Agenzia nazionale per le nuove tecnologie, l'energia
e lo sviluppo economico sostenibile

Ministero dello Sviluppo Economico

RICERCA DI SISTEMA ELETTRICO

Attività analitiche a supporto del progetto di canali sperimentali per Jules Horowitz Reactor (JHR)

Mauro Cappelli

RdS/2012/137

ATTIVITÀ ANALITICHE A SUPPORTO DEL PROGETTO DI CANALI SPERIMENTALI PER
JULES HOROWITZ REACTOR (JHR)

Mauro Cappelli ENEA

Settembre 2012

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico - ENEA
Area: Governo, Gestione e Sviluppo**,** del Sistema Elettrico Nazionale
Progetto: Nuovo Nucleare da Fissione: Collaborazioni Internazionali e sviluppo Competenze in Materia Nucleare
Responsabile del Progetto: Massimo Sepielli,ENEA

	Sigla di identificazione	Distrib.	Pag.	di
ENEA Ricerca Sistema Elettrico	PAR2011-ENEA-L1B2-021	L	1	8

Titolo

Attività analitiche a supporto del progetto di canali sperimentali per Jules Horowitz Reactor (JHR)

Descrittori

Tipologia del documento: Rapporto Tecnico

Collocazione contrattuale: ACCORDO DI PROGRAMMA Ministero dello Sviluppo Economico – ENEA sulla Ricerca di Sistema Elettrico PIANO ANNUALE DI REALIZZAZIONE 2011

Progetto 1.3.1: Energia nucleare: NUOVO NUCLEARE DA FISSIONE: COLLABORAZIONI INTERNAZIONALI E SVILUPPO COMPETENZE IN MATERIA NUCLEARE, PAR 2011.

Argomenti trattati: Termoidraulica, Meccanica strutturale, Normativa tecnica, Sistemi di controllo, Fisica dei Reattori Nucleari, Neutronica, Schermaggio delle Radiazioni, Metodo Montecarlo

Sommario

Il presente documento descrive l'attività svolta in collaborazione tra ENEA e CEA su temi legati allo sviluppo del nuovo reattore sperimentale Jules Horowitz Reactor (JHR) in costruzione a Cadarache (Francia) da un consorzio internazionale (comprendente anche l'Unione Europea) sotto l'egida dell'OECD/NEA.

L'ENEA partecipa secondo varie forme di collaborazione sia direttamente con i propri ricercatori sia attraverso collaborazione bilaterale con le singole Università italiane.

In questo documento è riportato il risultato della collaborazione già iniziata nelle annualità precedenti tra ENEA, CIRTEN (nella fattispecie, l'Università di Bologna) e JHR per la progettazione di canali sperimentali del costruendo reattore. In particolare, il documento riporta uno studio sull'apporto di potenza generata da sezioni di prova contenenti elementi "attivi" nel caso di spegnimento del reattore in condizioni normali e in condizioni di sicurezza (safety shutdown).

Parallelamente a questa attività, si è iniziata una nuova linea di collaborazione sulla strumentazione e il controllo volta al progetto di nuovi sistemi avanzati da impiegare nelle facilities sperimentali in corso di completamento. L'attività svolta, descritta nei due deliverables allegati, redatti a cura dell'Università dell'Aquila, rappresenta uno studio preliminare per sistemi di controllo avanzati da applicare in reattori di nuova concezione.

Note

Copia n. **In carico a:**

2			NOME			
			FIRMA			
1			NOME			
			FIRMA			
0	EMISSIONE	25/9/2012	NOME	M. Cappelli	P.C. Incalcaterra	M. Sepielli
			FIRMA			
REV.	DESCRIZIONE	DATA		REDAZIONE	CONVALIDA	APPROVAZIONE

Sommario

Descrizione Generale Della Partecipazione Enea Al Progetto Jules Horowitz Reactor (JHR) 3

ALL. 1: Power Transient Analyses For Reflector Experimental Devices During Shutdowns In Jules Horowitz Reactor 6

ALL. 2: Study And Analysis Of Control Systems Implementable By Programmable Logic For Performance And Safety Improvements Of Novel Nuclear Plants 7

ALL. 3: Development of a Simulation Environment for the Analysis of Control System Performance for Performance and Safety Improvements of Novel Nuclear Plants 8

Descrizione Generale Della Partecipazione Enea Al Progetto Jules Horowitz Reactor (JHR)

Il PAR 2011 prevede il prosieguo della collaborazione con CEA nella fase di progettazione del reattore Jules Horowitz per sperimentazioni su materiali e sistemi per retattori nucleari di attuale e futura generazione. ENEA e CIRTEN (Università di Bologna) hanno sviluppato uno studio che si colloca nella partecipazione alla progettazione di canali sperimentali del reattore "Jules Horowitz Reactor (JHR)" , al momento in corso di costruzione presso il Centro CEA di Cadarache. In questo ambito, l'attività ENEA-CIRTEN ha riguardato le simulazioni neutroniche e termoidrauliche a supporto della progettazione di alcuni canali sperimentali dello JHR. Nuove attività potranno riguardare la progettazione di prove sperimentali appropriate per la validazione di codici di calcolo per la sicurezza nucleare.
La ricerca europea si avvale di numerosi reattori realizzati specificatamente per studi sui materiali (Material Testing Reactors, MTR), finalizzati all'avanzamento tecnologico del progetto degli impianti nucleari di potenza. Il reattore Jules Horowitz (JHR) è un rettore di ricerca da 100 MWth del tutto innovativo nello scenario dei reattori sia europei che mondiali: infatti è stato progettato per produrre contemporaneamente un flusso di neutroni veloci nel core del reattore ed un flusso di neutroni termici nella zona circostante il core, realizzata con un mantello di berillio, adeguatamente configurato sia per espletare la funzione di riflettore sia per ospitare apposite sezioni di prova (si veda l'allegato **Power Transient Analyses For Reflector Experimental Devices During Shutdowns In Jules Horowitz Reactor** per una descrizione più accurata del reattore e per informazioni più dettagliate sulle sue caratteristiche).

Ambedue i flussi neutronici raggiungono valori molto elevati (2÷4 x 10^{14} n/cmq*s per il flusso veloce e 5x10^{13} ÷ 5x10^{14} n/cmq*s per il flusso termico): questa caratteristica, unita ad un accurato progetto della zona reattore – mantello, fa sì che nel core e nel moderatore di JHR possano essere alloggiate sezioni di prova che sono anche configurabili come veri loop sperimentali. Esse consentono di simulare, in condizioni termoidrauliche e nucleari reali, situazioni operative di particolare criticità per reattori di IV generazione, con riferimento soprattutto a quelli di interesse per il Sustainable Nuclear Energy Technology Platform in ambito UE (reattori veloci a Sodio, a piombo o piombo-bismuto, a gas). In particolare gli alti flussi neutronici conseguibili consentono di produrre elevati "dpa rate" (8÷16 dpa/year), permettendo di sperimentare, in tempi ragionevolmente brevi, gli effetti degli irraggiamenti su quei materiali che si prevede potranno essere impiegati in reattori di IV generazione .
JHR può ospitare numerose sezioni di prova contenenti anche elementi di combustibile, da provare all'interno del reattore stesso: gli alti flussi tipici di JHR chiaramente inducono, nei combustibili presenti nelle sezioni di prova che li ospitano, potenze termiche lineari consistenti, dipendenti dall'arricchimento del combustibile in prova, ma comunque sempre dell'ordine di parecchie centinaia di W/cm. Infatti un combustibile sperimentale con arricchimento già dell'1% in U 235,quando sottoposto

al flusso di neutroni termici presente nel mantello, può raggiungere una potenza termica lineare di 600 W/cm.
JHR consentirà in definitiva, la raccolta di moltissimi dati sperimentali di importanza essenziale sia per approfondire aspetti riguardante l'esistente famiglia di reattori già operativi sia per avere evidenze sperimentali sulle interazioni dei neutroni con materiali/leghe innovativi, sul comportamento dei combustibili nelle varie combinazioni attualmente in studio (build-up dei gas di fissione, bruciamento degli attinidi ed altro) su processi di trasmutazione etc.
Il presente lavoro, descritto nell'allegato **Power Transient Analyses For Reflector Experimental Devices During Shutdowns In Jules Horowitz Reactor**, si propone di studiare l'apporto di potenza generata da sezioni di prova contenenti elementi "attivi" installati nel reattore per scopi di ricerca, nel caso di spegnimento del reattore in condizioni normali e in condizioni di sicurezza (safety shutdown). Si è supposta pertanto la contemporanea presenza di più sezioni di prova contenenti elementi di combustibile ovvero costituite esclusivamente da specifici elementi di combustibile (è questo il caso di sperimentazioni previste in appositi canali del core e del riflettore.), il cui apporto di potenza dovrà essere tenuto in conto in caso di spegnimento per il dimensionamento del sistema di raffreddamento.
Proprio per la presenza di tali sezioni di prova "attive" che danno un contributo di potenza anche dopo lo spegnimento del reattore è importante definire l'evoluzione della temperatura in caso di shut-down per manovre di normale esercizio o di safety.

Lo studio fa riferimento a progetti di sezioni di prova già definiti ovvero in avanzato stato di definizione, e che quindi potranno verosimilmente essere installate contemporaneamente su JHR.

Le analisi sono state condotte con codice DULCINEE (cinetica neutronica) e TRIPOLI 4.7 Monte Carlo (accoppiamento core-riflettore), ed assumendo alcune ipotesi significative circa la sequenza di inserzione delle barre di controllo.

Una nuova linea di ricerca è stata introdotta in tale collaborazione a partire dalla presente annualità. Una delle ultime fasi del progetto di un reattore nucleare (per la produzione elettrica o a scopo di ricerca) è tipicamente dedicata all'implementazione della strumentazione e dei sistemi di acquisizione dati, controllo e protezione a garanzia della sicurezza.
L'ENEA ha iniziato su questi temi una collaborazione con il JHR, anche attraverso il contributo dell'Università dell'Aquila e del DEWS (Centro di Eccellenza dello stesso Ateneo).
In questa annualità, è stato svolto uno studio preliminare volto all'indagine delle potenzialità di alcuni dispositivi innovativi come gli FPGA (Field Programmabale Gate Array) sui sistemi di comando e controllo dei reattori nucleari di generazione avanzata e sulle principali criticità che si riscontrano nella realizzazione di prodotti hardware e software volti alla mitigazione o risoluzione delle problematiche legati alla sicurezza d'impianto e all'ottimizzazione delle performances rispetto a quanto previsto da progetto.

L’attività ha interessato lo studio e l’analisi dello stato dell’arte dei sistemi di controllo oggi implementati e in via di sviluppo, con particolare attenzione a implementazioni che coinvolgano logiche programmabili. Una volta acquisiti i riferimenti scientifici, si è provveduto a uno studio di fattibilità sulla realizzazione di un sistema di controllo basato su tecnologia FPGA e quindi alla progettazione di un prototipo di tipo *general purpose*.

Tale studio risulta di particolare importanza come valutazione preliminare alla progettazione di un sistema di controllo remoto di tipo DCS (Distributed Control System), in corso di definizione all’interno della collaborazione con il JHR. Tipicamente, il progetto si basa su componentistica “classica” di tipo PLC, con sistema di supervisione centralizzato.

Tali PLC sono distribuiti in zone dedicate alle esperienze (p.e. nella zona dei canali sperimentali) e controllate da una sala controllo che ha il compito di gestirne l’affidabilità in ogni istante.

Tale sistema, che è integrato in una rete di interconnessione ridondata ad alta affidabilità e alto flusso di dati, con server per lo smistamento dei dati e stazioni di supervisione e verifica di correttezza dei dati, potrebbe essere implentato anche in una tecnologia più innovativa come gli FPGA, le cui potenzialità sono in fase di valutazione a livello internazionale per tutte le tipologie di impianto.

L’industria nucleare sta infatti indagando sempre più, negli ultimi tempi, il ricorso a tecnologie più innovative e in grado di garantire una migliore personalizzazione per applicazioni del mondo nucleare. Tra queste nuove tecnologie, si segnalano per qualità e prestazioni gli ASIC (Application Specific Integrated Circuit), i CPLD (Complex Programmable Logic Device) e gli FPGA (Field Programmable Gate Array).

Tali dispositivi condividono la comune caratteristica di programmabilità da parte dell’utente, che consente di personalizzare sia l’hardware che il software alla specifica applicazione richiesta, senza dover passare per hardware e software generali (e spesso sovradimensionati rispetto alle reali necessità), garantendo così tra le altre cose una maggiore immunità alle intrusioni indesiderate contro la riservatezza dei dati.

In tale contesto, rivestono sempre più successo le architetture FPGA, che garantiscono linguaggi di programmazione altamente sviluppati come Verilog o VHDL, capaci di tradurre più agelvolmente funzioni complesse nella tecnologia richiesta.

Con l’intento di indagare le potenzialità di tale tecnologia è stato acquisito un sistema commerciale basato su tecnologia FPGA programmabile tramite linguaggio ad alto livello LabVIEW© (confronta rapporto PAR2011-ENEA-L1B2-014).

ALL. 1: Power Transient Analyses For Reflector Experimental Devices During Shutdowns In Jules Horowitz Reactor

CIRTEN

Consorzio Interuniversitario per la Ricerca TEcnologica Nucleare

UNIVERSITA' Di BOLOGNA

Dipartimento di Ingegneria Energetica, Nucleare e del Controllo Ambientale

DIENCA

Progettazione neutronica e termofluidodinamica di canali sperimentali di reattori di ricerca

POWER TRANSIENT ANALYSES FOR REFLECTOR EXPERIMENTAL DEVICES DURING SHUTDOWNS IN JULES HOROWITZ REACTOR (JHR)

Autori

UNIBO :	**Patrizio Console Camprini** **Carlo Artioli** **Marco Sumini** **Francesco Teodori**
CEA:	**Christian Gonnier** **Bernard Pouchin** **Serge Bourdon**

CERSE-UNIBO RL 1353/2012

BOLOGNA, 31 LUGLIO 2012

Lavoro svolto in esecuzione dell'Attività LP1.B2.a
AdP MSE-ENEA sulla Ricerca di Sistema Elettrico - Piano Annuale di Realizzazione 2011
Progetto 1.3.1 "*Nuovo Nucleare da Fissione:*
collaborazioni internazionali e sviluppo competenze in materia nucleare"

Abstract

European nuclear research takes advantage of several Material Testing Reactors (MTRs) aiming at technological enhancement of power plants design and operation to support industries, utilities and regulators. Jules Horowitz Reactor (JHR) is intended to be the 100 MW MTR that attains the most significant experimental capacity in Europe. It is designed to host several devices and fuel experiments are planned to be performed inside the reflector. Since some fuel samples will be loaded within this area, a power transient evaluation is important to define temperature evolution during shutdowns for safety and reactor operation purposes. The present analysis considers Normal and Safety Shutdown procedures and takes into account different compositions concerning the equilibrium cycle.

