65.0	212.0	142.3	-85.2
D3M3	40.8	25.4	72.3	58.5	-77.1
D4M3	20.1	10.4	114.6	99.6	-469.5

Second Round Results

Since no clear trend can be inferred from the tests with a sample size of 10, the sample size of the four modules with the best results was increased to 100. As the statistical sample is larger, the error bars represent the confidence interval at 90% confidence. Table 2.10 provides a summary of the differences across the tested boards and Figure 2.16 an overview of the results (Table 2.11). Comparing with the first round, these results show a more stable behavior across the full frequency span. However, an alteration of the behavior persists for higher frequencies.

The main conclusion to take from this testing is that it is safe to increase the chopping frequency up to around 10 kHz, purely on the drift point of view. Also, if further testing were to be conducted at a

Table 2.10: Summary of the module differences regarding Chopper and Snubbers. Identification of the chopper models coherent with Table 2.6.

Board	Chopper	Snubbers
D1M3	А	No
D3M3	А	Yes
D4M3	С	Yes
D4M4	С	No

Table 2.11: Round two of testing, with 100 short circuit pulses with half hour duration. Mean (μ) and standard deviation (σ) of the absolute drift presented with all results expressed in $\mu V \cdot s$.

Fre	Frequency		D1M3		D3M3		D4M3		D4M4	
CP	[Hz]	μ	σ	μ	σ	μ	σ	μ	σ	
32	488.28125	39.3	36.1	46.4	32.2	45.9	36.8	40.4	39.9	
64	976.5625	44.1	37.9	33.5	23.4	37.9	27.1	38.7	33.0	
128	1953.125	47.0	32.8	44.5	35.4	54.2	59.0	30.4	22.1	
256	3906.25	37.2	30.2	43.9	41.4	56.3	44.5	30.8	23.3	
512	7812.5	42.2	29.2	48.3	43.1	52.7	38.4	33.1	26.3	
1024	15625	76.1	61.9	43.4	29.7	14.5	10.6	24.6	28.9	
2048	31250	145.4	112.6	40.1	33.1	122.6	84.4	72.3	68.6	



Figure 2.16: Mean values for the second round of the testing, with 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.

different frequency, say 8 kHz, these results show that the drift measurements could be compared with previous tests at 1 kHz. Also, the importance of taking a large statistical sample in testing was shown.

The results of the second round versus the first (see Figures 2.13 and 2.16) show different trends and therefore conclusions. For future testing, it should be noticed that averaging the final drift result of up to 10 experiments might not be sufficient to take development relevant conclusions.

The plot in Figures 2.17 and 2.18 show a side-by-side comparison of two modules with the same chopper but with and without snubbers. It is noticeable that the results are very constant. Even though, in Figure 2.17, for the two highest frequencies, the module without snubbers shows worse results, we also have to factor that there are other components changed, besides the snubbers (including the ADC). In both plots, the model without snubbers seems to have less drift for the stable low frequency region, although very slightly for the modules with the chopper A.

Looking at the plots of the boards with and without snubbers, changing the chopper component, we can observe a difference of behavior in the high frequencies. There is however no apparent frequency influence on the drift that can clearly be attributed to the different chopper.



Figure 2.17: Mean values for the modules with the chopper A. 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.



Figure 2.18: Mean values for the modules with the chopper C (**D4 Rome** design). 100 short circuit pulses with half hour duration. Absolute drift versus chopping frequency, as in Table 2.11.

Chapter 3

Real-Time Network Performance

Developed by ITER CODAC, the SDN ensures high-availability, deterministic transport for real-time feedback control data and asynchronous events. This network will mediate the communication between the Plant System Instrumentation and Control (PS I&C), for both sensing and actuation and the Plasma Control System (PCS), running the plasma control algorithms.

Technically, it is based on UDP multicast over a 10 Gigabit Ethernet (10 GbE) protocol. Given the shear size of the plant system – both physical size and in number of subsystems – the network infrastructure is composed of between 50 and 100 PCS nodes in different locations. Additionally, the plant system requires different control cycles ranging from fractions of Hz to a few kHz [22].

A control cycle is considered to have the following steps: (i) acquisition and pre-processing of a signal by a sensor node; (ii) communication to the PCS; (iii) computation of the required action, according to the control algorithm; (iv) communication to the actuator node; and finally, (v) actuation. This five-step procedure has to be performed in the control cycle period the communication times (latency) have to be low having for this network a maximum value of $100 \ \mu s$. Going one layer deeper, this means that is these $100 \ \mu s$ have to account for: (i) software Application Programming Interface (API); (ii) UDP protocol stack; (iii) Operating System (OS) and driver overheads; (iv) delays in switches; and (v) signal propagation in cables.

Another particularly sensitive time in a control network is the difference in arrival time of successive packets (jitter). A control system can be designed with a given control cycle period to cope with the latency of the network, as long as that time value is precise. Therefore, the jitter budget for the SDN is $50 \ \mu s$ and the total bandwidth comprised between $25 \ MB/s$ and $100 \ MB/s$.

In addition to this regular and predictable network load (control cycles), SDN shall also support transmission of asynchronous events with guaranteed delivery within 1 ms.

3.1 Purpose of the Experiment

The purpose of this experiment is to test the performance of the network setup to be implemented in the the ITER SDN. Instead of the whole magnetic diagnostic, the test rig will be based on a smaller system,

the ITER Electron Cyclotron Resonance Heating (ECRH). The same network infrastructure (hardware and software) will be implemented in the ECRH system, connecting all the gyrotrons to the Electron Cyclotron Plant Control (ECPC) and in the magnetics diagnostic to connect and centralize the data acquired by many distributed data acquisition systems.

The network to be modeled by these tests connects a master, the ECPC, to an array of slaves, compromising several gyrotrons (of different makes), the associated transmission lines and launchers. Each of these components requires different instructions and has a different feedback at a frequencies ranging from 1 to $10 \ kHz$.

In order to measure to what extent the communication is reliable, this test will simulate the communication between the ECPC and its plant systems and test it in ITER-like working conditions and stress loads. Specifically, the network should: (i) be robust, having no packet loss; (ii) low-latency (< 100 μs round trip); (iii) have a low jitter.

The ultimate goal of these tests is to prove the SDN is a viable as a single network protocol to manage the interchange of real-time data between the ITER plant systems. This technology can also be used inside complex plant system to control and monitor the behavior of internal plant components, where the industrial norm prevails and which imposes hardware restrictions such as the use of 1GbE interconnects.

3.2 Hardware and Test Rig

In this setup, Hardkernel ODROID-C2® single board computers (see figure 3.1) create a fairly large SDN eco-system that can be deployed without consuming large amounts of power and requiring little floor space, thus allowing to easily increase the number of participants in the network. Nevertheless, all the time sensitive measurements are performed in ITER fast-controller machines and not on these low performance computers. The ITER fast-controllers, supplied by IO, are the same models that will be used in the plant systems of the tokamak, ensuring the relevance of this test bed.

3.2.1 ODROIDs

The slaves in the ECRH system are simulated by the ODROID single board computers. With a retail price of 56\$ (June 2017), this small, low power consumption computers (Figures 3.1 and 3.2) constitute a cheaper alternative to using ITER Fast Controllers or even conventional off-the-shelf conventional PCs. Besides the savings in costs (initial and running) this solution also saves floor space, allowing the experiment to be run in F4E.

The performance (specially the real-time performance) of these computers is way inferior to the fast controllers, but since SDN libraries can run in the ODROID's ARM architecture, the generated traffic is indistinguishable from the data generated by a fast controller or a slave in the ECHR system or an integrator board in the magnetics diagnostic. These computers shall therefore not be used for sensitive time-stamping and might impose different latency footprints than the ITER grade hardware, but for stress



Figure 3.1: ODROID-C2 ARM 64 bit 1.5 GHz quad core single board computer used for the experiment. Source: [23].

testing and traffic generation, this setup allows to scale up the number of participants on the network, without compromising the accuracy of the experiment – different, real computers, independent of each-other.

Given the broad offer of single board computers, the ODROID C2 was chosen due to its technical specifications, particularly, its Gigabit Ethernet support. Table 3.1 list the most relevant specifications of the ODROID machines used, software and hardware-wise. To successfully execute the test-plan of the experiment, some software modifications had to be made to computers in kernel-space. These alterations are explained in section 3.3.2.