Index

Index **3**

1 Introduction **4**

1 The Jules Horowitz Reactor Project *4*

1.1 European Material Testing Reactor for Research and Development 4

1.2 Reactor Core Description 5

1.3 JHR Experimental Capacity 6

2 Research Description **19**

2.1 Thermalhydraulic and Neutronic Core Model 19

2.2 Shutdown Systems for JHR 20

2.3 Core-Reflector Coupling 21

2.4 Reflector Device Calculations 23

2.5 Conclusions and Perspectives 25

3 References **27**

1 Introduction

1 The Jules Horowitz Reactor Project

1.1 European Material Testing Reactor for Research and Development

European policy regarding energy supply and resources availability, as well as infrastructures, is expected to gather different domestic power plant fleets searching more and more for shared scenarios. Growing demand for electrical power and greenhouse gases reduction strategies make nuclear energy a significant source in the European mix, since very low environmental impact and economical competitiveness are achieved.

At present time, several nuclear power plants are connected to the grid supplying around 30% of European electrical needs. Research and development supporting this significant network take advantage of Material Testing Reactors (MTR) which provide experimental data to industries, utilities and regulators. In fact, these facilities are very important to study properties of materials that have to withstand critical thermalhydraulic and radiation conditions during operation.

The French Atomic Energy Commission (CEA) has then launched the construction of a new MTR – the Jules Horowitz Reactor (JHR) - at Cadarache research centre in the framework of an international collaboration.

JHR is intended to become the most important MTR in Europe for the next century. It has been conceived in order to investigate structural material and fuel properties linking industrial and research needs of all partners.

- Sustainability of power supply is also related to energy market and then life extension of power plants is a key point in network management, aiming at reducing capital costs and enhancing competitiveness. GenII nuclear plant lifespan needs several material tests in order to keep the best safety level respecting components qualification procedure. So JHR is equipped with experimental loops able to simulate present PWR, BWR, CANDU and VVER technologies using both a core fast flux and a proper in-reflector thermal neutron spectrum. GenIII reactors deployment phase requires experimental R&D to support and validate material certification during all the long life of this innovative nuclear plant generation. On the other hand JHR allows fuel performances optimization through sample irradiation achieving higher burnup and better resource exploitation.
- Safety topics are investigated by means of accident simulations and component tests. LOCA (loss of coolant accident) scenarios are then reproduced, as well as power transients and ramps, in order to study fuel materials behaviour. In addition, data concerning fuel properties during normal and abnormal operations are provided for utilities and industries.
- High neutron fluxes and elevated temperature loops have been designed in order to reproduce challenging conditions for GenIV reactors research and development. A quite in-core fast neutron spectrum is also capable to reproduce SFR, GFR or LFR neutronic features. Preparing for next innovative GenIV reactors, many experimental data are needed concerning radiation interaction with advanced materials - namely graphite, nickel alloys and ceramics. JHR has been conceived for sample selection and testing in representative conditions about neutron flux and thermal stresses. Ageing effects and mechanical deformation are expected to be reproduced as far as operational irradiation periods are concerned. High in-core DPA (Displacements Per Atom) rates allow important experiments aimed at improvement of cladding and in-vessel component materials such as austenitic stainless steel or ferritic steel (mainly utilized in sodium and lead-cooled technologies). In addition, optimized devices have been prepared to study minor actinides partitioning and transmutation issues. Fission gases build-up investigation is particularly focused on americium burning system analysis.
- Increasing in nuclear medicine diffusion and reliability makes radioisotope production and stock very important and strategic activities in the research reactor domain. JHR is going to be capable to supply from around 25% of European needs in Molybdenum-99 to about 50% in case of particular market demand or critical procurement.

- Nuclear research always needs facilities for education and training of young scientists and engineers. Then JHR international project is also aimed at expertise transfer and enhancement to support future European technology.

1.2 Reactor Core Description

JHR is a 100 MW pool-type light water cooled material testing reactor which has been designed to achieve high experimental capacity. The core rack is a 60 cm height cylinder made of aluminium in which 37 drilled holes can host both 34 fuel elements and 3 sample holders within the so called “large” test positions. On the other hand, “small” test locations are placed in the centre of 7 cylindrical plate fuel elements in order to reach the fast flux as close as possible to the fuel. These 10 experimental slots are available at the same time in order to maximize fast neutron utilization for structural irradiation and high dpa rates.

Two different nominal conditions are envisaged to optimize experimental availability and operative costs: the first one is about 100 MW thermal power as described before and the second foresees some 70 MW. Even a possible twofold core charging is accepted: 37 or 34 fuel elements depending on test needs from international partners.

The operational cycle is about 25 days and a particular U_3Si_2 metallic fuel 27% enriched in U235 is used. The ultimate fuel is expected to be a metallic UMo 19.75% enriched alloy which is still under qualification at present time, nevertheless it is planned to be supplied after the starting phase. The cladding material is an aluminium alloy and every fuel element is composed by 8 cylindrical and concentric plates kept together by 3 stiffeners.

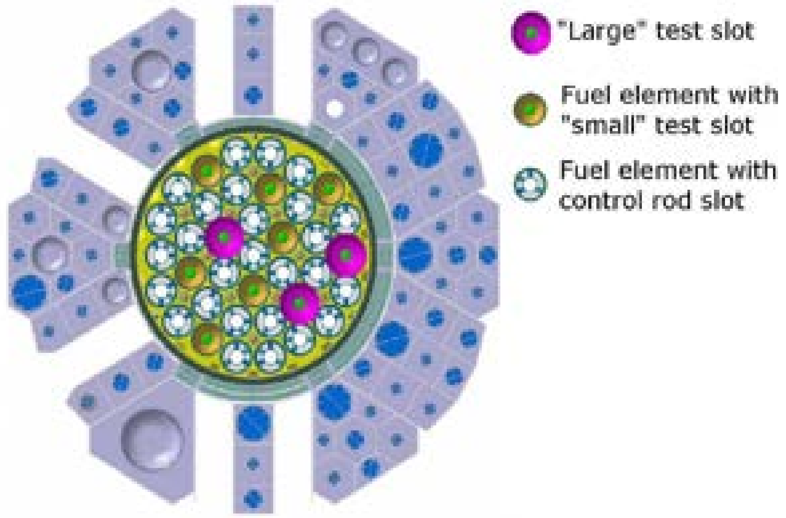

"Large" test slot
Fuel element with "small" test slot
Fuel element with control rod slot

Figure 1) JHR core cross-section

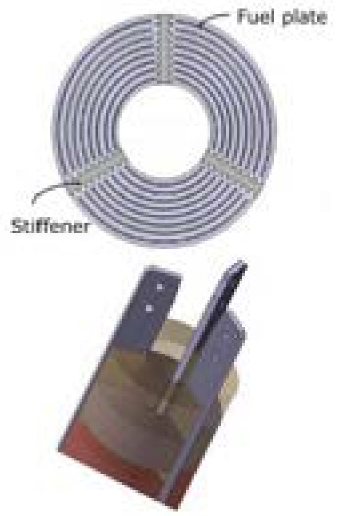


Figure 2) JHR fuel element

Remaining 27 fuel elements not hosting sample holders are utilized for control rods insertion. In the inner part, two cylindrical shell hafnium rods are envisaged to pilot the reactor both to provide poisoning or depletion compensation and to assure safety shutdowns.

Outside the core, a beryllium reflector allows to get a thermal neutron flux suitable for several test concerning fuel properties. Here neutrons coming from the centre of the reactor undergo many collisions and slow down up to representative energies to simulate LWR spectra.

In order to perform analyses of material properties, it is necessary to keep specimen temperature under control. Then a zircaloy shield is placed around half a core to reduce gamma heating in some reflector regions.

1.3 JHR Experimental Capacity

JHR material testing reactor is expected to achieve a relevant position within the framework of European research and development in nuclear technology. In order to cope with this important mission, several experimental devices have been conceived and developed for different kinds of nuclear reactors.
JHR will work under the umbrella of a nuclear facilities network to take advantage of many operative experiences all over Europe and to share important know-how. The design team and several international collaborations have highlighted the most relevant experimental needs of today industries, utilities and regulators as well as the future ones.
The challenge is to create a modern, flexible and integrated user-facility keeping the best safety standards during all the possible experimental procedures. As far as cost optimization is concerned, international market trends have been estimated to match top level performances and reactor management issues.
JHR irradiation capability is essentially twofold. Many material tests are planned to be carried out inside the core thanks to a significant fast neutron flux ranging from 2 10^{14} n/cm^{2}/sec up to 4 10^{14} n/cm^{2}/sec and a high dpa rate (from 8 to 16 dpa/year). This in-core spectrum is suitable for cladding and in-vessel components study. On the other hand, the reflector positions allow a thermal flux ranging from 5 10^{13} n/cm^{2}/sec up to 5 10^{14} n/cm^{2}/sec that may induce linear power up to 600 W/cm in 1% U235 enriched fuel samples designed for fuel property enhancement studies. In addition, even fast flux of some 7 to 8 10^{13} n/cm^{2}/sec is available in some reflector locations.
JHR experimental infrastructure takes advantage of all the facilities designed for reactor support. A large area around the core hosts a *Fission Products Analysis Laboratory* in which on-line measurements - concerning gas and liquid compounds of interest - are allowed thanks to top level instrumentation and a very close location with respect to reactor core. This reduces latency time as far as very short-lived nuclides are concerned and several activation analyses and contamination evaluations are even possible there.

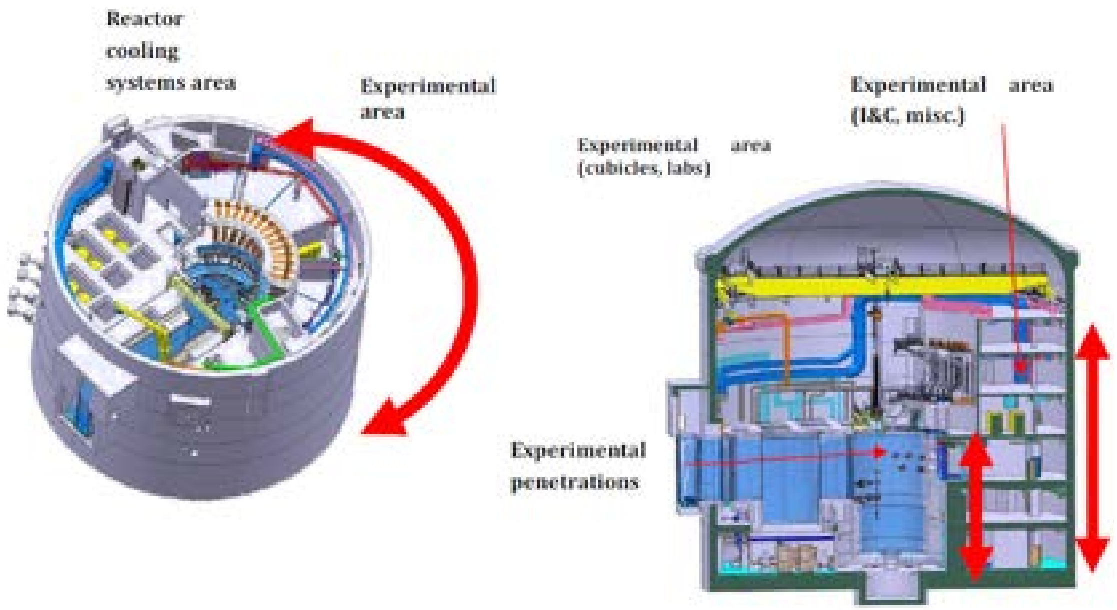

Reactor cooling systems area
Experimental area
Experimental area (I&C, misc.)
Experimental area (cubicles, labs)
Experimental penetrations

Figure 3) JHR experimental areas

A *Dosimetry and Radioprotection Laboratory* is also envisaged in order to assure the best safety levels for all the operators and for further tests which may be carried out in out-of-pile environments.
Corrosion and chemical reactions are induced depending on test needs for what concerns challenging material simulations, then a *Chemistry Laboratory* is planned to support conception and performances of representative sample conditions.
JHR is also equipped with 4 *Hot Cells* to support device management since the first test stages. They are conceived for sample preparation, installation, tuning of test conditions during experiment and finally sample removal and stock. Hot cells for β/γ radiations are planned to be used for fuel test cycle operation as well as waste management. In addition, a α/β/γ hot cell is expected to be used for specific contamination handling –

namely failed fuel elements or samples – coming from devices or reactor core; even cask preparation is allowed.

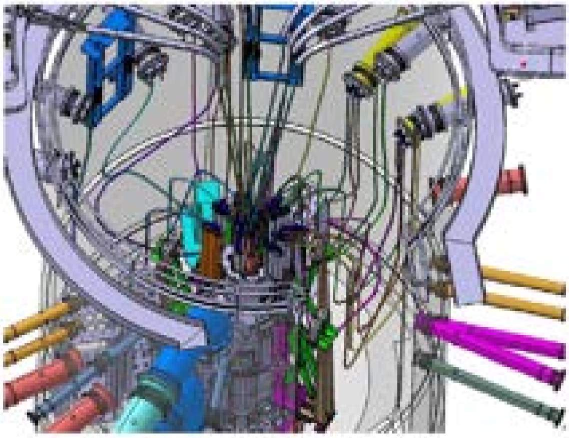


Figure 4) JHR core and device connections

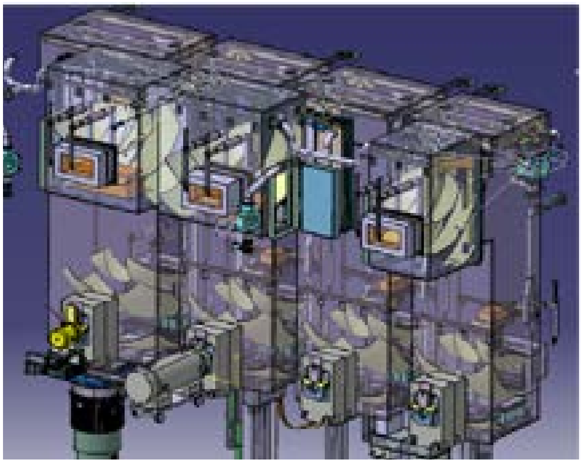


Figure 5) JHR hot cells

Reactor piloting procedures and test staff interactions are optimized; reactor control room and device management instrumentations are designed to get closer and more interactive in order to reduce error likelihoods and improve reactor cycle utilization.

JHR device design involves several important steps starting from experimental needs identifying to the complete construction of the facility. At a first stage the experimental scenario is described in terms of irradiation conditions and possible time-dependent behaviour as well as temperature and pressure fields. Corrosion and chemical representativeness of the test are evaluated as well as mechanical stress or strain loading. Computations of neutronic and thermalhydraulic features are then performed. Cooling issues are considered both in forced and in natural convection, then interaction with core main loop is studied. After a first R&D engineering stage, all necessary equipments are taken into account concerning instrumentation and control during all the test cycle. Possibly, fission product analyses and loop contaminations are implemented and dedicated chemical and radiological devices are supplied. Connections with existing and shared facilities among JHR infrastructures are then considered in order to optimize test procedures. Hot cells and non-destructive examination benches are then integrated and manipulation risks are evaluated. Furthermore, the device design is carried out with respect to safety standards and control system management. In addition, safety analyses are performed in order to evaluate all possible device-core interactions in terms of accident and incident starting events - particularly for what concerns heat evacuation and cooling. In addition, for in-pile parts all barrier tightness is expected to fit with all regulator requests.

Out-of-pile parts, electrical and hydraulic auxiliaries are then integrated with all plant tools. After this design procedure is complete, technical tests are foreseen in dedicated facilities, purchasing and component procurement are planned until device construction and operation.

JHR experimental capability is mainly aimed at representative nuclear power plant conditions. Several reactor technologies are simulated as far as thermal-hydraulic, neutronic and chemical features are concerned (PWR, BWR, CANDU, SFR, GFR, LFR, HTR, VHTR, VVER). Time-dependent power behaviour and ramps are utilized to simulate reactor transients concerning fuel samples. Accidental and incidental conditions are studied for safety purposes and regulator support. On-line radiological and chemical monitoring, as well as induced mechanical constraints, allow modern sample feature tuning and optimize control of data acquisition and quality. All these relevant enhancements in testing capability improve matching between predictive models by multiphysics computations and experimental data.

Core Experimental Devices

Several material tests are planned to be performed within the core thanks to a suitable fast neutron flux. About 10 experimental positions are available there for cladding and in-vessel component alloys such as stainless steel, titanium, zirconium and nickel alloy, aluminium and control rod materials. Some tests are aimed also at chemical corrosion programs under the umbrella of European projects for nuclear material property studies.

Among these 10 test locations, 3 devices replace fuel elements and for those the maximum allowed external diameter is around 94 mm. Remaining 7 positions are in the centre of the innermost fuel plates and can host devices at most 36 mm large.

Design topics have mainly been oriented to challenging on-line multiple strain control and measurement. Material irradiation growth is evaluated by means of samples in which free blades undergo simple irradiation effects and loaded blades exhibit superimposed strains. Provided the same irradiation conditions, it is then possible to separate the effects.

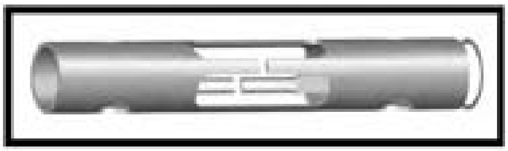


Figure 6) Device for irradiation growth test

Axial stress relief is also evaluated for simple cladding and structural blades at which a previous load is imposed. Superposition phenomena are possibly investigated within gamma and neutron radiation fields. On-line stress and strain control allows even dynamic tensile variations in JHR core. Creep tests regard mechanical behaviour under particular or time-dependent load. Within the framework of innovative experimental R&D, a device capable to measure and control both radial and axial stresses is under development. These two kinds of strain normally depend on geometrical dimensions and inner pressure of a cladding material specimen. Nevertheless, a device capable to set axial and bi-axial strains tuning their ratio is under development. It is expected to increase knowledge about irradiation growth effect on cladding since stress superposition issues will be investigated.

Figure 7) Material strain investigation

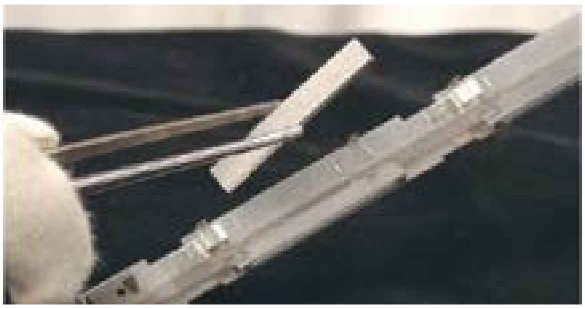


Figure 8) Simple blade stress analysis

Since mechanical properties depend on temperature and radiation conditions, it is necessary to precisely measure and control these parameters. Then several studies have been carried out to better evaluate gamma heating both through experimental tests and thanks to nuclear data enhancement. Loop thermalhydraulics has been optimized to reach suitable flat temperature profiles within the samples. For this reason, liquid metal NaK is utilized at loop temperature of about 450°C and the objective is reached since only +10°C hot spots at instrumentation contacts are present. Technical solution has been proved to be effective since very smooth axial profiles are achieved. Operational temperatures are possibly changed through electrical heaters and electromagnetic pump performances.

Furthermore, corrosion and chemical effects analyses are envisaged since device loops are possibly connected to external filtration and purification systems.

Cladding and internals materials are then analysed ranging from operative to accidental scenario configurations. JHR experimental data will be a relevant reference for selection, qualification, optimization, processing, lifetime assessment, licensing and abnormal operation tests both for safety regulators and industries.

Reflector Experimental Devices

Fuel properties are typically studied taking advantage of thermal representative neutron spectra. In fact within JHR beryllium reflector, both fixed and moving experimental locations allow different kinds of fuel tests. Foremost, it is worth to notice that thermalhydraulic separation of device loops with respect to the core one assures safe and flexible experimental management. So it is possible to reproduce a lot of operational conditions – namely PWR, BWR, and VVER – tuning pressure and temperature parameters of the single cooling loop. HTR environments are created by means of gas circuits properly connected to chemical facilities. In addition, heavy water cooling may allow CANDU operational simulations for HWR reactor technology. JHR is so capable to support GenII and GenIII nuclear power plants as far as fuel optimization, property improvement and resources exploitation are concerned.
Single pin devices have been designed but also multiple sample locations starting from European research needs concerning Calisto loop in BR2 reactor.
Experimental configurations allow steady-state irradiations for both short and long periods, medium power transients characterized by different power ramp slopes are also envisaged. It is even possible to simulate accidental or incidental scenarios thanks to precise regulation of power generation inside the samples.
Fixed positions inside the reflector are utilized to reproduce LWR conditions, CANDU heavy water cooled and fast reactor chemical environments. Some 20 fixed slots are available with external maximum dimension of about 100 mm. One particular position may host up to 200 mm large test devices.
In-reflector slots are designed to host test samples placed on moving structures in order to modify distance with respect to the core. Here corrosion tests are expected to be carried out as well as LWR fuel studies. LOCA accidents may be reproduced as far as loss-of-coolant scenarios are concerned. High temperature reactors are interesting as well for GenIV purposes and they may be simulated even in thermal neutron spectrum conditions.
JHR reflector may host up to 6 moving structures in order to utilize linear powers up to 600 W/cm and to reproduce power ramps from nominal 200 W/cm/min up to 700 W/cm/min just changing sample distance from the core and then flux intensity within the fuel. It is possible to achieve a maximum velocity of some 50 mm/s using a test slot of about 350 mm depth.

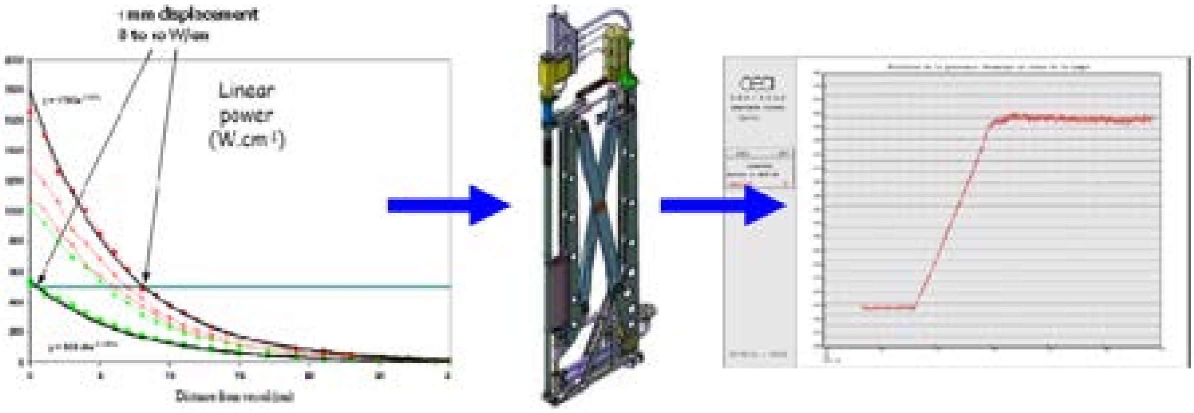


Figure 9) JHR reflector moving structures for power ramps

It is worth to highlight that irradiation in research reactor is a necessary stage for fuel study and development. Foremost, many innovative materials are investigated through neutron radiation doses within small samples. It is now important to create homogeneous irradiation conditions and host as much specimen as possible to compare several samples, even with well-known compounds. New materials can be compared and a first selection is performed.
A second "understanding test" is carried out to obtain a precise evaluation of many physical and structural parameters taking advantage of destructive and non-destructive examinations. Knowledge about mechanical and chemical properties is deepened and operational properties of innovative fuels are then predicted.

The last irradiation concerns effective normal and abnormal operational conditions of the sample. Power ramps and long period depletions as well as abnormal transients are simulated in order to prepare, qualify and license the advanced fuels to be utilized in a nuclear power plant.

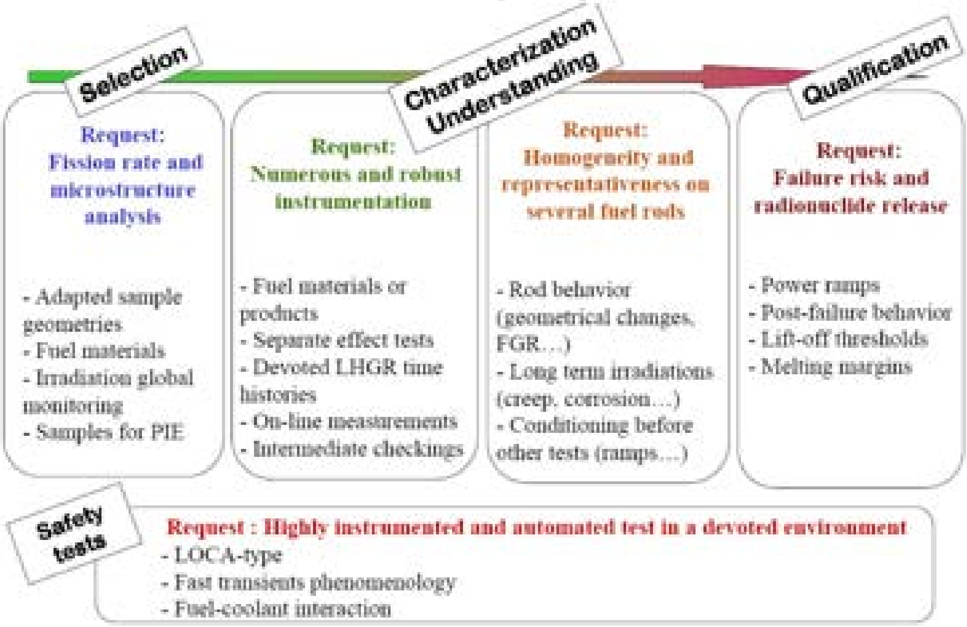

Selection
Characterization
Understanding
Qualification
Request:
Fission rate and microstructure analysis
- Adapted sample geometries
- Fuel materials
- Irradiation global monitoring
- Samples for PIE
Request:
Numerous and robust instrumentation
- Fuel materials or products
- Separate effect tests
- Devoted LHGR time histories
- On-line measurements
- Intermediate checkings
Request:
Homogeneity and representativeness on several fuel rods
- Rod behavior (geometrical changes, FGR…)
- Long term irradiations (creep, corrosion…)
- Conditioning before other tests (ramps…)
Request:
Failure risk and radionuclide release
- Power ramps
- Post-failure behavior
- Lift-off thresholds
- Melting margins
Safety tests
Request : Highly instrumented and automated test in a devoted environment
- LOCA-type
- Fast transients phenomenology
- Fuel-coolant interaction

Figure 10) Innovative fuel R&D process

JHR Experimental Devices

Development phases of several JHR experimental devices are carried out in order to cope with material and fuel irradiation objectives.

Foremost **CALIPSO** in-core test device has been conceived mainly for cladding materials investigation (in-Core Advanced Loop for Irradiation in Potassium SOdium). It is expected to be placed inside "small" in-core locations in the middle of the fuel elements to take advantage of high neutronic flux and relevant DPA rate up to some 15 DPA/y. It has been designed to achieve a uniform temperature profile between samples of the same batch through a NaK liquid metal cooled loop. In fact, maximum temperature difference is about 8°C for axial profile and about 7°C between specimens. In addition, temperature stability with respect to time is obtained. Very challenging cooling loop has been conceived as far as safety issues and confined, autonomous long period utilizations are concerned. A proper circuit allows tuning mass flow rate thanks to an electromagnetic pump placed at top of the rig (mass flow rate of some 2 m^3/h and pressure drop of about 1,25 bar). In the bottom part of the device loop a heat exchanger is placed in order to remove gamma heating and to control sample temperature. Variable heat exchange surface allows twofold operating conditions ranging from 250°C up to 450°C for what concerns typical LWR operating conditions temperatures. Even higher performances are supposed to be simulated reaching 600°C with particular design configuration.