OS	Ubuntu 16.04.1 LTS
Kernel	3.14.79-89 aarch64
CPU	Amlogic S905 Quad Core Cortex [™] -A53 1.5 GHz 64 bit ARMv8
RAM	Samsung K4B4G1646D : 2 GByte DDR3 32 bit RAM (512 MByte x 4pcs)
Ethernet PHY	Realtek RTL8211F compliant with 10Base-T, 100Base-TX, and 1000Base-T IEEE 802.3 standards
Power	5V2A DC input
General Purpose Input and Output (GPIO)	Libraries available for Python and C++

Table 3.1: Selection of specifications for the ODROID C2 used in the experiment. Hardware specifications according to the manufacturer [23].

3.2.2 ITER Fast Controller

The main components of the experiment are two powerful computers that communicate using the SDN. These machines are from the pool of computers that (at the time of the experiment) are being considered for the role of ITER Fast Controllers. One of the computers plays the role of the master and the other a slave, as do the ODROIDs. This computers are referred throughout this thesis by the aliases miniCODAC1 and miniCODAC2 for the master and the slave respectively.

The computers used are not identical but very similar, sharing the same CPU and network cards and running the same linux MRG (Messaging Real-Time Grid) kernel intended for use in deterministic response-time situations. Table 3.2 shows the most relevant soft and hardware specifications. The usage of a MRG kernel can bring significant improvements in the real-time performance of the computers, as previously demonstrated using the JET Real-Time Data Network [17].

Table 3.2: Selection of specifications for the ITER supplied fast controller computers (miniCODAC1 and miniCODAC2) used in the experiment.

OS	Red Hat Enterprise Linux Workstation 6.5
Kernel	Kernel 3.10.0-514.rt56.210.el6rt.x86_64
CPU	Intel® Xeon® CPU E5-2418L 2 GHz
Network Card	Intel® I350 Gigabit Network Connection
CODAC Core Sys- tem	5.3.0

3.2.3 Physical Installation

The test rig for this experiment consists of a Gigabit switched network. The centerpiece of the setup is therefore a Cisco® Catalyst 2960 Series Network switch. The switch was not modified softwarewise, being loaded with the default settings. Plugged to this switch are the two miniCODACs and the ODROIDs. To ensure the proper safety and organization, the ODROIDs are disposed in three stacks of five, assembled in a structure of a PC case. This structure also includes four independent power supplies, of which, three of them, supply each set of five ODROIDs. This structure is shown in Figure 3.2.

All the Ethernet cables are Category 6 (CAT 6) with minimized crosstalk and noise.

3.3 Test Methodology and Execution

The underlying principle for this set of tests is that a master (miniCODAC1) creates a SDN topic that the slaves subscribe to and reply. A more detailed explanation of what a topic is and the subscription mechanism is provided in section 3.4.1. During the operation of the real plant system, the ECPC will, for instance, send a start/stop command or inquire all the subsystems about their state. The test plan mimics this procedure, being the payload of the transmission (which the composition is defined in a topic) configurable. Consequently, each subsystem has to reply to the master accordingly. This time, the communication should not be one-to-many (i.e. multicast) but rather one-to-one (i.e. unicast), as each individual slave sends its own message back to the master. Figure 3.3 provides an illustration of



Figure 3.2: Structure with the ODROIDs as part of the test rig. In this picture one can identify the PC case, the 4 independent power supplies and wiring, the 3 stacks of 5 ODROID single board computers each and the CAT 6 Ethernet cables going out of the case. The pieces of paper visible on the photo identify the ODROIDs by their IP addresses.

the basic idea behind the tests.

The focus of these tests is to measure the round-trip time for the communication between the two PCs, while there is a variable stress on the network produced by the parallel communication with the ODROIDs. In an analogy with a ping-pong game, the master serves (ping) and the slaves return (pong). Given this analogy, the developed software (see section 3.4) for the master is sdn-ping while the slaves run sdn-pong.



Figure 3.3: Illustration of the basic principle behind the experiment. Left: In the simulated scenario, the master represents the ECPC and the slaves gyrothrons or other auxiliaries of the ECRH system. The master transmits to every slave a message, using UDP multicast. This message is structured as a SDN topic, of which the master is the publisher and all the slaves subscribe to. Right: in the test rig, the master is miniCODAC1 and the slaves are miniCODAC2 and the array of ODROIDs. Having received the message the slaves process and reply directly to the master using UDP unicast.

3.3.1 Test Plan

The tests execute the following procedure, changing the number of slaves, the payload size and the imposed delay on the slaves, between receiving the master message and sending the reply.

Test 0 – Control

Test 0 is the control test – there are no ODROIDs involved; the network consists only of the two miniCO-DACs, the master and the slave. A baseline roundtrip latency will be determined, at different publishing frequencies. Table 3.3 shows the configurations for this test, characterized by the publishing frequency, number of iterations (sent packages) and consequent test duration.

Table 3.3: Configurations for the control test (test 0). Tests 0-1 to 0-4 have one million iterations at an increasing publishing frequency, test 0-5 repeats the 1 kHz test with ten times more iterations.

Test id	Frequency (Hz)	Iterations ($x10^6$)	Duration (x1000s)
0-1	100	1	10
0-2	500	1	2
0-3	1000	1	1
0-4	2000	1	0.5
0-5	1000	10	10

Test 1 - No Delay

In this test the master will be publishing at a set frequency and all the slaves will subscribe and respond to the master using unicast. There is still no delay on any of the slaves but, unlike Test 0, the ODROIDs are online. The formation of two characteristic response patterns is therefore expected: from the mini-CODAC and the ODROIDs. With this setup (see Figure 3.3), the effect of the response payload size will also be tested, by having a test in which the payload size exceeds the expected payload for regular SDN communications (1192 B instead of the usual 200 B). If the results show unexpected or non-standard behavior that might indicate a correlation with the payload size, further testing shall be conducted.

Table 3.4 shows the configurations for this test, characterized by the publishing frequency, number of iterations (sent packages) and consequent test duration, as well as the payload size.

Table 3.4: Configurations for the no delay test (test 1). Tests 1-1 sets a reference with the miniCODAC and the 15 ODROIDs replying with no added delay and a regular payload size. Test 1-2 doubles the publishing frequency, test 1-3 increases the payload size and test 1-4 has fewer slaves.

-	Test id	Frequency (<i>Hz</i>)	Iterations (x10 ⁶)	Duration (x1000s)	Response payload size (B)	Number of ODROIDs
	1-1	100	1	10	200	15
	1-2	200	1	5	200	15
	1-3	100	1	10	1192	15
	1-4	100	1	10	200	9

Test 2 – miniCODAC Over Concentrated Traffic

In this test the master will be publishing at a set frequency and all the slaves will subscribe and respond to the master using unicast. The miniCODAC slave will have a fixed delay from the time the message is received to the actual response (see Figure 3.4). This delay value will be determined after the analysis of the results of Test 1. The objective is to simulate a transmission during, before, and after an heavy traffic window. Therefore, the delay on the miniCODAC slave is adjusted to a value that, according to the previous experimental data, best replicates this conditions. Table 3.5 shows the configurations for this test, characterized by the publishing frequency, delay value, number of iterations (sent packages) and consequent test duration.



Figure 3.4: Illustration of the basic principle behind test 2. Left: The master transmits to every slave a message, using UDP multicast, structured as a SDN topic. Center: the miniCODAC slave performs a busy sleep for a set time, while the ODROIDs naturally take more time processing and transmuting. Right: all slaves reply directly to the master using UDP unicast.

Table 3.5: Configurations for the test with concentrated traffic (test 2). The delay value is the sleep time on that machine between the reception of the package from the master and sending the reply. These tests aim at timing the miniCODAC reply to occur simultaneously, before and after the bulk of the ODROID traffic window.

Test id	Frequency (<i>Hz</i>)	Iterations (x10 ⁶)	Duration (x1000s)	miniCODAC2 delay (µs)
2-1	100	1	10	420
2-2	100	1	10	350
2-3	100	1	10	520

Test 3 – miniCODAC Over Distributed Traffic

In this test the master will be publishing at a set frequency and all the slaves subscribe and respond to the master using unicast each at a different fixed delays from the time the message is received to the actual response. Both the miniCODAC and the ODROIDs are delayed (see Figure 3.5), as to ensure the miniCODAC transmission is made in a time frame where the ODROID traffic is spread out. In Table 3.5 *n* represents the ODROID id, where the miniCODAC slave has id = 0, thus each consecutive ODROID is set with a delay of 10 μ s more than the previous. The table shows the configurations for this test, characterized by the publishing frequency, delay value, number of iterations (sent packages) and consequent test duration, as well as the delay values for the miniCODAC and ODROIDs.