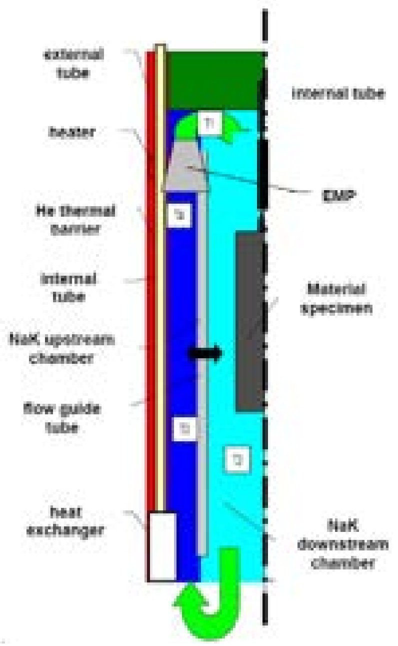


Figure 11) CALIPSO loop layout

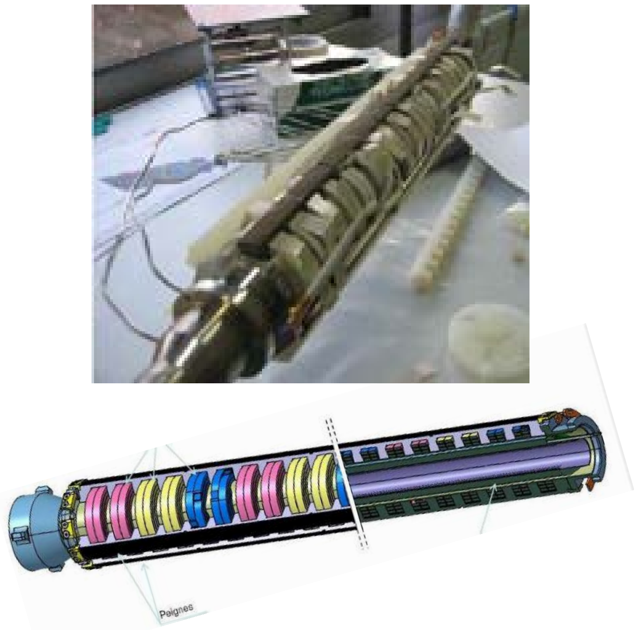


Figure 12) EM pump layout and prototype

Moreover, CALIPSO test device is planned to perform stress and strain tests allowing for on-line measurements and optimal control of experimental parameters. Sample holder design takes advantage of OSIRIS concept and expertise. Up to 5 experimental bases are embarked with 3 pre-pressurized tubular samples placed at 120° on each.

Conceptual design and thermalhydraulic analysis of critical components have been carried out and detailed study of pump and heat exchanger has been completed. Industrial procurement phase has now started and a prototype is under development.

Another liquid metal cooled test device is expected to be hosted inside the JHR core - namely **MICA** (Material Irradiation CApsule). Here a NaK cooled loop is devoted to stress relief and creep analysis basically for cladding materials. Moreover, deformation and swelling analysis with on-line measurements are performed. Within this device it is possible to apply bi-axial stress to up to 10 tubular samples.

Thermalhydraulic features of a natural convection loop are utilized within a temperature range from about 200°C to around 450°C. Technological design utilized important experience from CHOUCA devices in OSIRIS material testing reactor and GRIZZLI apparatus.

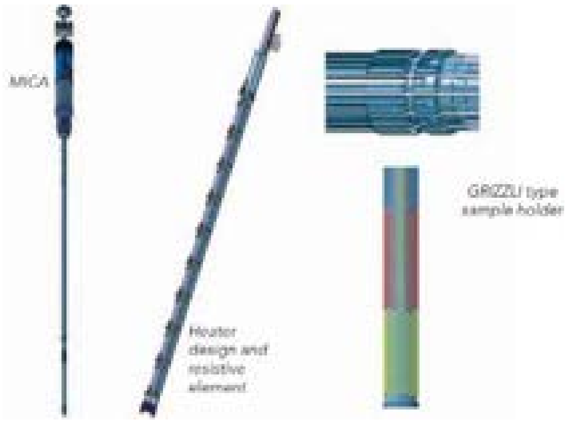

MICA
GRIZZLI type sample holder
Heater design and resistive element

Figure 13) MICA device layout

In addition, **MELODIE** experimental sample holder (MEchancal LOading Device for Irradiation Experiments) is under development. Normal and incidental conditions in LWR parameters induce fission gas release and pellet cladding interactions which lead to complex and multi-axial thermo-mechanical loadings. In order to increase fuel burnup and to get better safety features, it is necessary to obtain more reliable experimental data. Within the framework of European work package WP 1.1 program, French CEA and Finnish VTT joined the same project to realize this test device. MELODIE is expected to allow creep analyses concerning PWR cladding materials and a full on-line control of bi-axial stress; corresponding strain measurements are foreseen as well. Mechanical studies will be possible in a quite high temperature environment of about 350°C.

An improved MELODIE technology device is planned to be hosted in JHR. Design stage is completed and realization and production phase has already started.

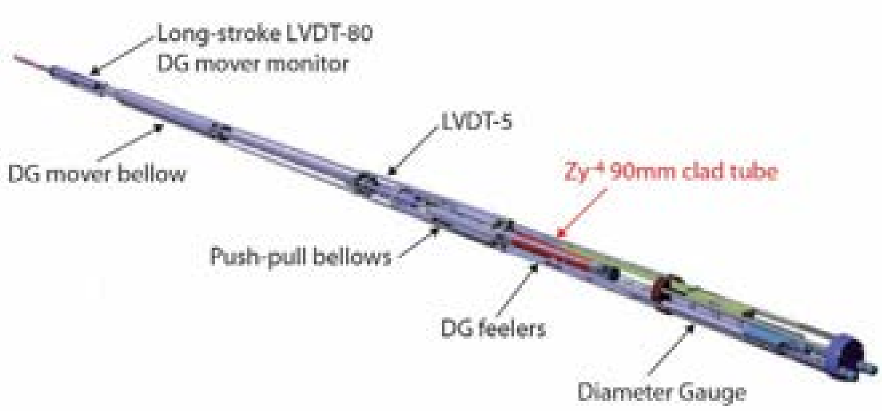


Figure 14) MELODIE sample holder layout

JHR fast in-core neutron spectrum can be utilized also for GenIV R&D purposes. GFR technology envisages SiC/SiC composites as structural material for fuel containment. Then irradiation creep evolution at elevated temperatures combined with high fast neutron doses is a key topic in order to improve structure performances. Thus **CEDRIC** device (Creep Experimental Device for Research on Innovative Ceramic) is designed to be able to apply controlled stress for quantitative analyses and to precisely measure resulting strains. Superposition effects are investigated utilizing two samples: the first is just irradiated and the second one undergoes the same radiation dose and controlled stress at the same time. High temperature environments are simulated ranging from 600°C up to about 1000°C. A rig similar to a CHOUCA device is filled with helium gas and placed within JHR core to exploit a suitable fast flux. CEDRIC irradiation analyses have been performed inside OSIRIS reactor core. Moreover, the CROCUS device has investigated SiC/SiC bond stability with respect to SiC compound structure. In 2007 several samples have been irradiated and some important data have been obtained. Taking advantage of OSIRIS tests expertise, an improved technology will be used in the JHR.

In order to provide support in R&D for future GenIV reactor technology as well as for property enhancement of present LWR, large irradiation capacity devices are being realized within the framework of JHR project. Natural convection helium-cooled loop is under development aimed at exploiting the fast core flux through large test slots. The objective is to investigate innovative materials – namely advanced stainless steels (austenitic or ferritic steels utilized in sodium and lead-cooled technologies) or ceramics. Basically, LWR technology steels or zircalloy are interesting for such an irradiation capacity.

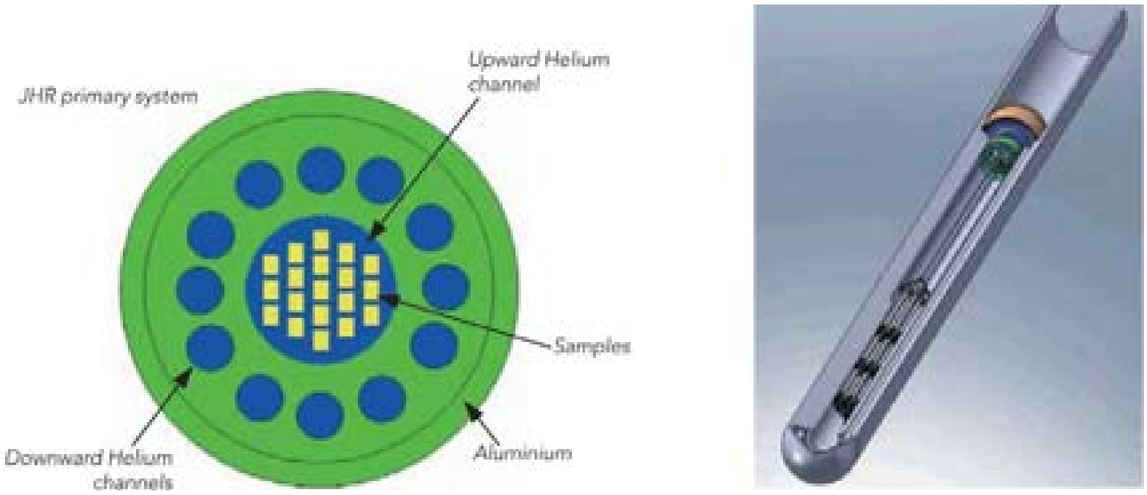


Figure 15) In-core large slot irradiation capability

GenIV advanced systems conceived to use innovative fuels and burn minor actinides need more precise data to R&D process support. SFR and LFR technologies are getting key concepts in future nuclear power plant fleets and thus JHR design team foresaw some experimental capabilities to cope with these challenges. Firstly evaluations have been carried out to simulate helium fission production within an Am matrix through flux, enrichment and minor actinide composition changing in order to exploit thermal flux. Helium build-up is relevant and both swelling and internal pressure make it a critical issue in fast reactor design.

Moreover, JHR test capability is expected to be oriented also towards reduction of fuel pellet data uncertainties as well as long term behaviour. Future experimental needs taken into account are also accidental simulations about innovative fuels and clad failure investigations.

Furthermore, many experimental apparatus are devoted to fuel properties enhancement. **MADISON** device (Multi-rod Adaptable Device for Irradiations of experimental fuel Samples Operating in Normal conditions) is designed to perform fuel tests concerning PWR, BWR and VVER reactor technologies. It can embark 4 fuel pins (even 8 pins capacity is conceived) and reproduce normal operating conditions not aiming at clad failure. In order to exploit a proper thermal neutron flux it is placed inside the JHR reflector. Nominal reactor operation conditions are achieved also through an independent loop in which representative thermalhydraulic and chemical conditions are set up (PWR conditions achieved through pressure of some 160 bar and temperature of about 320°C). Different slow power transients are induced thanks to a moving structure whose distance from JHR core is controlled to modify neutron flux within the fuel samples. MADISON is expected to study either slow power slopes or long period irradiations (up to 3 years). Fuel material properties (microstructure, fission gas release, mechanical features…) are investigated with respect to burn-up and linear heat generation rates. Clad corrosion and crack initiations are also interesting topics for long irradiation tests.

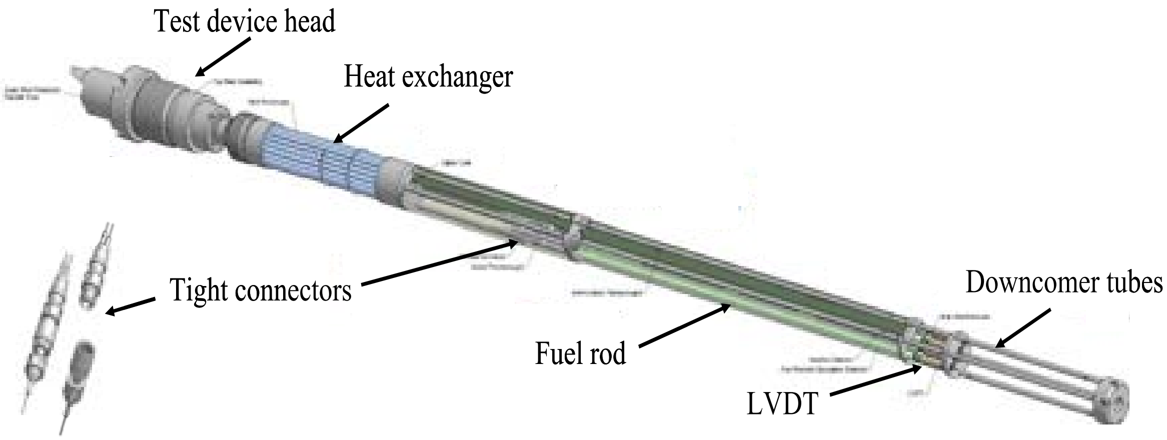


Figure 16) MADISON device layout

Very homogeneous neutron irradiations as well as high precision measurements are significant features of this device. Nominal linear power envisaged is about 400 W/cm simulating high burn-up fuels by means of 1% U235 enriched UO_2 samples.

MADISON is going to perform different kind of experiments:

- selection tests to irradiate and compare innovative samples
- characterization tests to irradiate few samples and to obtain many physical information
- qualification and validation tests to reproduce reactor operative conditions

This device is capable to utilize several JHR facility apparatus and examination tools.

MADISON design takes advantage of important collaborations between French CEA and IFE Halden expertise which started from domestic know-how to reproduce in JHR an innovative and challenging loop. Design and feasibility phases are completed; realization and manufacturing stages are ongoing.

Moreover, **ADELINE** in-reflector experimental device is conceived in order to perform single LWR fuel pin tests concerning up to limit and incidental scenarios. It is hosted on a moving structure and several power ramp tests take advantage of the ISABELLE1 device expertise in OSIRIS reactor. Rod internal over-pressurization and free gas sweeping, as well as fuel centre melting approach, are investigated since clad failure configuration is allowed in this apparatus. Then precise measurements in clad failure timing and linear heat generation rate related to incidental situations are achieved in this device.

Moreover, normal conditions after clad failure are envisaged since the loop is designed to operate with contaminated coolant. Fission gas release during transients is detected by Fission Product Laboratory instrumentations through on-line gamma spectrometry and delayed neutron detection techniques. In addition, permanent purifications and radiology controls are performed on the out-of-pile part of the loop. In order to limit the amount of contaminated coolant a jet pump is installed inside the device. The thermalhydraulic and chemical representative environment are achieved for what concerns failure simulations.

ADELINE apparatus is placed on reflector moving structure to set thermal flux and then power levels. Typically both PWR, BWR and VVER technologies are studied and either UO_2 12% 235U enriched fuel or MOX 20% Pu/(Pu+U) enriched fuel are utilized.

Moving structure allows power ramps and a sample test transient may induce a first irradiation plateau (1 day up to 1 week) at a linear power of some 100 W/cm; then a ramp is induced ranging from 100 W/cm/min up to 700 W/cm/min. Furthermore a high power plateau is kept for about 24 hours at about 620 W/cm. As explained before, the facility design allows withstanding clad failure during this procedure.

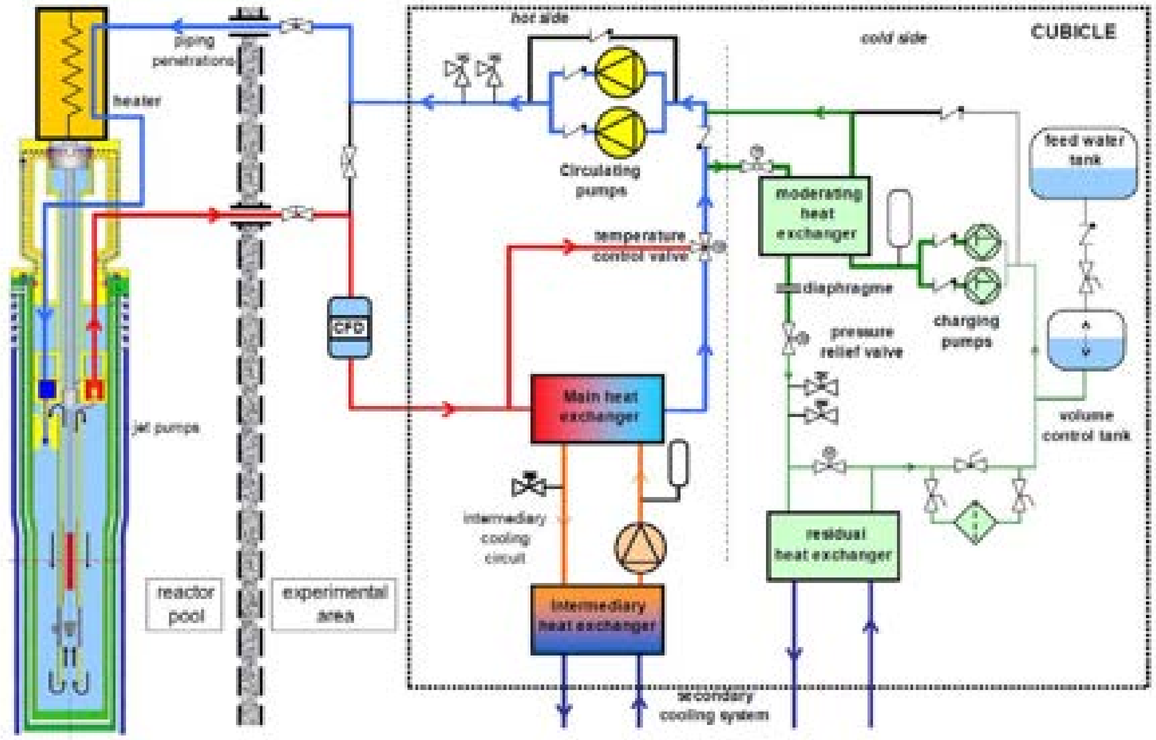


Figure 17) ADELINE loop layout

LORELEI experimental device (Light water One Rod Equipment for LOCA Experimental Investigations) is designed to investigate LOCA transients. Basically thermalhydraulic aspects, radiological consequences and mechanical issues of this kind of accident are the objective of foreseen simulations. Since this device is located on a moving in-reflector test slot, power level is expected to be controlled and independent loop assure safety requests to be respected. LORELEI design is aimed at understanding of ballooning and burst of cladding materials and corrosion phenomena in elevated temperature environment.

Simulation scenarios take into account a first depletion phase in order to create a representative fission products inventory within the fuel matrix.

Then a dry-out phase through gas injection is devoted to start the loss of coolant. Steam is produced thanks to evaporation of some liquid water in the bottom part of the device.

Power is tuned by means of device displacing and a homogeneous temperature profile is reached through both several electrical heaters and a screen to flatten the neutron flux.

Fission Product Laboratory is always connected to the loop to perform on-line radiological and chemical analyses about the test evolution.

This device design is based on GRIFFON apparatus which has been operated inside the SILOE reactor under the umbrella of the FLASH Program for LOCA accidents.

Finally, **OCCITANE** experimental device (Out-of-Core Capsule for Irradiation Testing of Ageing by NEutrons) is devoted to steel analysis and investigations. Since vessel is a critical component for a nuclear power plant and a barrier according to defence-in-depth safety approach, it is necessary to qualify material features to prove their lifetime within a neutron radiation field. OCCITANE technology is based on IRMA device which is located in OSIRIS reactor. Irradiation is performed in an inert gas atmosphere and a temperature of some 230°C to 300°C is expected. On the other hand, dose rates of some 100 mdpa/y are envisaged. Optimization phase is now concerned with best location within the reflector slots. Temperature control is a key point and a thermal and mechanical simulation study is going on.

JHR Irradiation Devices for Medical Purposes

Medical radioisotopes production and procurement is becoming a more and more strategic issue within the framework of worldwide healthcare system. Nuclear physics application has allowed precise imagining and effective therapies for twenty years. Nuclear medicine diagnostics involves several kinds of radiation and procedures taking advantage of unstable artificial nuclei undergoing decay. In this branch of medicine, the most important radioisotope is Technetium-99m and it is used nowadays in around 20 millions diagnostic procedures: half of them are bone scans, and the remaining half is roughly divided between kidney, heart and lung scans. This radionuclide is suitable for medical imaging since it has got a half-life of about 6 hours which is low enough to allow the patient to leave the hospital after short delay. Conversely the half-life of the parent nuclide – the Molybdenum-99 – is some 66 hours long enough to be transported from the processing sites to the end user facilities.
Therefore Technetium-99m supports around 85% of nuclear medicine diagnostics and then Molybdenum-99 (Mo99) production and stock are key issues in worldwide healthcare management.

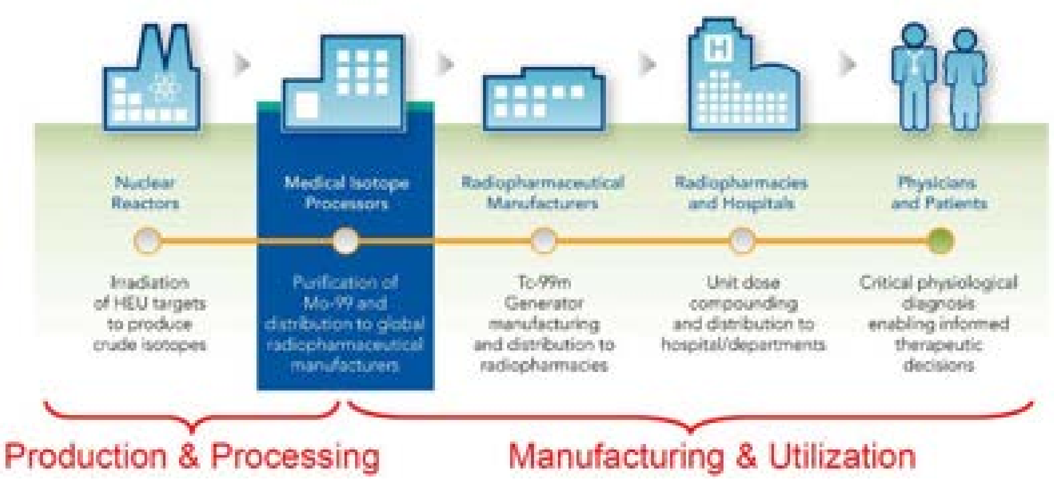

Nuclear Reactors
Medical Isotope Processors
Radiopharmaceutical Manufacturers
Radiopharmacies and Hospitals
Physicians and Patients
Irradiation of HEU targets to produce crude isotopes
Purification of Mo-99 and distribution to global radiopharmaceutical manufacturers
Tc-99m Generator manufacturing and distribution to radiopharmacies
Unit dose compounding and distribution to hospital/departments
Critical physiological diagnosis enabling informed therapeutic decisions
Production & Processing
Manufacturing & Utilization

Figure 18) Molybdenum-99 production and utilization

Within a European framework, a research reactor network assures Mo99 stocks for hospitals and medical processes. Recent 2008 and 2009 crisis in Mo99 supply highlighted the need for an infrastructure optimization concerning transportation, stock and management. On the other hand, it has been possible to realize the importance of joint efforts performed by multipurpose material testing reactors and their vital role in providing radioisotopes.
Since the construction of an industry owned Mo99 production dedicated reactor has failed in Canada (MAPLE project), European NEA and OECD strategies have envisaged restoring the existing research reactor infrastructure managed by public bodies and networked in order to assure proper spare production. In fact, European market is conceived as a multi-line backup system. Conversely other local markets – namely Australia, Canada and South Africa – are single-line structured and less able to prevent or face supply shortage.
European research reactors construction started before a significant diffusion in nuclear medicine diagnostics and then overcapacity has always allowed the market to keep the prices low. People in Europe accessing these diagnostics have increased up to 9 millions. Thus it is necessary to ensure future production provided that short term availability of these facilities is compromised due to temporary maintenance and extended time scale in replacement of ageing existing research reactors.
European production network was formed until recently by 3 reactors:

- BR2 (Mol, Belgium) is a multipurpose research reactor operated by SCK-CEN and it is expected to leave service by 2023.
- HFR (Petten, Netherlands) is a multipurpose research reactor owned by European Commission and operated by NRG, it is expected to leave service by 2018.

- OSIRIS (Saclay, France) is a multipurpose research reactor operated by CEA and it is expected to leave service in 2015.

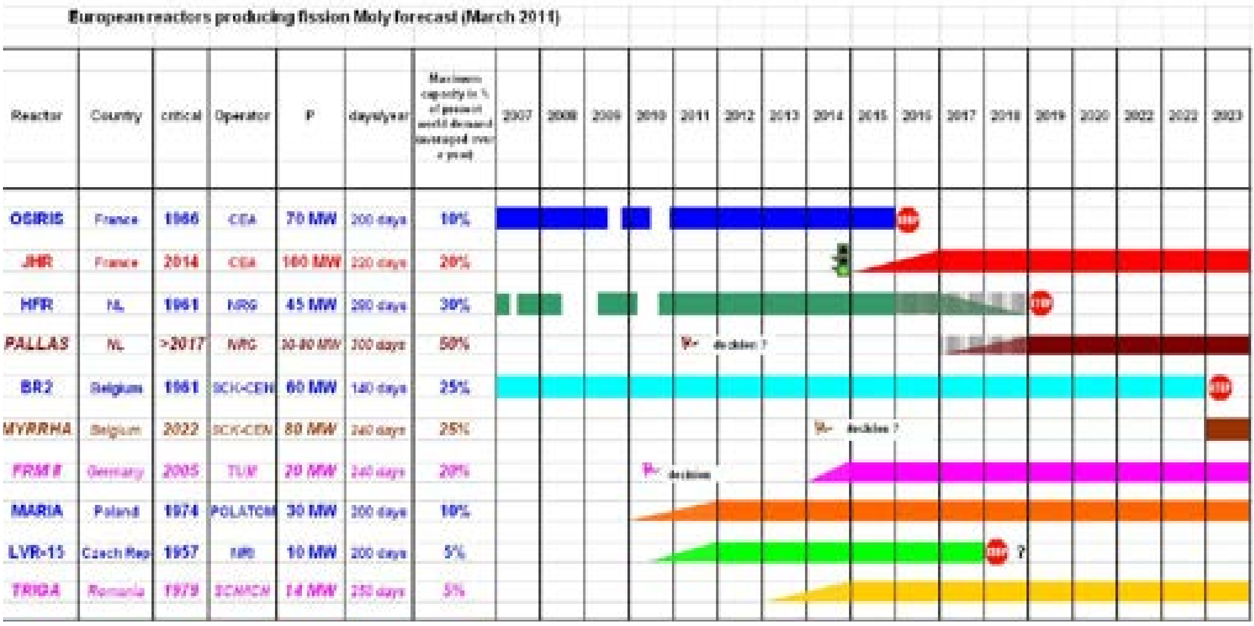
Recently, two facilities joined this infrastructure in order to enhance production:

- MARIA (Otwock-Swierk, Poland) is a multipurpose research reactor operated by Institute of Atomic Energy POLATOM which started production in 2010 and is expected to achieve around 50% of European needs by 2012).
- LVR-15 (Rez, Czech Republic) is a multipurpose research reactor refurbished in 1989 and operated by RCR. It is expected to be operated until 2029 at least.

Operating lives and shutdown timetable are strictly demanding for reactor replacement and future European scenarios are planned in order to assure Mo99 supply system.

- FRM (Munich, Germany) is a new research reactor operated by TUM since 2005 which is supposed to reach peak production around 60% of European demand
- TRIGA (Pitesti, Romania) is a multipurpose research reactor operated by INR until 2030.
- JHR (Cadarache, France) is going to be a new material testing reactor under construction and it will be operated by CEA. It is expected to reach criticality in 2016 and to be able to supply about 35% of European need up 50%.
- PALLAS (Petten, Netherlands) is a multipurpose research reactor to be operated starting from 2017 by NRG to replace HFR
- MHYRRA (Mol, Belgium) is designed to be an accelerator driven system (ADS) scheduled to be operated by SCK-CEN from 2022 in order to face to 100% of European demand.

Several facilities are supposed to face increasing in European Mo99 need provided that some 200%/250% of market demand has to be guarantee by total peak generations. It is necessary to be able to withstand unexpected plant shutdowns or planned maintenance. In addition, it is useful to get a backup capacity in order to supply other regional process lines all over the world.