Figure 3.5: Illustration of the basic principle behind test 3. Left: The master transmits to every slave a message, using UDP multicast, structured as a SDN topic. Center: the miniCODAC slave performs a busy sleep for a set time, while each ODROIDs waits for incrementally longer time. Right: all slaves reply to the master using UDP unicast.

Table 3.6: Configurations for the test with distributed traffic (test 3). The delay value is the sleep time on that machine between the reception of the package from the master and sending the reply. n stands for the index of the ODROID ($n \in [1, 15]$). Test 3-2 acts as a control.

Test id	Frequency (<i>Hz</i>)	Iterations $(x10^6)$	Duration (x1000s)	miniCODAC2 delay (µs)	ODROIDs delay (µs)
3-1	100	1	10	420	10x <i>n</i>
3-2	100	1	10	420	_

3.3.2 Preparation of the Computers for the Experiment

ODROIDs

The ODROID-C2 single-board computers run a Ubuntu 16.04LTS Linux distribution with Mate desktop environment, available at the official ODROID website [23]. The available image should be flashed to a Micro-SD card (16GB, SanDisk used) using the recommended tools. The C2 model used has a Quad Core CortexTM-A53 1.5GHz 64bit ARMv8 processor. In order to ensure the precision required for the experiments, the processor cores should be isolated and running at a constant frequency. To achieve that:

• The boot argument isolcpus should be passed to the bootloader by modifying the file media/boot/boot.ini in the SD card. The option isolcpus 1-3 should be appended to the boot arguments:

setenv boot args "(...) isolcpus=1-3"

(line 135 by default). This specifies that while cores 1 through 3 (indexed at 0) are reserved for the test scripts operations while core 0 remains available for general system interrupts. A reboot is required for changes to take effect.

• The CPU governor should be set to userspace in order to manually control the CPUs frequency and this frequency set to one of the available options. The command

\$ cpufreq-info

displays the available CPU running frequencies and

```
$ sudo cpufreq-set -g userspace
$ sudo cpufreq-set -f 1000MHz
```

sets the governor and frequency, respectively. Re-running the first command should display an increasing usage of the set frequency while none in the other frequencies. This commands should be ran every time the system reboots.

• For the tests, the IP address must also be set manually, editing /etc/network/interfaces. The last byte of the address was set according to the ODROID id as 100 + id (101, 102, ..., 115).

Since the ARMv8 processor does not allow the direct reading of the High Resolution Timer (HRT), the timing for this test is done using the Generic Timer of the CPUs. In order to do that, though, a driver has to be configured, enabling the reading of the registers. Since the ODROIDs installation lacks the linux sources, this driver has been cross-compiled for the Linux kernel installed in the ODROIDs, following the steps in Annex A.

In order to run this test with the SDN library, some additional software has to be installed. The file frommaster.zip has to be copied to and unpacked on each ODROID. Inside, there are two scripts

that run the necessary commands to install (on the first run) and configure (to be run after each reboot): install_ODROID and startup_ODROID. The first script sets the CPU governor and frequency as describes before, installs libxml2 library and its dependencies, compiles the SDN library and sets the necessary environment variables. The latter differs from the former by not installing the libraries.

miniCODACs

The miniCODAC machines, supplied by IO are equipped with a Intel® Xeon® E5-2418L CPU with 8 cores with core virtualization running a MRG Real-time (SMP PREEMP RT) operating system using the Linux kernel 3.10.0-514.rt56.210.el6rt.x86_64. From the standard Red Hat distribution installed on these machines, the MRG can be installed by running:

```
$ cd /etc/yum.repos.d/
$ wget http://ftp.scientificlinux.org/linux/fermi/slf5x/x86_64/RPM-GPG-KEYs/
RPM-GPG-KEY-cern
$ rpm --import RPM-GPG-KEY-cern
$ wget http://linuxsoft.cern.ch/cern/mrg/slc6-mrg.repo
$ yum groupinstall 'MRG Realtime'
```

In order to improve the real-time performance, the core virtualization must be stopped. In these machines, that option was active and locked in the BIOS (Basic Input/Output System). Therefore, in order to overcome this problem, the virtualized cores were disabled in the boot parameters. In order to prevent unintended changes to the CPU working frequency, the power management options were also disabled. Most importantly, the cores 1 trough 3, to be used by the programs, were isolated. Additionally, the clock source was set as the Time Stamp Collector (TSC). Since two machines are not identical, the following steps differ from one machine to the other, mainly because miniCODAC2 does not accept the limitation to 8 CPU cores put in place to nullify the virtualization (locked on in the BIOS) and the shutting down of the power management utility.

The boot arguments should be modified to ensure a optimization of the tests. This is done be appending the following to /boot/grub/menu.lst.

On miniCODAC1:

```
isolcpus=1-3 intel_idle.max_cstate=0 processor.max_cstate=0 idle=poll selinux=0 maxcpus=8
clocksource=tsc tsc=reliable acpi=off
```

On miniCODAC2:

isolcpus=2,3 intel_idle.max_cstate=0 processor.max_cstate=0 idle=poll selinux=0 maxcpus=8
clocksource=tsc tsc=reliable

A reboot is necessary for these changes to take effect. In order to ensure an optimized segregation of the cores the utility **tuna** is run with the following arguments:

On miniCODAC1:

```
$ tuna -c 1,2,3 -i
$ tuna -q *eth1-* -c 1,2,3 -x -m -p FIF0:99
On miniCODAC2:
$ tuna -c 2,3,8,9,10,11 -i
$ tuna -q *eth1-* -c 2,3,8,9,10,11 -x -m -p FIF0:99
```

This instructions segregate cores to be used and then moves and spreads among them all threads involving the network interface to be used in the testing (eth1 in this case), while raising their priority in the scheduler. In order to stop lock the CPIU frequency, one must also run the following command:

\$ sudo service cpuspeed stop

3.4 Test Implementation Tools

In order to perform the described tests, custom software had to be written. Essentially this means one program running on the master and another on the slaves. These programs were written in C++ and use the SDN software in the form of the libraries sdn-base.h and sdn-api.h.

3.4.1 SDN Software

The SDN software is part of CODAC Core System (CCS) and is therefore available on on the miniCO-DAC machines, which come with CCS preinstalled. Nevertheless, the SDN software was updated to the version 1.0.7_nonCCS. The SDN software was also installed on the ODROIDs, integration in the ARM architecture was not a problem. The SDN software is used by CODAC and Plant System Instrumentation and Control (I&C) application software to interface to, and communicate in a homogeneous way over SDN. It provides the necessary functions to structure and support communications over the SDN. It is also featured with monitoring, notification and logging mechanisms to support fault detection, isolation and investigation as well as some other useful structures and routines, compatible with multi-threading applications (*thread safe*). The SDN software package also includes examples, of which some were used as the basis for the developed programs. This ensures that the code is written in a similar fashion, using the same mechanisms, both for the SDN transmission and for auxiliary functions (buffer implementation, isolation routines, etc.). It also ensures that the code is similar in performance to the actual codes to be used in ITER and that these test are supposed to simulate.

SDN Topics

SDN communication is performed using UDP, meaning all the necessary checks and security features must be implemented in the SDN software. This is specified in the requirement SDN-SRS-F-001 – Anonymous publish subscribe paradigm in [22], which also imposes a loose coupling between publisher and subscribers. The way this is achieved is through the definition of SDN topics. A topic characterizes

the message to be sent via UDP. Subscribers do not bind to a publisher, but rather register to a topic, which in itself encodes (explicitly or implicitly) a particular multicast address. In the same way, a publisher addresses a topic rather than any specific receiver. A subscriber has no indication of who (if any) the publisher is.

The SDN packets include a fixed header, limited to the minimum information required at the receiver to perform sanity checks on the topic [22]:

- SDN protocol version, to be able to discriminate between SDN and unforeseen traffic and support interoperability across versions of SDN;
- Topic version number or a hash key derived from topic definition, to ensure participants encode/decode serialized data in a consistent manner;
- Topic instance number, to detect missing packets; and
- Sender timestamp to allow for detecting late packet arrival.