European reactors producing fission Moly forecast (March 2011)

Reactor	Country	critical	Operator	P	days/year	Maximum capacity in % of present world demand (averaged over a year)
OSIRIS	France	1966	CEA	70 MW	200 days	10%
JHR	France	2014	CEA	100 MW	220 days	20%
HFR	NL	1961	NRG	45 MW	280 days	30%
PALLAS	NL	>2017	NRG	30-80 MW	300 days	50%
BR2	Belgium	1961	SCK-CEN	60 MW	140 days	25%
MYRRHA	Belgium	2022	SCK-CEN	80 MW	240 days	25%
FRM II	Germany	2005	TUM	20 MW	240 days	20%
MARIA	Poland	1974	POLATOM	30 MW	200 days	10%
LVR-15	Czech Rep	1957	NRI	10 MW	200 days	5%
TRIGA	Romania	1979	SCN/ICN	14 MW	250 days	5%

Figure 19) Research reactor replacement timetable

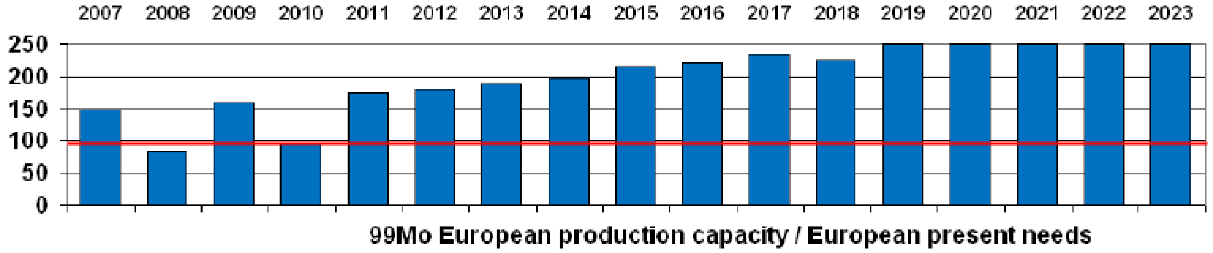


Figure 20) Evolution in percentage supply of Mo99 European demand (%)

In this critical and strategic European scenario concerning healthcare system, research reactors play an important role to secure public issues. A more and more collective and networked approach is envisaged to fit with demand and guarantee operational flexibility in order to maintain low prices and fair trade market. Replacing ageing reactors and add a significant generation capability to facility network will increase reliability and will strengthen European procurement and stock capacity.

JHR is also expected to become a very important keystone even in this framework. It is planned to contribute to a nuclear research infrastructure which is capable to share experimental capacity and high level technological facilities for the benefit of European community. Progress in nuclear science is getting more and more present in everyday life and research reactors have got prepared to match with all these challenging scenarios.

2 Research Description

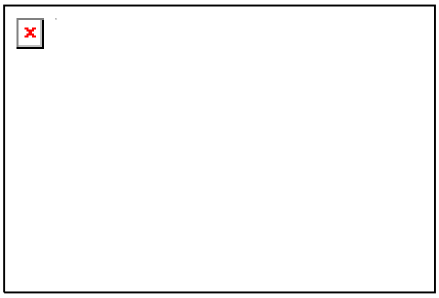
2.1 Thermalhydraulic and Neutronic Core Model

Power transient analyses of nuclear reactors are very important design steps in order to study the evolution of temperature in many significant components during shutdown, start-up or regulation phases. In fact, material resistance and mechanical properties strongly depend on temperature. Mainly, fuel and cladding heating are controlled to prevent them to melt and then release some radioactive nuclides into the cooling loop. Safety defence-in-depth envisages the integrity of fuel matrix and cladding shell as first and second barrier against leakage of radioactive fission products.

Power transients during reactor shutdown are particularly evaluated in the present analysis. Energy release throughout the core depends on neutron and gamma radiation transport but basically the power source is related to fission reaction rate in fissile material – namely the fuel.

Control rod insertion causes an increasing in neutron absorption due to the hafnium tendency to neutron capture. It induces a reduction in neutron population within the system at a time scale compared to mean generation time of prompt neutrons. Consequent temperature drop has neutronic feedbacks related to Doppler broadening, moderation and absorption changes. Then, neutron kinetics requires a coupled thermalhydraulic and neutronic analysis of the system in order to account for these interactions. For this purpose, the DULCINEE kinetics code is utilised. It has been developed by the French Institute of Radiation Protection and Safety (IRSN) and it computes power evolution according to pointwise neutron kinetics approach.

Foremost, a representative thermal model has been conceived in order to well describe the temperature evolution within the fuel, the cladding and the coolant. Providing separation of variables and computing energy conservation for average temperatures in fuel, cladding and coolant, physical similarity is reached coherently with the need for a correct time behaviour description. Then JHR fuel equivalent plate has been simulated both considering the average fuel element and the hottest plate in the core.



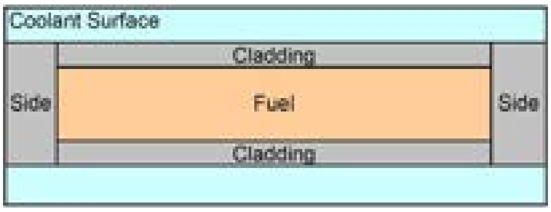
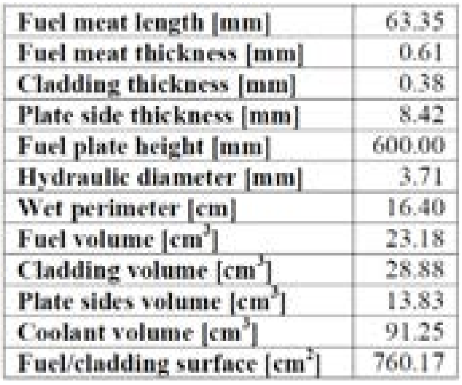
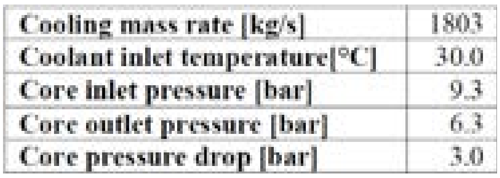


Figure 21) JHR fuel plateFigure 22) JHR model plate



Fuel meat length [mm]	63.35
Fuel meat thickness [mm]	0.61
Cladding thickness [mm]	0.38
Plate side thickness [mm]	8.42
Fuel plate height [mm]	600.00
Hydraulic diameter [mm]	3.71
Wet perimeter [cm]	16.40
Fuel volume [cm^3]	23.18
Cladding volume [cm^3]	28.88
Plate sides volume [cm^3]	13.83
Coolant volume [cm^3]	91.25
Fuel/cladding surface [cm^2]	760.17



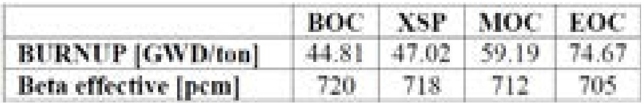
Cooling mass rate [kg/s]	1803
Coolant inlet temperature[°C]	30.0
Core inlet pressure [bar]	9.3
Core outlet pressure [bar]	6.3
Core pressure drop [bar]	3.0

Table 1) JHR model plate dimensions

Table 2) JHR core boundary conditions

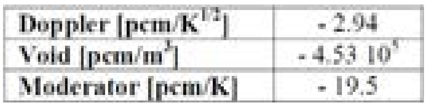
Moreover, the neutronic parameters of the system account for the delayed neutron fraction, the mean neutron lifetime and the time constants of neutron precursor groups. Fuel composition changes due to fission products poisoning and burnup of heavy nuclei. As far as related effect on neutron kinetics, the equilibrium cycle has been divided into four steps: Beginning of Cycle (BOC), Xenon Saturation Point (XSP), Middle of Cycle (MOC) and End of Cycle (EOC).

The following table shows a negligible Pu239 build-up which induces delayed neutron fraction to remain quite constant as well as delayed contributions related to precursor groups; deterministic APOLLO-MOC calculations have evaluated neutron lifetime around 40 µsec.



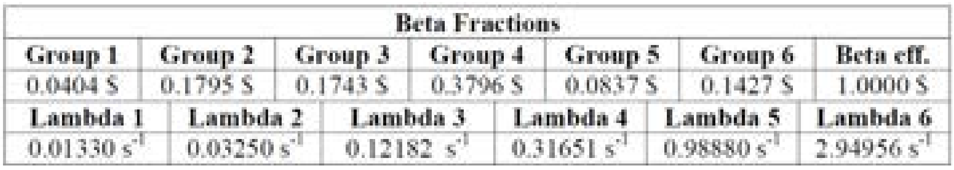
	BOC	XSP	MOC	EOC
BURNUP [GWD/ton]	44.81	47.02	59.19	74.67
Beta effective [pcm]	720	718	712	705

Table 3) Equilibrium cycle compositions



Doppler [pcm/K$^{1/2}$]	- 2.94
Void [pcm/m^3]	- 4.53 10^5
Moderator [pcm/K]	- 19.5

Table 4) Feedback coefficients



Beta Fractions						
Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Beta eff.
0.0404 $	0.1795 $	0.1743 $	0.3796 $	0.0837 $	0.1427 $	1.0000 $
Lambda 1	Lambda 2	Lambda 3	Lambda 4	Lambda 5	Lambda 6	
0.01330 s^{-1}	0.03250 s^{-1}	0.12182 s^{-1}	0.31651 s^{-1}	0.98880 s^{-1}	2.94956 s^{-1}	

Table 5) Neutron kinetics parameters

Finally the thermal effects of the system are related to neutronic description through reactivity feedback coefficients. Then it is possible to consider Doppler effect and change in moderation due to void or coolant dilatation.

DULCINEE code has allowed modelling JHR as far as thermal features are concerned and feedback relations impact on neutron kinetics. Once pointwise kinetics model has been solved, power transients have been evaluated.

2.2 Shutdown Systems for JHR

In the present analysis only power transients during shutdowns are considered. JHR safety approach envisages two different shutdowns for reactor piloting – namely Normal Shutdown (NS) and Safety Shutdown (SS). As explained before, JHR is equipped with 27 control rods for core poisoning compensation, reactor piloting and emergency shutdown. A group of 19 Compensation Rods (CR) is designed to one by one withdrawal throughout the cycle to provide extra reactivity and assure system criticality. Moreover, a 4 Pilot Rods (PR) bank is kept as close as possible to the core mid-plane in order to take advantage of the highest differential worth. Remaining 4 Safety Rods are clustered in a bank as well and completely extracted from the core during normal operations.

Normal Shutdown utilizes just the Pilot Rods bank and the injection starts from criticality position until core bottom. Since this initial insertion depends on fission product poisoning, antireactivity introduced by control devices changes during reactor cycle as well as insertion speed. Then Pilot Rods worth may change from some 2200 pcm up to 2700 pcm.

By means of the DULCINEE code, thermalhydraulic coupling with control rod antireactivity insertions has allowed to compute the Normal Shutdown power transient.

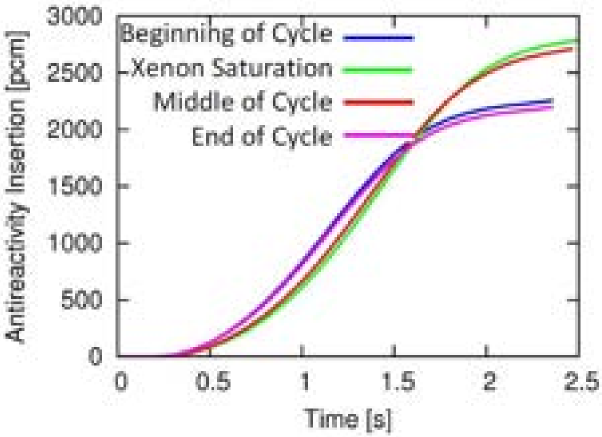


Figure 23) Normal Shutdown antireactivity injection

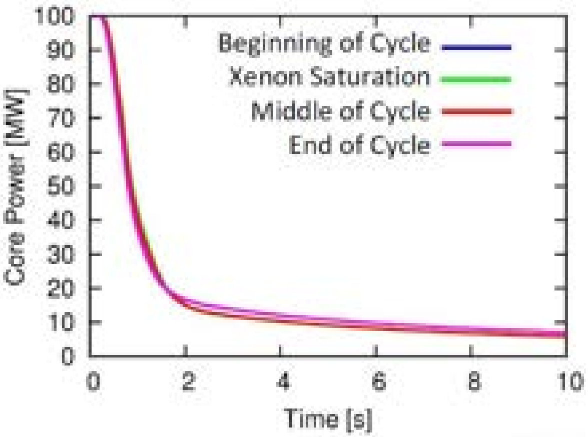


Figure 24) Normal Shutdown power transient

On the other hand, Safety Shutdown has been conceived to provide a higher amount of antireactivity and it employs both the Pilot Rods and the Safety ones. It is a Normal Shutdown with a complete Safety Rods insertion at the same time from top of the core in just 1.23 sec.

As shown in the previous case, even power transient during Safety Shutdown has been evaluated.

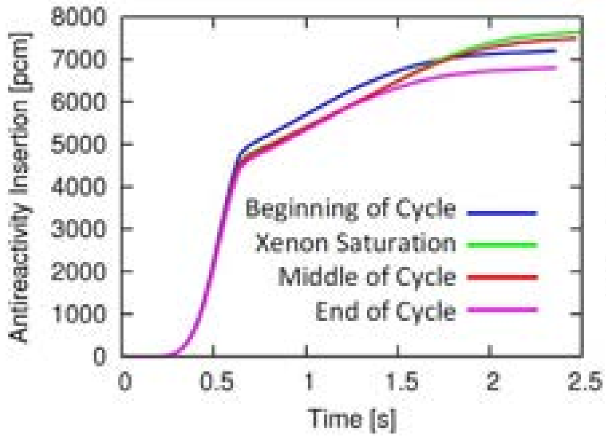


Figure 25) Safety Shutdown antireactivity injection

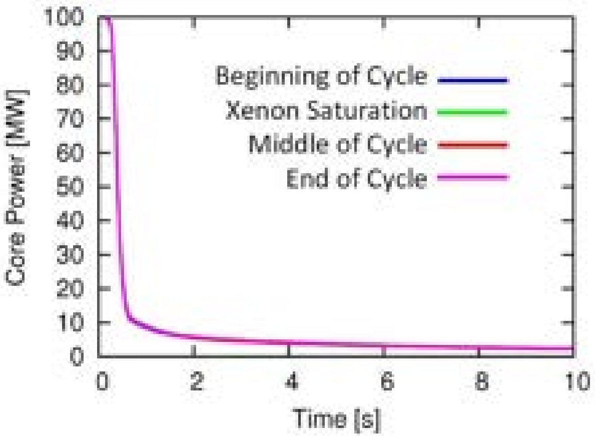


Figure 26) Safety Shutdown power transient

2.3 Core-Reflector Coupling

JHR fast core spectrum sustains nuclear chain reaction and neutrons getting out of the central region towards the reflector undergo fission within the test samples. So there is a strong coupling between these two reactor parts that is worth to consider during transients.