The topic, including header information and the SDN data, can be loaded to the programs trough a XML topic definition file. In the programs developed for the tests, such files were created, and copied to all machines. The following code shows the topic definition file for the topic published by the master and for one of the slaves responses: payload-64-odroids.xml and slaveResponse0.xml, respectively:

Listing 3.1:	payload-64-odroids.xm	۱l
--------------	-----------------------	----

Listing 3.2: slaveResponse0.xml

```
<?xml version="1.0"?>
<!-- Implicit size -->
<topic name="slaveresponse0" version="1.0">
<attribute rank="0" name="Counter" dataType="uint64"/>
```

```
<attribute rank="1" name="TimeStamp" dataType="uint64"/>
<attribute rank="2" name="CommandSlave0" dataType="uint64"/>
<attribute rank="3" name="SlaveId" dataType="uint8"></attribute>
</topic>
```

These topic definition files set the name of the topic, the version and the payload composition (attributes). The size is in both cases implicit from the composition of the payload plus the fixed header size. On the master topic, the multicast address is configured explicitly. If it was not or would the other topic be multicasted this address would be generated automatically using the metadata from the topic definition itself.

3.4.2 Test Programs

In order to execute the tests, several programs had to be custom made: a program to run in the master miniCODAC and another for the slaves, script to manage installing and running the test plan on all the ODROIDs, and data analysis tools. All the software was written in C++ and designed with a command line interface and all the configuration is done using command-line flags and arguments. Both test programs have to send and receive SDN packets. This is a achieved with resource to two classes on the SDN API: publisher and subscriber.

sdn-ping

The role of the master in this experiment is to: (i) publish the topic to all slaves, (ii) receive the replies and compute the delay, (iii) perform some basic statistics. The program sdn-ping performs all this functions making use of multithreading and CPU core segregation capabilities.

The program makes use of n + 2 threads, being n the predefined number of slaves: the main thread for the publishing, one thread for statistics and n for the subscribers.

Table 3.7 shows the command line options for the execution of the program. Some of these options relate to the test plan (publishing frequency, number of slaves, ...) while others are technical, like correction to the CPU frequency or the CPU core affinity.

The publisher is one of the key elements of the code. The following code block shows the publishing mechanism. Relevant features are: (i) the publisher only starts when all the subscribers are set; (ii) performs a busy sleep; (iii) sends in the payload of the SDN packet the time in units of its timer, as to ensure that all time computations are made in the master and with the as little processing as possible.

Listing 3.3: Sniplet of sdn-ping7.cpp: Publishing mechanism.

```
while(run<N_SLAVES); //wait for server thread setup
log_info("Start_publisher");
printf("Start_publisher\n");
uint64_t i=0;
uint64_t till_time;
```

		1			
Option		Argument	Description		
-h	help		Print usage.		
-v	verbose		Verbose mode, measurement data is printed on stdout.		
-s	stats		Enable real-time statistics and exports raw and processed data.		
-0	odroids	n	Number of slaves (ODROIDs or PCs). Defaults to 6.		
-a	affinity	core_id	Run threads on core_id and the next two consecutive CPU cores, defaults to 1.		
-c	count	sample_nb	Stop after sample_nb are published, defaults to 10.		
-f	cpufreq	freq	CPU or timer frequency in Hz. Defaults to 24MHz for aarch64 and 2.2GHz for amd64.		
-i	iface	iface_name	Use <pre>iface_name</pre> as SDN interface, defaults to the default SDN in- terface if exported as an environmental variable.		
-m	mcast	mcast_addr	Publish in a UDP/IPv4 multicast address of the form 'ip_addr:port', e.g. '239.0.0.1:60001'.		
-u	ucast	ucast_addr	Subscribe to a UDP/IPv4 unicast address of the form 'ip_addr:port', e.g. '127.0.0.1:60001'.		
-р	period	period_ns	Publication period in <i>ns</i> , defaults to 100000000 (1Hz).		
-b	binwidth	width	Bin width for statistics histograms, in ns.		
-tp	pubtopic	topic_name	Publish to topic_name, defaults to 'payload-64-odroids'.		
-ts	subtopic	topic_name	Subscribe to topic_name, defaults to 'slaveResponse'.		

Table 3.7: Description of the terminal arguments for sdn-ping.

```
uint64_t t0;
uint64_t periodTicks=CPU_FREQ*1e-9*period;
t0 = readTimer();
till_time = t0;
while ((_terminate != true) && (i < count)){</pre>
       //WAIT
       while (t0<till_time){</pre>
               t0=readTimer();
       }
       till_time += periodTicks;
       p_topic->SetAttribute((char*) "TimeStamp", (void*) &t0);
       p_topic->SetAttribute((char*) "Counter", (void*) &i);
       if (pub.Publish() != STATUS_SUCCESS){
               log_warning("Unable_to_publish_on_'%s'", iface_name);
       }
       i += 1;
       if (verbose) printf("PUB<sub>L</sub>:__sent<sub>L</sub>%lu_pubTime:__%lu\n",i,t0);
```

Each subscriber is launcher in a separate thread, and all these threads are assigned to core_id+1, being core_id the core running the main thread. The subscription mechanism works as follows:

```
//Client LOOP
run++;
while (_terminate != true ){
       if (sub.Receive() != STATUS_SUCCESS){
              /* Blocking receive */
              if (verbose) fprintf(stdout, "%"_WARNING_-Unable_to_receive_on_'%s'\n",

→ p->iface_name);

              log_warning("Unable_to_receive_on_'%s'", p->iface_name);
              continue;
       }
       /* Reaching here, the message has been received */
       itd.subTime = readTimer();
       sub.m_topic->GetAttribute((char*) "TimeStamp", (void*) &itd.pubTime);
       sub.m_topic->GetAttribute((char*) "Counter", (void*) &itd.identifier);
       sub.m_topic->GetAttribute((char*) "SlaveId", (void*) &odroidId);
       double tdiff=(double)(itd.subTime-itd.pubTime)/((double)CPU_FREQ)*1000000000;
       if (statistics){
              //TO STATISTICS
              tsd.slaveId = odroidId ;
              tsd.packId = (uint32_t)itd.identifier;
              tsd.latency = (uint32_t)tdiff;
              while (p->statFifo->TryLock()!= STATUS_SUCCESS);
              if(p->statFifo->PushData(tsd)!= STATUS_SUCCESS )
                     printf("ERROR:_PushData_failed\n");
              p->statFifo->ReleaseLock();
       }
       if (verbose)fprintf(stdout, "SUB_:_odroid_id:%3u_time_dif:_%1f_ns\n", odroidId,
           \hookrightarrow tdiff);
       count -= 1;
}
```

Listing 3.4: Sniplet of sdn-ping7.cpp: Subscribing mechanism.

The most relevant parts of the subscriber routine are: (i) tells the publisher it is ready to receive packets by increasing the counter run; (ii) performs a blocking receive; (iii) upon receiving, reads the timer and retrieves the data from the payload of the received packet (publishing time, counter and slave index); (iv)

}
computes the delay, using the read time at reception and the publishing time, that comes in the payload; (v) if the statistics option is active, saves the packet data in a thread safe First-In-First-Out (FIFO) buffer.

The statistics thread has three main functions: (i) save the raw data (straight from the subscriber threads) in a binary file for post processing, (ii) perform a binning for histograms, writing files with the bins and respective counts every 2000^{th} iteration and finally, (iii) computing the average, standard deviation and extremes in real time. The way this is achieved is shown in the following code block:

Listing 3.5: Sniplet of sdn-ping7.cpp: Statistics mechanism.

```
//STATISTICS LOOP
uint32_t i=0;
while(_terminate==false){
       if (p->statFifo->TryLock()== STATUS_SUCCESS){
              if (p->statFifo->PullData(tsd) != STATUS_ERROR){
                    p->statFifo->ReleaseLock();
                     //SAVE RAW DATA
                     fwrite(&tsd.slaveId,sizeof(uint8_t),1,f1);
                     fwrite(&tsd.packId,sizeof(uint32_t),1,f1);
                     fwrite(&tsd.latency,sizeof(uint32_t),1,f1);
                     //PROCESS DATA FOR STATS
                     avg = (avg*i+tsd.latency)/(i+1);
                     sd = sqrt((sd*sd*i+(avg-tsd.latency)*(avg-tsd.latency))/( i+1));
                     if (tsd.latency > max) max = tsd.latency;
                     if (tsd.latency < min) min = tsd.latency;</pre>
                     if (tsd.latency > p->period) printf("STAT:_LATENCY_ABOVE_PUB_

→ ",tsd.slaveId, tsd.packId, tsd.latency);

                    printf("STAT:%8u\tavg:%8lu\tsd:%8lu\tmax:%8u\tmin:%8uu[ns]\r",i
                        \rightarrow +1,avg,sd,max,min);
                    fflush(stdout);
                     //QUICK STATISTICS
                     bin = ((tsd.latency/BIN_WIDTH)*BIN_WIDTH + BIN_WIDTH/2 )/1000;
                    hist_t.put(bin);
                     for (uint8_t j=0;j<N_SLAVES; j++){</pre>
                            if (tsd.slaveId == j){
                                  hist[j].put(bin);
                            }
                     }
                     if (verbose) printf("\n");
                     if (i % 2000==0){
                            hist_t.print(f3);
```

sdn-pong

The role of the slaves in this experiment is to: upon receiving the master communication, copy from the received information the master publishing timestamp to a new SDN packet and then, either wait for a pre-configured delay time, or send the packet right away, using unicast, back to the master. Since all the actions are sequential, there is only one main thread, locked to an isolated CPU core.