If absorbing materials such as control rods are inserted into the core, the flux shape changes and then it exhibits a significant dependence on their position, even reduced power causes flux amplitude to decrease.

The propagation of core flux decrease occurs throughout the reflector with a twofold time scale. Foremost the prompt neutron lifetime – about 40 microsec – is the order of magnitude of about 90% of reactor thermal power. This sudden reduction can be thought of as immediate compared to control rod kinetics.

On the other hand, delayed neutron production time scale goes from some tenths of second up to about 50 sec, depending on precursor group. They account for at most 1% of total power and they continue to be produced according to neutron precursors decay laws. It is a sort of intrinsic power generation capacity of irradiated fissile materials and it has to be evacuated by the coolant loops for a period of time which is proportional to initial power level.

Both prompt and delayed neutron generations are present within reflector fuel samples. So this double time scale in neutron supplying to devices has to be thought of as overlapped with local multiplication. The latter causes a delay superposition which affects power production in specimens.

A precise evaluation needs particular neutron kinetics issues to be taken into account during shutdown analyses. On the other hand, a complex geometric domain requires the utilization of Monte Carlo 3D codes to properly consider core-reflector interactions, provided a lack of useful symmetries.

Code state of the art in Monte Carlo techniques doesn't allow the user to perform time dependent calculations related to variations concerning geometry or compositions. In fact, multiphysics analyses related to thermal, mechanical or domain shape feedbacks are still under development by several software and code within the framework of international projects. So it is not currently possible to account for a neutronic source with changes with respect to time due to control rods insertion. It is not even possible to properly evaluate the neutron irradiation received by the samples under the absorber induced flux shape modification.

Then TRIPOLI 4.8 Monte Carlo transport code has been utilized to compute neutron flux distribution with static calculations all over the reactor and the reflector. This code has been developed by French Atomic Energy Commission (CEA) and it solves transport equations utilizing a statistical approach to integral equations taking advantage of a large number of particle simulations by means of parallel computing.

Normally, neutron transport techniques take into account the static solution related to Boltzmann equation coupled with neutron precursor group decay.

Provided $\varphi(\vec{r}, E, \Omega, t)$ the neutron flux depending on space phase – namely geometric position, velocity and direction – it changes with respect to time as follows:

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial\varphi}{\partial t} = -G\varphi - \Sigma\varphi + \Sigma_s\varphi + M_p\varphi + \sum_i \lambda_i C_i \\ \dfrac{\partial C_i}{\partial t} = -\lambda_i C_i + M_{R_i}\varphi \end{cases}$$

These equations, coupled with usual boundary conditions, rule evolution of neutron flux with respect to time. The first takes into account a balance between geometrical leakage (G), absorption and out-scattering (Σ), in-scattering (Σ_s) and prompt neutron production (M_p). The last sum provides delayed neutron contributions. On the other hand, the second equation rules the build-up of neutron precursors due to delayed production term (M_R).

Supposed a time independent problem, derivatives are turned to zero and the previous equation, always coupled with proper boundary conditions, yields:

$$k\,L\,\varphi = \left((1-\beta)\chi_p + \sum_i \chi_{Ri}\beta_i \right) M\varphi$$

The effect is to compute the neutron production throughout a fissile material considering a weighted generation spectrum which influences all nuclear reactions. This holds at constant power during nominal operation when the ratios between the delayed neutron populations and the prompt one are constant and equal to beta fractions. During shutdowns which take about 1 or 2 sec, prompt neutron population decreases very quickly but delayed neutrons continue to be injected into the system by means of precursors decay.

Static calculations evaluate just flux shape related to a constant neutron production which corresponds to a virtual constant reactor operation for every given control rod insertion.

Monte Carlo code gives results that need to be normalized compared to meaningful integral values of the system, typically the total amount of fission reactions which is well defined either in nominal or in dynamic analysis.

DULCINEE shutdown transient evaluations provide the total core power with respect to control rod positions and time. Therefore TRIPOLI 4.7 static calculations have been coherently normalized conserving the total amount of fission reactions.

On one hand conservation of the intensity of the flux is respected, on the other hand prompt-to-delayed neutron fraction is overestimated during shutdown due to Monte Carlo approach. This effect simulates a more energetic neutron population with a higher likelihood to get out of the core and then to reach the reflector devices.

Even neglecting effective energy distribution in neutron flux, a more conservative spectrum is considered and then a better evaluation for what concerns safety issues is obtained.

Since the objective of the present study is the power release in reflector devices, it is worth not just to consider flux shape but also fission reaction rate distribution.

In addition, even fission products kinetic energy is the most important contribution to power deposition; neutrons and gamma radiation are significant energy carriers. The first ones undergo scattering reactions and absorption generating gamma rays, gamma photons heat up metals and structures due to attenuation processes.

So it is not possible to consider that all the amount of energy released is deposited at the same place since it diffuses depending on particular radiation transport laws.

In order to take also into account the core contribution to reflector device heating for radiation leakage, energy deposition calculations are suitable in order to be coherent with physical phenomena.

Gamma radiation is created during fission process and it leaves the site diffusing all around from about 10^{-14} sec and 10^{-7} sec after the reaction, then it is called prompt gamma radiation. Unstable fission products usually decay in several ways, often they emit gamma photons even with time scale longer then fission ones, typically of the order of 10^{4} sec. Activated nuclei undergo gamma emissions as well, and all these are called delayed gammas.

TRIPOLI code doesn't perform material evolution calculations and it can take into account just the prompt gamma radiation.

Then Monte Carlo analysis has considered neutron contribution in energy deposition – namely scattering and fission products kinetic energy – and prompt gamma heating to materials.

2.4 Reflector Device Calculations

Since in-core neutron spectrum is quite fast, several test requiring thermal irradiations are performed inside the beryllium reflector taking advantage of its capability in slowing down neutrons. As explained before, fuel properties are studied for different irradiation levels and with different power ramps simulating various operation conditions.

MADISON test device has been designed to perform fuel irradiation in order to reproduce nuclear power plant normal operations. It will be placed on a moving structure which will be able to set neutron flux properly tuning the distance from the core centre. Normally, it is expected to be loaded with 4 uranium dioxide 1% U235 enriched fuel pins aiming at relatively high burnup fuel simulations.

ADELINE experimental device is aimed at post-failure and abnormal operation conditions studies. It will host just 1 fuel pin (UO_2 1% U235 enriched) and it is expected to perform power ramps up to material limits and simulate normal conditions after partial cladding damage.

Moreover, MOLFI device is placed within the reflector but it is devoted to radioisotope production for medical purposes. Molybdenum 99 is obtained by means of U235 fissions. For that, several AlU targets are put on moving structures to be irradiated and to achieve Moly99 build-up. An example of core and reflector configuration is shown in the picture below.

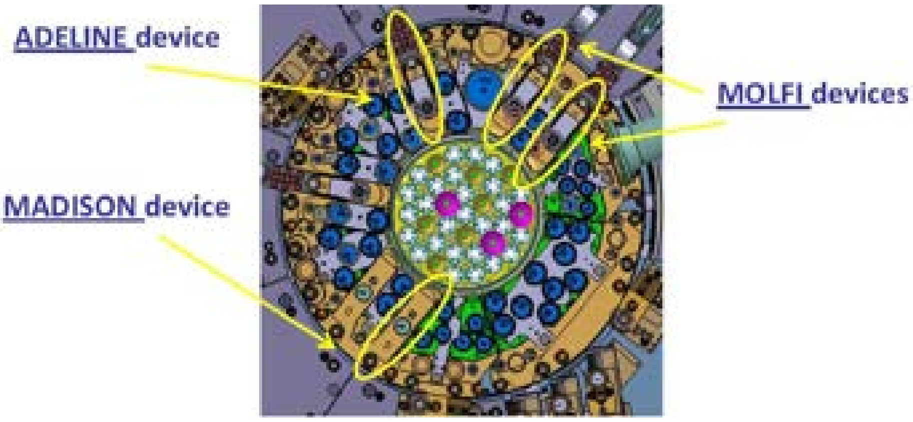


Figure 27) JHR core and reflector devices configuration

Regarding in-reflector test devices, cooling is provided through independent loops designed to be flexible and to reproduce many reactor thermalhydraulic conditions (PWR, BWR).
Safety approach and procedures require power generation evaluations during transients since it is mandatory to control fuel temperatures even for what concerns fuel samples inside experimental in-reflector devices.
As piloting is achieved by means of core control rods, it is necessary to evaluate the reflector devices response to manage their power transients during reactor shutdowns. In addition, if safety temperature thresholds are reached in some fuel loaded samples, it is worth to know their power behaviour and then the related kinetic delays which occur during consequent reactor shutdown. Therefore it is necessary to aim at neutronic coupling description between core and reflector considering absorber insertion effects.
The Monte Carlo model to evaluate energy deposition and then power transients in JHR reflector devices consists of 4 MOLFI devices, 2 MADISON and 2 ADELINE test samples. Different reflector locations are considered for the same experiment aiming at location effect investigation.
Moving structure allows device displacement but the configuration in which the samples are close to the core has been considered since it corresponds to the highest core-reflector coupling.
A simplified analysis considers device power proportional to core power through a time dependent coefficient. This accounts for delays and flux shape modifications. In order to be conservative, power coupling have been evaluated just for nominal configuration and complete control rod insertions after both Normal Shutdown and Safety Shutdown.
The implemented TRIPOLI model is showed in the following picture.

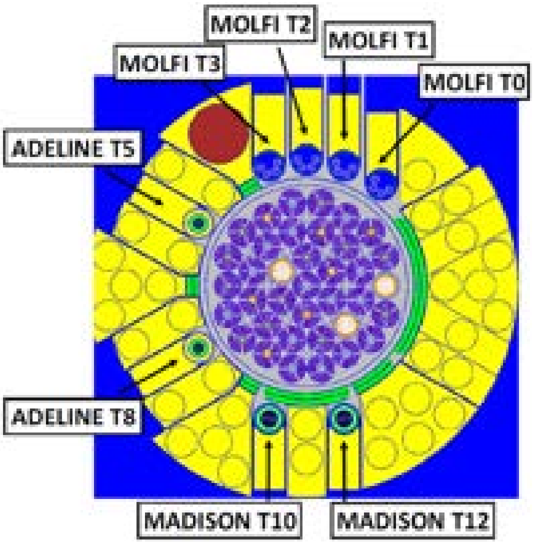
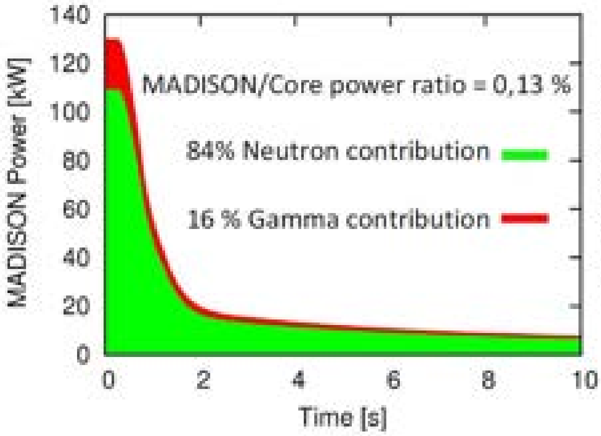


Figure 28) TRIPOLI model for JHR core and reflector

These coefficients refer to energy deposition calculations which take into account neutrons and prompt gamma radiation as well. Such a criterion accounts for radiation transport and basically gamma diffusion.
These ratios have been treated and for every device the coefficient corresponding to the highest power level has been kept as constant during the entire transient. Needless to say, the most conservative configuration corresponds to complete control rod insertions. In fact, the flux shape exhibits significant peaks within reflector fissile samples due to absorber injection in the core.
Therefore an evaluation related to the asymptotic composition has been used even during the very first part of the transient. So starting power is overestimated and safety issues are respected.
Finally Normal Shutdown and Safety Shutdown power transients have been evaluated for every reflector device through TRIPOLI energy deposition coupling coefficients and DULCINEE core transients. BOC and EOC power profiles have been considered regarding Normal Shutdown since they are more conservative. On the other hand, average profile has been implemented for Safety Shutdowns.
Device power transients have then been computed. Neutrons and prompt gamma contributions to sample heating are highlighted.



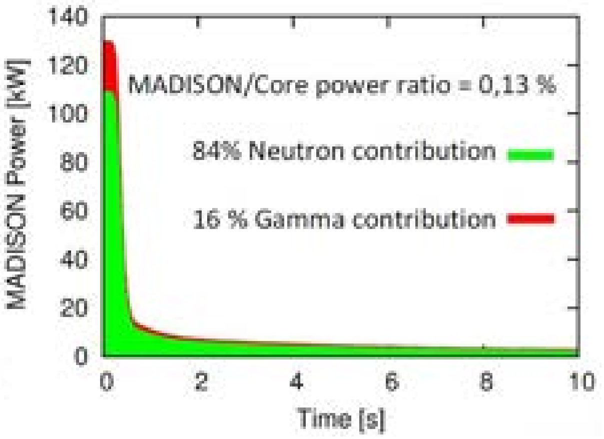
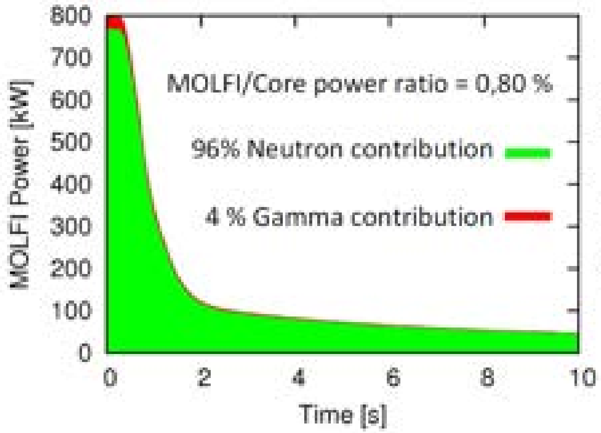


Figure 29) MADISON Normal Shutdown transient

Figure 30) MADISON Safety Shutdown transient



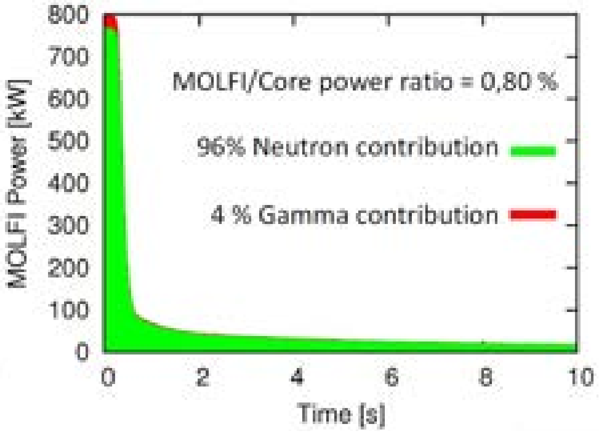


Figure 31) MOLFI Normal Shutdown transient

Figure 32) MOLFI Safety Shutdown transient

Significant difference in neutron and gamma contribution is due to higher MOLFI enrichment. Neutron contributions account for fission products initial kinetic energy and scattering reactions. Then a relatively higher fission rate changes fission-to-gamma balance in energy deposition.