This program is ported for both aarch64 and amd64 architectures being able to run in both the mini-CODACs and the ODROIDs.

Table 3.8 shows the command line options for the execution of the program.

Option		Argument	Description			
-h	help		Print usage.			
-v	verbose		Verbose mode, measurement data is printed on stdout.			
-a	affinity	core_id	Run threads on core_id and the next two consecutive CPU cores, defaults to 1.			
-c	count	sample_nb	Stop after sample_nb are published, defaults to 10.			
-f	cpufreq	freq	CPU or timer frequency in Hz. Defaults to 24MHz for aarch64 and 2.2GHz for amd64.			
-i	iface	iface_name	Use iface_name as SDN interface, defaults to the default SDN in- terface if exported as an environmental variable.			
-m	mcast	mcast_addr	Publish in a UDP/IPv4 multicast address of the form 'ip_addr:port', e.g. '239.0.0.1:60001'.			
-u	ucast	ucast_addr	Subscribe to a UDP/IPv4 unicast address of the form 'ip_addr:port', e.g. '127.0.0.1:60001'.			
-p	period	period_ns	Publication period in ns, defaults to 100000000 (1Hz).			
-tp	pubtopic	topic_name	Publish to topic_name, defaults to 'slaveResponse'.			
-ts	subtopic	topic_name	Subscribe to topic_name, defaults to 'payload-64-odroids'.			

Table 3.8: Description of the terminal arguments for sdn-pong.

Additional Programs Created

Additionally to the test programs some scripts were written in order to execute the test plan and process the data. A collection of bash scripts helps dealing with performing the same action on all the ODROIDs, by reading a file with the IPv4 addresses of the ODROIDs Table 3.9 provides a description of each script and Listing 3.6 shows one of such scripts.

Script	Description
runOdroids.sh	Run the commands provided as argument on all the ODROIDs.
runOdroidsRoot.sh	Run the commands provided as argument on all the ODROIDs as root.
scpToOdroids.sh	Copy files to all the ODROIDs using scp.
scpFromOdroids.sh	Copy files from all the ODROIDs using scp.
runTests.sh	Run sdn-pong with the configuration for Test 1.
runTests2.sh	Run sdn-pong with the configuration for Test 2.
runTests3.sh	Run sdn-pong with the configuration for Test 3.
killOdroids.sh	Kill the test program sdn-pong and screen on all the ODROIDs.

Table 3.9: Description of the bash scripts created to automate the ODROIDs usage.

Listing 3.6: Bash scipt runTests3.sh.

For the post-processing of the data the following programs were written (Table 3.10):

Table 3.10: Description of the supporting programs for the experimental data analysis.

Script	Description
SD.cpp	C++ program to run through the binary file containing one experiment's data and outputing the statistical results. Allows applying some correction factor to the values.
binning.cpp	C++ program to run through the binary file containing one experiment's data and perform a custom binning of the latency values in order to plot histograms.
timeDependance.cpp	C++ program to run through the binary file containing one experiment's data and export a text file with a smoothed trend of the latency over the experiment duration.
difference.py	A Python script to get the total ODROID traffic histograms by subtracting to the total traffic binning, the counts corresponding to the miniCODAC.

3.5 Test Results and Analysis

The test plan described in section 3.3 was executed over a spawn of a week following the test order and with minimal tweaking on the software (not influencing the test results). All the control and monitoring of the execution of the tests was done in the master machine – miniCODAC1. The procedure to start these tests is:

- run sdn-pong on miniCODAC2 (slave),
- run the appropriate bash script (see section 3.4.2) to run sdn-pong on all the ODROIDs,
- run sdn-ping on miniCODAC1 (master).

The program sdn-ping is run on the master with the flag -s to enable real-time statistics. Therefore, when the execution of the test is complete, the logs are saved and copied to archive.

In this section, the statistical results are presented and discussed, following the same structure of the test plan. The value measured is the round trip delay, measured as the time between the publishing of the topic and reception of the individual slave response. The measurement of this time is performed using the master's High Resolution Timer, in independent threads, running on isolated cores. The analysis is based on the average and Standard Deviation (SD) of this values, as well as by the histograms with $20 \ \mu s$ bins, unless explicitly stated otherwise.

The round trip delay is a direct measurement of the latency in the SDN transmission, while the standard deviation relates to the jitter.

3.5.1 Test 0 – Control

Test 0 is the control test with only the miniCODADs communicating with one-another. Table 3.11 shows the statistical results. The values for the mean round trip delay are higher than the requirement for ITER, nevertheless there are two key differences between this test setup and the ITER conditions: the network switch and the fact that a 1Gbit network is used instead of a 10 Gbit.

The standard deviation values are one order of magnitude lower than the maximum-minimum difference. As a consequence, the distribution of these values, shown in the histograms in Figure 3.6 are very sharp.

Table 3.11: Test 0 statistical results for the round-trip measurements. Tests 0-1 trough 0-4 have a increasing publishing frequency, while test 0-5 is a repetition of the test at $1 \ kHz$ (0-3) with 10 times more statistical elements.

Test id	Frequency (<i>Hz</i>)	Mean (µs)	Standard Deviation (μs)	Maximum (µs)	Minimum (μs)
0-1	100	119.1	2.8	248.5	101.2
0-2	500	117.8	2.5	227.6	97.9
0-3	1000	116.5	1.7	225.9	99.8
0-4	2000	116.1	2.3	241.4	93.4
0-5	1000	117.5	2.4	244.1	89.5



Figure 3.6: Histogram with the results of the test 0 for the study of the publishing frequency influence on the round-trip delay. Red: 100 Hz (test 0-1), green: 500 Hz (0-2), blue: 1 kHz (0-3), magenta: 2 kHz (0-4). A slight dependence of the round-trip delay with the publishing frequency is visible. High frequency peeks also appear sharper.

A small dependence of the latency with the publishing frequency is also noticeable in the decreasing

mean and mode (see Figure 3.6). This correlation is unexpected but very small, as the difference between the averages are in the order of σ . The publishing frequency itself should not influence the latency, but in this test plan the experiment time is also dependent on the publishing frequency, as the packets are sent consecutively. To see if there is any major alteration of the latency overtime, the round trip delays were plotted as function of the experiment time (see Figure 3.7). In this plot, the line is smoothed by taking a moving average. We can observe that there is no clear slope, as in, the latter packets having the highest latencies. Nevertheless there is one phenomenon influencing the latency at the experiment time timescale. The smoothed line shows that the round trip times average to one of two bands, with little oscillation (on average) but quite distant between themselves. This phenomenon lasts seconds and is most probably caused by the behavior of the switch or the network cards. Regarding the dependence of the latency with the publishing frequency, it can be a mere expression of the master's internal clock skew.



Figure 3.7: Round-trip delays in test 0-1 (100 Hz) over time, smoothed by a moving average to improve readability. Figure 3.6 shows (in red) the equivalent histogram. The measured round-tip values distribute themselves around two distinguishable values: around 118 and 123 μ s, also visible on the histograms.

3.5.2 Test 1 – No Delay

Test 1 aims at establishing the ODROIDs traffic profile and how it influences th miniCODAC distribution. Table 3.12 shows the statistical results. The first thing that is noticeable in the table is that the average increased over the test 0, even though the ODRODs' traffic is not simultaneous with the miniCODAC's. Figure 3.8 shows the histograms of the tests with 0, 9 and 15 ODROIDs and a correlation of the latency with the number of ODROIDs is clear. While apparently unexpected, the reason behind this phenomenon may be the tests software. The program sdn-ping generates one thread for each subscriber but all these threads are bound to the same CPU core. It is then up to the CPU scheduler to manage the load between the threads, and therefore, the more subscribers online, more delay is expected.

Table 3.12: Test 1 statistical results for the round-trip measurements. Results shown for the miniCODAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15). Tests 1-1 is the control test with 15 ODROIDs, 100 Hz publishing frequency, and small payload size. Test 1-2 has a 200 Hz publishing frequency, test 1-3 a larger payload size and test 1-4 only 9 ODROIDs online.

Test id Replier		Mean (µs)	Standard Deviation (μs)	Maximum (µs)	Minimum (μs)
1-1	0	123.8	2.9	250.0	104.6
	1-15	562.2	22.8	1040.3	125.1
1-2	0	122.1	2.9	359.1	94.8
	1-15	558.2	22.8	932.4	115.1
1-3	0	138.5	2.7	436.8	119.5
10	1-15	611.7	54.6	989.4	138.7
1-4	0	121.0	2.6	251.1	102.0
	1-9	536.0	13.7	1098.8	126.7

Regarding the publishing frequency influence, the histograms in Figure 3.9 show the same behavior already analyzed in test 0.

Figure 3.10 shows what happens to the results when the payload size of the replies is increased. On the miniCODAC reply pattern we can see that the latency has increased, as expected. On the ODROIDs, there is an alteration of the reply profile. Under bins of $20 \ \mu s$ (less than half of the SD) this appears in the histogram as a set of peeks.



Figure 3.8: Zoom of histogram of the round-trip delay values for three different number of ODROIDs online: 0 (test 0-1, red), 9 (test 1-4, green) and 15 (test 1-1, blue). Only a zoom in the region of the miniCODAC peek is shown as the ODROID traffic provide no additional information. All tests have a publishing frequency of 100Hz. A slight dependence of the round-trip delay with the number of ODROIDs online is visible.



Figure 3.9: Histogram of the round-trip delay values of all the slaves for two different publishing frequencies: 100 Hz (test 1-1, red) and 200 Hz (test 1-2, green). The traffic coming from the miniCODAC and from the ODROIDs appear in different bands, being the first distinguishable from the rest. A slight dependence of the round-trip delay with the publishing frequency is visible.



Figure 3.10: Histogram of the round-trip delay values of all the slaves for two different publishing payload sizes: 200 B (test 1-1, red) and 1192 B (test 1-3, green). Both tests have a publishing frequency of 100Hz. The traffic coming from the miniCODAC and from the ODROIDs appear in different bands, being the first distinguishable from the rest. A clear influence of the payload size on the delay values is observed.

3.5.3 Test 2 – miniCODAC Over Concentrated Traffic

Having figured the reply patterns for the miniCODAC and the ODROIDs in the previous tests, test 2 aims at delaying the miniCODAC so that its replies arrive simultaneously (2-1), before (2-2) and after (2-3) the ODROIDs traffic pattern. Table 3.13 shows the statistical results.

Table 3.13: Test 2 statistical results for the round-trip measurements. Results shown for the miniCODAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15). Tests 2-1, 2-2 and 2-3 have the miniCODAC response delayed by 420, 350 and $520 \ \mu s$ respectively.

Test id Replier		Mean (µs)	Standard Deviation (µs)	Maximum (µs)	Minimum (µs)
2-1	0	573.3	13.8	882.4	517.4
	1-15	568.1	25.4	972.1	127.2
2-2	0	469.8	1.7	609.8	452.2
	1-15	558.2	22.8	932.4	115.1
2-3	0	639.7	2.5	1076.5	621.1
	1-15	562.4	21.7	1251.4	124.6

It is noticeable in the histograms in Figure 3.11 that the shape of the miniCODAC traffic peek is similar to the control (red) when the transmission happens just before the ODROIDs (green). It suffers widening when simultaneous (blue) and, since the ODROIDs traffic shape has a "tail" a region with over 100 counts, there is also a widening (to a lesser extent) in the magenta peek. All these results were expected. It is even noticeable on the miniCODAC distribution simultaneous with the concentrated ODROID traffic the formation of two peeks. When competing with the ODROIDs' traffic, the miniCODAC is just one more slave, with a behavior identical to the ODROIDs, as the plot of the delays over time (Figure 3.12) show.

One can also see how the ODROID traffic increases the latency on the miniCODAC transmission by comparing the mean delay with the expected. In test 1-1, under the same conditions but with no delay on the miniCODAC, the mean latency was 123.8 μs . In Table 3.14 a comparison of the expected latency with the obtained is presented, leading us to conclude that the ODROIDs traffic delayed the miniCODAC transmission by an additional 30 μs on average.

Table 3.14: Comparison of the latencies obtained in test 2 with the expected. The expected delay is computed as the sum of the mean round trip delay for the test 1-1 (123.8 μ s) with the intentionally added delay (420, 350 and 520 μ s respectively).

Test id	Expected Delay (µs)	Delay (µs)	Variation (μs)
2-1	543.8	573.3	+ 29.5 (5.4%)
2-2	473.8	469.8	- 4.0 (0.8%)
2-3	643.8	639.7	- 4,1 (0.6%)



Figure 3.11: Histogram of the round-trip delay values of all the slaves for different values of delay on the miniCODAC slave from the time the master message is received to emission of the reply. The packets coming from the miniCODAC slave are represented at full while from the ODROIDs are represented by the outline. In red test 1-1 as a control test with no delay; in green test 2-2 with 350 μ s delay, in order to receive the miniCODAC replies just before the ODROIDs'; in blue test 2-1 with 420 μ s delay, in order to have the miniCODAC replies simultaneously with the ODROIDs'; and in magenta test 2-3 with 520 μ s delay, in order to have the miniCODAC replying just after the ODROIDs'.



Figure 3.12: Round-trip delays in test 2-1 of selected slaves over time, smoothed by a moving average to improve readability. In red the miniCODAC response, in green, blue and magenta, the replies of ODROIDs 5, 10 and 15, respectively. Figure 3.11 shows (in red) the equivalent histogram.

3.5.4 Test 3 – miniCODAC Over Distributed Traffic

Having established the profile of the minCODAC traffic in a fully saturated narrow time band, in test 3 the ODROIDs' traffic is spread over a larger time frame. This is achieved by delaying the miniCODAC by the same amount as the previous test but, instead of every ODROID replying immediately, each successive ODROID has 10 μ s more delay. This way the ODROID traffic is spread out, and the miniCODAC traffic appears in the middle of these transmissions. Table 3.15 shows the statistical results. One can immediately notice that the SD value is larger than that on the equivalent test with no additional traffic (test 1-1, 2.9 μ s) and lower than that in the situation where the ODROID traffic is concentrated (test 2-1, 13.8 μ s). This is an expected result.

Table 3.15: Test 3 statistical results for the round-trip measurements. Results shown for the miniCODAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15).

Test id	Replier	Mean (µs)	Standard Deviation (μs)	Maximum (µs)	Minimum (μs)
3-1	0	542.8	7.7	722.4	514.8
	1-15	580.9	13.3	1039.6	201.5
3-2	0	537.1	1.0	614.7	520.3

Observing the histogram for test 3-1 (Figure 3.13) it is noticeable that the miniCODAC traffic is simultaneous with the ODROIDs, that now distribute themselves in a wider time band, and that the peek is sharper. This is most visible in the histograms in Figure 3.14 where a comparison is made between the three scenarios: no ODROID traffic, in cyan; concentrated traffic, in purple; and distributed traffic, in green.

Figure 3.15 shows the distributions over time of the traffic for a selection of slaves.



Figure 3.13: Histogram of the round-trip delay values of all the slaves with the miniCODAC replies in red and the rest of the slaves in green (test 3-1). The miniCODAC is delayed by $420 \ \mu s$ while the ODROID index *n* has a delay given by $10 \ \mu s \times n$.



Figure 3.14: Histogram of the round-trip delay values for the miniCODAC traffic only. Comparison of the performance with concentrated ODROID traffic, test 2-1, in purple, achieved by a delay of 420 μs on the miniCODAC; with distributed ODROID traffic, test 3-1 in green, achieved by a delay of 420 μs on the miniCODAC and $10 \ \mu s \times n$ delay on the n^{th} ODROID; and in a control test with no ODROID traffic and a delay of the same 420 μs on the miniCODAC, test 3-2 in cyan.



Figure 3.15: Round-trip delays in test 3-1 of selected slaves over time, smoothed by a moving average to improve readability. In red the miniCODAC response, in green, blue and magenta, the replies of ODROIDs 5, 10 and 15, respectively. Figure 3.13 shows the equivalent histograms.

3.6 Conclusions and Possible Improvements to the Test Setup

The conducted tests try to reproduce ITER SDN implementation and operation with maximum reliability but at the same time reproduce with a low cost setup a complex and state of the art network which would otherwise cost thousands of Euros. This compromise means that the network put in place is a Gbit Ethernet network instead of a 10Gbit Ethernet network that ITER shall have. Nevertheless, test 0 showed that the latency requirement of $100 \ \mu s$ latency (round trip) is almost met in this setup, therefore, a network with up to 10 times the throughput should have no problem meeting the requirement. Also, the network switch used is a off the shelf model, with no particular throughput requirements, and the ones used in ITER are switches with a better performance for the SDN (in particular with cut-through capabilities).

Regarding the jitter, test 2 results have shown that even in heavy traffic conditions and with this setup, the experienced jitter is very low. Test 3 confirmed that the presence of other participants in the network is a major factor on both the latency and jitter, changing the expected transmission time spectrum.

This experiment also showed that under heavy traffic, on average, an increase in the latency of 5.4% is to be expected. In absolute terms, under this test setup, this meant a $30 \ \mu s$ additional transmission delay, a figure that is high but still under the $50 \ \mu s$ jitter budget.

Given these results, one can speculate that the 'weakest link' in this setup is the network switch, since the profile of the latency distributions are consistent with experiments conducted at CCFE but the magnitude of the latency is higher. In fact, tests 1-1, 2-1 and 3-1 were reproduced as test 4, with a HP A5500 Series switch, obtaining the results in Table 3.16. This test was performed with only half of the SDN packets sent by the master. The histograms for each test are shown in Figures 3.16 to 3.18.

Table 3.16: Test 4 statistical results for the round-trip measurements. Tests 4-1 trough 4-3 are repetitions of tests 1-1, 2-1, and 3-1 respectively with a different network switch and 500000 packets instead of one million. Results shown for the miniCODAC (replier 0) and the average of the results for the individual ODROIDs (repliers 1-15).

Test id	Replier	Imposed Delay (μs)	Mean (μs)	Standard Deviation (µs)	Maximum (µs)	Minimum (μs)
4-1	0		58.0	8.7	993.3	49.1
	1-15		520.0	25.5	948.8	126.1
4-2	0	420	519.4	22.4	997.4	467.6
<u>-</u>	1-15		524.1	27.0	926.6	150.6
4-3	0	520	595.1	12.8	995.0	564.1
	1-15	10× <i>n</i>	562.4	17.9	950.5	222.5

With the new network switch the latency on the miniCODAC lowered to $58 \ \mu s$, a 53% reduction from the value in test 1-1, and thus making it complaint with the ITER requirement. The repetition of the test plan with this switch is a proposal to continue this experiment further.

Another possible addition to this experiment is to create a new test that tries to concentrate the



Figure 3.16: Histogram of the round-trip delay values of all the slaves with no imposed delays (test 4-1). In red the miniCODAC replies and in blue the ODROID replies.



Figure 3.17: Histogram of the round-trip delay values of all the slaves. Imposed delay of $520 \ \mu s$ on the miniCODAC (test 4-2). In red the miniCODAC replies and in blue the ODROID replies.

ODROID traffic even further. Two solutions for this are proposed: (i) synchronization of the ODROIDs using their GPIO (General Purpose Input-Output), (ii) asynchronous traffic generation with the ODROIDs, using a control algorithm to sync the clock of each ODROID with the master.

Connecting all the ODROIDs to one 'master' ODROID, instead of each ODROID sending his reply after a certain delay, the ODROIDs would reply only after the 'master' ODROID sets the line to the high



Figure 3.18: Histogram of the round-trip delay values of all the slaves. Imposed delay of $520 \ \mu s$ on the miniCODAC and $10 \times n \ \mu s$ on the ODROID n (test 4-3). In red the miniCODAC replies and in blue the ODROID replies.

voltage level. This way, the ODROID thaffic is expected to arrive more concentrated to the switch, as the GPIO interface is faster than the Ethernet and the only timer involved is the one of the 'master' ODROID. Figure 3.19 shows a schematic of this proposed test.



Figure 3.19: Illustration of the basic principle behind test the GIPO proposed test. Left: The master transmits to every slave a message, using UDP multicast, structured as a SDN topic. Center: the miniCODAC and a 'master' ODROID perform a busy sleep for a set time, while the remain ODROIDs wait for their GPIO input to toggle state. Right: the ODROID master' toggles the state of the line, sending a go signal for all ODOIDS to reply to the master using UDP unicast.

Another option to mitigate the dependence on the ODROID low precision timers passes by making the ODROID traffic asynchronous. At first glance this would make the replies even more out of sync, but this would allow using the regular SDN packets coming from the master to calibrate a corrective factor to the timer counter on each ODROID. Preliminary tests to this theory were conducted with a linear corrective factor applied to the ODROID timer counter value. With only one linear factor, the synchronization algorithm does not attempt to correct the timers to an absolute time but rather to mitigate

the clock skew (assumed to be linear) in relation to the master. Results of this tests with two ODROIDs, using a Proportional and Integral (PI) control algorithm, show that it is able to reduce the drifting the two clocks experience from each other to a great extent.

Chapter 4

Conclusions

This thesis describes the work conducted on two different but interconnected topics in the scope of the F4E CODAC group. On the magnetics diagnostic, the developed work had a more theoretical and descriptive nature, on a state-of-the-art component of one the largest scientific projects ever put in place; while on the SDN testing there is a stronger practical component, with an innovative test plan put in place from scratch. On both activities, the work conducted was focused on small part of much larger projects, the ITER Magnetic Diagnostics and the ITER ECRH system, integrated in the work of the F4E CODAC group. Both these projects are large deliverables, in respect to budget, human and technological resources involved that are, at the time of writing, on the development phase. This adds to even the most minute component a great deal of responsibility and thoroughness, almost as a piece of a giant jigsaw puzzle.

Regarding the magnetic diagnostic, part of the documentation work present on this thesis was contributed for the Design Description Document (DDD) and was included in the ITER Preliminary Design Review of the integrator (which was jointly held in June by F4E and the ITER central organization). Regarding the most practical part, the tests on the chopping frequency dependence, two major conclusions can be drawn: (i) given the stochastic nature of the phenomena that causes the drift, the sample size can have an influence on the conclusions drawn and should be increased if comparing performance results of different boards; (ii) the chopping frequency can be safely increased in a band up to at least 10 kHz without compromising performance. Given that there is, at the time of writing, still no agreement on which design to follow for the industrial production of the integrators, these conclusions are still relevant assuming that further testing (if not development) is required. Additionally, and only glanced in Chapter 2, the tools developed to automate the result collection for the tests was used by the CODAC group and its partners for the Phase II testing of the integrator boards.

As for the real-time network testing described in Chapter 3, the objectives were achieved. While tests with similar measurements (pinging) are not uncommon, those are usually performed only between two machines or, if in the context of a large network, after the network is fully established. The test bed put in place allowed simulating a frailly large network, at development phase. This was possible to achieve by employing low-cost single-board computers that have the additional benefice of saving floor-space, a

scarce resource in the premises of F4E. While this test rig does not simulate fully the ITER conditions – 1 Gbit network instead of 10 Gbit, and as the hardware to be used in ITER in the future is still not decided/available – the results obtained are promising as, even in these conditions, the distributions of the traffic are very sharp (predictable) and meeting the latency and jitter requirements. It also became evident the importance of the switch used for the performance. Even if the performance figures are ultimately dependent on the hardware, this study proved the reliability of the SDN, that might allow other systems in the ITER to make use of the SDN software and protocol for their communications. This would bring a great deal of homogeneity among the ITER systems, and ultimately bring savings in cost and human resources, as one flexible and safe protocol is used, instead of custom protocols for each system, which might have reliability or safety issues.

Bibliography

- [1] A. Neto, S. Arshad, F. Sartori, G. Vayakis, G. Ambrosino, A. Batista, I. Bas, R. Campagnolo, B. Carvalho, G. D. Magneval, G. D. Tommasi, O. Dominguez, J. Fernandez-Hernando, A. Pironti, S. Simrock, J. Sousa, C. Sterle, A. Vergara, A. Winter, and L. Zabeo. Conceptual architecture of the plant system controller for the magnetics diagnostic of the {ITER} tokamak. *Fusion Engineering and Design*, 96–97:887 890, 2015. ISSN 0920-3796. doi: http://dx.doi.org/10.1016/j.fusengdes. 2015.06.063. URL http://www.sciencedirect.com/science/article/pii/S0920379615300910. Proceedings of the 28th Symposium On Fusion Technology (SOFT-28).
- [2] M. Keilhacker, R. J. Hawryluk, and R. S. Pease. Jet deuterium-tritium results and their implications [and discussion]. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357(1752):415–442, 1999. ISSN 1364503X. URL http://www.jstor.org/stable/55121.
- [3] INTERNATIONAL ATOMIC ENERGY AGENCY. ITER Technical Basis. Number 24 in ITER EDA Documentation Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2002. URL http: //www-pub.iaea.org/books/IAEABooks/6492/ITER-Technical-Basis.
- [4] J. Wesson. *Tokamaks; 4th ed.* International series of monographs on physics. Oxford Univ. Press, Oxford, 2011. URL https://cds.cern.ch/record/1427009.
- [5] G. Vayakis. 55.a0 magnetic diagnostic system design description document. Technical report, ITER, 2011.
- [6] P. Spuig, P. Defrasne, G. Martin, M. Moreau, P. Moreau, and F. Saint-Laurent. An analog integrator for thousand second long pulses in tore supra. *Fusion Engineering and Design*, 66–68: 953 – 957, 2003. ISSN 0920-3796. doi: http://dx.doi.org/10.1016/S0920-3796(03)00382-X. URL http://www.sciencedirect.com/science/article/pii/S092037960300382X. 22nd Symposium on Fusion Technology.
- [7] D. Zhang, X. Yan, E. Zhang, and S. Pan. A long time low drift integrator with temperature control. *Review of Scientific Instruments*, 87(10):105119, 2016. doi: 10.1063/1.4964806. URL http://dx. doi.org/10.1063/1.4964806.
- [8] J. G. Bak, S. G. Lee, D. Son, and E. M. Ga. Analog integrator for the korea superconducting tokamak advanced research magnetic diagnostics. *Review of Scientific Instruments*, 78(4):043504, 2007. doi: 10.1063/1.2721405. URL http://aip.scitation.org/doi/abs/10.1063/1.2721405.

- [9] D. M. Liu, B. N. Wan, W. Z. Zhao, B. Shen, Y. G. He, B. Chen, J. Huang, and H. Q. Liu. Development of an alternating integrator for magnetic measurements for experimental advanced superconducting tokamak. *Review of Scientific Instruments*, 85(11):11E826, 2014. doi: 10.1063/1.4891706. URL http://dx.doi.org/10.1063/1.4891706.
- [10] Y. Xu, X. Ji, Q. Yang, T. Sun, B. Yuan, S. Liang, L. Ren, and J. Zhou. Development of a low-drift integrator system on the hl-2a tokamak. *Review of Scientific Instruments*, 87(2):023507, 2016. doi: 10.1063/1.4940027. URL http://dx.doi.org/10.1063/1.4940027.
- [11] E. J. Strait, J. D. Broesch, R. T. Snider, and M. L. Walker. A hybrid digital-analog long pulse integrator. *Review of Scientific Instruments*, 68(1):381–384, 1997. doi: 10.1063/1.1147835. URL http://dx.doi.org/10.1063/1.1147835.
- [12] A. Werner. W7-x magnetic diagnostics: Performance of the digital integrator. Review of Scientific Instruments, 77(10):10E307, 2006. doi: 10.1063/1.2220073. URL http://dx.doi.org/10.1063/ 1.2220073.
- [13] A. C. Neto, D. Alves, B. B. Carvalho, G. D. Tommasi, R. Felton, H. Fernandes, P. J. Lomas, F. Maviglia, F. G. Rimini, F. Sartori, A. V. Stephen, D. F. Valcárcel, and L. Zabeo. A real-time architecture for the identification of faulty magnetic sensors in the jet tokamak. *IEEE Transactions on Nuclear Science*, 61(3):1228–1235, June 2014. ISSN 0018-9499. doi: 10.1109/TNS.2014.2326336.
- [14] G. Raupp, G. Neu, W. Treutterer, V. Mertens, D. Zasche, T. Zehetbauer, and A. U. Team. Control process structure of asdex upgrade's new control and data acquisition system. *Fusion engineering and design*, 74(1):697–705, 2005.
- [15] B. Penaflor, J. Ferron, R. Johnson, and D. Piglowski. Current status of the upgraded diii-d real-time digital plasma control system. *Fusion engineering and design*, 71(1):47–52, 2004.
- K. Kurihara, I. Yonekawa, Y. Kawamata, M. Sueoka, H. Hosoyama, S. Sakata, T. Ohshima, M. Sato, K. Kiyono, and T. Ozeki. Status and prospect of jt-60 plasma control and diagnostic data processing systems for advanced operation scenarios. *Fusion engineering and design*, 81(15):1729–1734, 2006.
- [17] D. Alves, A. C. Neto, D. F. Valcárcel, R. Felton, J. M. López, A. Barbalace, L. Boncagni, P. Card, G. D. Tommasi, A. Goodyear, S. Jachmich, P. J. Lomas, F. Maviglia, P. McCullen, A. Murari, M. Rainford, C. Reux, F. Rimini, F. Sartori, A. V. Stephen, J. Vega, R. Vitelli, L. Zabeo, and K. D. Zastrow. A new generation of real-time systems in the jet tokamak. In *2012 18th IEEE-NPSS Real Time Conference*, pages 1–9, June 2012. doi: 10.1109/RTC.2012.6418367.
- [18] R. Felton, K. Blackler, S. Dorling, A. Goodyear, O. Hemming, P. Knight, M. Lennholm, F. Milani, F. Sartori, and I. Young. Real-time plasma control at jet using an atm network. In 1999 IEEE Conference on Real-Time Computer Applications in Nuclear Particle and Plasma Physics. 11th IEEE NPSS Real Time Conference. Conference Record (Cat. No.99EX295), pages 175–181, 1999. doi: 10.1109/RTCON.1999.842597.

- [19] A. C. Neto, F. Sartori, F. Piccolo, R. Vitelli, G. D. Tommasi, L. Zabeo, A. Barbalace, H. Fernandes, D. F. Valcarcel, and A. J. N. Batista. Marte: A multiplatform real-time framework. *IEEE Transactions on Nuclear Science*, 57(2):479–486, April 2010. ISSN 0018-9499. doi: 10.1109/TNS.2009.2037815.
- [20] A. J. Batista. F4e prototype of a chopper digital integrator for the iter magnetics. In 2016 29th Symposium on Fusion Technology (SOFT 2016), 2016.
- [21] MDSplus. http://www.mdsplus.org. Online, Accessed Aug 2017.
- [22] B. Bauvir. Sdn software requirements specification. Technical report, ITER-IO, 2014.
- [23] Hardkernel. http://www.hardkernel.com. Online, Accessed Jun 2017.

Appendix A

Driver Compilation Instructions

• Set up the toolchains and compiler:

```
$ sudo mkdir -p /opt/toolchains
$ sudo tar Jxvf gcc-linaro-5.3.1-2016.05-x86_64_aarch64-linux-gnu.tar.xz -C
/opt/toolchains/
$ sudo export ARCH=arm64
$ sudo CROSS_COMPILE=aarch64-linux-gnu-
$ sudo PATH=/opt/toolchains/gcc-linaro-5.3.1-2016.05-x86_64_aarch64-linux-gnu/
```

```
· Get and prepare the kernel:
```

bin/:\$PATH

```
$ git clone --dept 1 https://github.com/hardkernel/linux.git -b odroidc2-3.14.y
$ cd linux
$ vim Makefile
```

modify the version on the Make file so it reads: EXTRAVERSION = 89 (line 4).

• Compile the kernel:

```
$ make odroidc2_defconfig
```

- \$ make -j4 Image dtbs modules
- \$ vim include/generated/utsrelease.h

modify the version so it reads: #define UTS_RELEASE \3.14.79-89 " (line 1);

\$ vim include/config/kernel.release

modify the version so it reads: 3.14.7989 (line 1).

Get and compile the driver:

\$ cd ../ko

\$ make

make sure the kernel path is correct!

• The driver is installed by running the insmod command in each ODROID:

\$ sudo insmod enable_arm_timers.ko