2.5 Conclusions and Perspectives

JHR future CEA material testing reactor is expected to perform several experiments on fuel samples inside reflector. Power transients are worth to be evaluated in these devices aiming at cooling system design and accident scenario simulations.
In the present analysis two shutdown transients have been considered – namely Normal Shutdown and Safety Shutdown. DULCINEE neutron kinetics code has been utilised for reactor core power evolution. On the other hand, reflector-core coupling has been computed by means of TRIPOLI 4.7 Monte Carlo transport code.
Then the most conservative control rod insertion configurations have been chosen and finally it has been possible to obtain the device power transients.
Energy deposition calculation through Monte Carlo method allowed computing different contribution in sample power regarding neutron-induced reactions and gamma heating.
Next task will be about the evaluation of a model which takes into account a multi-point kinetics approach in order to highlight core contribution to device heating. On the other hand an evaluation of the device multiplication contribution is envisaged.

Instead of weighting reaction rates necessary to compute kinetics constants, a twofold neutron transport approach is conceived: prompt and delayed neutrons are then considered with respect to their own time scale and multiplication reactions.

3 References

[1] P. Console Camprini, M. Sumini, C. Artioli, C. Gonnier, B. Pouchin, S. Bourdon, Thermal Hydraulic and Neutronic Core Model for Jules Horowitz Reactor (JHR) Kinetics Analysis, LP2 Project Report RdS/2011/39,
ENEA and Ministry of Economic Development AdP Framework Agreement
"Innovative Nuclear Fission Technology", CIRTEN-UNIBO

[2] P. Console Camprini, M. Sumini, C. Artioli, C. Gonnier, B. Pouchin, S. Bourdon,
Power Transient Analyses of Experimental In-reflector Devices during Safety Shutdown in Jules Horowitz Reactor (JHR),
In International Conference on PHYSics Of the Reactor (PHYSOR 2012),
Knoxville, Tennessee, USA, April 15-20, 2012

[3] P. Console Camprini, M. Sumini, C. Artioli, C. Gonnier, B. Pouchin, S. Bourdon,
Thermal Hydraulic and Neutronic Core Model for Power Transient Analyses of Reflector Experimental Devices during Shutdowns in Jules Horowitz Reactor (JHR),
In Meeting of the International Group on Research Reactors (RRFM 2012),
Prague, Czech Republic, March 18-22, 2012

[4] J.P. Chauvin, NA3.D1 – JHR Users Handbook, JHR Collaborative Project, Grant Agreement n° 206300, CEA, 2009

[5] S. Carassou, G. Panichi, F. Julien, P. Yvon, M. Auclair, S. Tahtinen, P. Moilanen, S. Maire, L. Roux,
Experimental Material Irradiation in the Jules Horowitz Reactor,
In Meeting of the International Group on Research Reactors (IGORR 2005),
Gaithersburg, Maryland, USA, September 12-16, 2005

[6] D. Parrat, M. Tourasse, C. Gonnier, S. Gaillot, P. Roux,
Irradiation Facilities and Examination Benches for Implementing Fuel Programs in the Future Jules Horowitz Material Testing Reactor,
In Meeting of the International Group on Research Reactors (IGORR 2009),
Beijing, P. R. China, October 28-30, 2009

[7] T. Dousson, P. Roux, C. Gonnier, L. Ferry, S. Gaillot, D. Parrat,
Experimental Devices in Jules Horowitz Reactor Dedicated to the Fuel Studies in Support to the Actual and Future Nuclear Power Plants,
In Meeting of the International Group on Research Reactors (IGORR 2012),
Prague, Czech Republic, March 18-22, 2012

[8] C. Colin, J. Pierre, C. Blandin, C. Gonnier, D. Moulin, M. Auclair, F. Rozenblum,
Experimental Devices in Jules Horowitz Reactor Dedicated to the Material Studies in Support to the Current and Future Nuclear Power Plants,
In Meeting of the International Group on Research Reactors (IGORR 2012),
Prague, Czech Republic, March 18-22, 2012

[9] A. Alberman, G. Krysztoszek, M. Ciocanescu, K. Charlton, L. Sannen, B. Ponsard, D. Ceuterick, W. Petry, H. Gerstenberg, V. Vrban,
European Research Reactor Position Paper by CEA, IRN, NRG, RCR, SCK-CEN, POLATOM, TUM, Scenario for Sustainable MolyBDENUM-99 Production in Europe

Author’s CV:

Marco Sumini: Nuclear Engineer, Associate Professor in Nuclear Reactor Physics at University of Bologna since 1987; research activities on Reactor Physics, Core Design and on Neutral and Charged Particle Transport. Experimental activities on Pulsed Power Plasma Devices.

Francesco Teodori: Nuclear Engineer, from 2006 Assistant Professor in Radiation Protection at University of Bologna. Main research interests in the analysis of Nuclear Power Plants accident consequences and on related Radiation Protection issues.

Patrizio Console Camprini: Nuclear Engineer, PhD student in Nuclear and Energy Engineering and Environmental Control at University of Bologna, seconded at CEA Cadarache research Centre within the collaborations between French Atomic Energy Commission (CEA), Italian National Agency for New Technologies, Energy and Sustainable Economic Development (ENEA) and the University of Bologna. Research topics on Jules Horowitz Reactor core design and kinetics analysis.

Carlo Artioli: Nuclear Engineer, serving as scientific researcher for the Italian National Agency for new Technologies, Energy and sustainable Economic Development (ENEA, former CNEN) in Bologna, he has been core design leader for the ADS EFIT reactor within the European IP EUROTRANS and for ELSY European reactor, as well as in charge of definition and designing of the LFR DEMO in the Italian Program. He was leader for the core design of ELFR and ALFRED within the European program LEADER.
Now retired, he cooperates with Societies and Universities as scientific consultant. He is professor of Reactor Core Design in the Postgraduate Master on Advanced Nuclear System in the Bologna University.

Christian Gonnier: Reactor Study Department, Jules Horowitz Reactor Service DER/SRJH Executive for Experimental Devices Development JHR Users Technical Interface about Reactor Performances and Devices Design, CEA, Cadarache,, France.

Serge Bourdon: Reactor Study Department, Jules Horowitz Reactor Service DER/SRJH Modeling and Computation Team - Themalhydraulics for Reactor Core and Devices, CEA, Cadarache, France.

Bernard Pouchin: Reactor Study Department, Jules Horowitz Reactor Service DER/SRJH Modeling and Computation Team - Neutronics and Photonics for Reactor Core and Devices, CEA. Cadarache, France.

ALL. 2: Study And Analysis Of Control Systems Implementable By Programmable Logic For Performance And Safety Improvements Of Novel Nuclear Plants

CENTER OF EXCELLENCE DEWS

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

UNIVERSITY OF L'AQUILA, V. G. GRONCHI 18, 67100, L'AQUILA, ITALY

DELIVERABLE 1

Study and analysis of control systems implementable by programmable logic for performance and safety improvements of novel nuclear plants

Studio ed analisi di sistemi di controllo implementabili con logiche programmabili per il miglioramento della prestazioni e della sicurezza di impianti nucleari di nuova concezione

Authors:
Stefano DI GENNARO, Bernardino CASTILLO–TOLEDO, Fabrizio MEMMI

Principal Investigator:
Prof. Stefano DI GENNARO

Project PAR 2011

June 30, 2012

Abstract
In this deliverable, the aspects concerning the digital implementation of a control law on a physical device are analyzed, from the viewpoint of the study, design and realization of supervisory, control and protection systems for improving the performance and safety of novel nuclear plants. In fact, it is well known that the implementation of control laws with zero order holders, commonly used, introduce a delay and hence could bring to unstable behaviors. A technique called self–triggered control allows determining the sampling times necessary to implement the controller, preserving the desired performance. In the digital logic system scenario, there are many architectures suitable for realizing different kinds of control algorithms, considering performances in terms of timing, power consumption and resources availability. Different implementation solutions are hence studied, from microprocessors to custom devices for digital processing, in order to obtain an evaluation about benefits and drawbacks. In particular, FPGA technology will be studied in detail, explaining how designers can exploit their potentialities in applications of parallel computation, control and digital signal processing.

Riassunto
In questo documento vengono analizzati gli aspetti relativi all'implementazione digitale di una legge di controllo su un dispositivo fisico, dal punto di vista dello studio, progetto e realizzazione di sistemi di supervisione, controllo e protezione per aumentare la prestazione e sicurezza di impianti nucleari di nuova concezione. Infatti è ben noto che l'implementazione di leggi di controllo mediante organi di tenuta di ordine zero, comunemente utilizzati, introducono un ritardo e quindi possono portare a comportamenti instabili. Una tecnica chiamata controllo auto–innescante permette di determinare i tempi di campionamento successivi per implementare il controllore, preservando le prestazioni desiderate. Nello scenario dei sistemi a logica digitale, vi sono molte architetture adatte a realizzare diversi tipi di algoritmi di controllo, considerando prestazioni in termini di tempi, consumo di potenza e disponibilità di risorse. Varie soluzioni implementative sono quindi studiate, dai microprocessori a dispositivi personalizzabili, per ottenere una valutazione dei benefici e dei svantaggi. In particolare la tecnologia FPGA sarà studiata in dettaglio, spiegando come i progettisti possono sfruttare i loro potenziali in applicazioni di calcolo parallelo, controllo e processamento del segnale digitale.

1 The Nonlinear Continuous Time Controllers Implemented by Digital Devices

In [1], [2], the supervision, control and protection systems for nuclear reactors of new generation has been analyzed. Using a mathematical model of the primary circuit, simple enough for the control purposes but accurate enough to capture the nonlinear, time–varying, switching nature of the plant, a dynamic level controller is determined for the pressurizer water level. Moreover, two dynamics controllers have been designed for the pressurizer pressure. These controllers may not use measurements of the pressurizer pressure, relying only on the pressurizer wall temperature measurements.

The aforementioned controllers are continuous time, namely the assume that the state variables, necessary to implement the feedback, are continuously measured, while the control is computed at each time instant and applied to the controlled system. In reality, such controllers are implemented by digital devices, much more flexible than analog devices. Other notable characteristics are the multitasking capability of the digital controllers, and the possibility of parallel data acquisition and computation. This last property allows faster computation times, so approaching the performance of the digitally implemented controller to that obtainable with an analogically implemented controller.

1.1 Digital Implementation of Control Strategies: Some Important Issues to Take into Account

A very popular way of determining and implementing a controller on a digital device is to design the controller assuming the continuous time behavior of the system, and then implement the continuous controller by means of zero order holders, commonly used for digital implementations. Such controller are usually named “emulated” controllers. It easy to understand that this design technique could bring to unsatisfactory behaviors of the controlled system if the sampling time is high. In fact, it is well known that the effect of the presence of zero order holders is equivalent to a delay of approximatively the half of the sampling time. As well known also in the case of linear systems, a delay could bring to unstable behaviors.

The effects of the sampling and zero order holders can be seen also in a alternative way. Let us suppose that the controller is determined considering the continuous time dynamics

$$\dot{x} = f(x, u)$$

where $x \in \mathcal{D}_x$ represent the vector of the state variables, $u \in \mathcal{D}_u$ is the vector of the input variables, $\in \mathcal{D}_x \subset \mathbb{R}^n$, $\in \mathcal{D}_u \subset \mathbb{R}^p$ are the domains where x, u take values, and $f : \mathcal{D}_x \times \mathcal{D}_u \rightarrow \mathbb{R}^n$ is the function describing the system dynamics. Usually, this function is requested to fulfill suitable conditions to ensure existence and unicity of the solution for each initial state $x(0) = x_0$, such as the (local) Lipschitz condition, namely

$$||f(x_1, u) - f(x_2, u)|| \leq L||x_1 - x_2||$$

with $x_1, x_2 \in \mathcal{D}_x$ two generic state values, and L a constant (in the local version of this property, L depends on the region Ω considered) called Lipschitz constant. The value of L measures how much f varies when x_2 differs from x_1, and is a measure of the "nonlinearity" of f. Considering that L can be determined considering the maximum value the jacobian norm $||\partial f/\partial x||$ takes on the region Ω that is taken into account, it is clear that L measures the "derivative" of f.

Let us assume that a controller $u = \alpha(x)$ has been determined somehow. This is a continuous time control, and should be better denoted as $u_t = \alpha(x_t)$, where t is the continuous time. For the sake of simplicity, the dependence on t is usually dropped. This controller is normally implemented considering the sampled state value x_k at time $t = k\delta$, and the (constant) numerical value $u_k = \alpha(x_k)$ is applied to the system by means of zero holder devices, that hold the value u_k over the time interval $[k\delta, (k + 1)\delta)$, where δ is the sampling time and $k \in \mathbb{Z}$ is an integer.

It is therefore clear that eventually the dynamics of the controlled system is given by

$$\dot{x} = f(x, \alpha(x_k))$$

which apparently differ from the ones

$$\dot{x} = f(x, \alpha(x))$$

obtained when the continuous time controller is applied. It is clear that the effect of the sampling is equivalent to a disturbance d acting on the system, since

$$\dot{x} = f(x, \alpha(x_k)) = f(x, \alpha(x)) + d$$

where

$$d = f(x, \alpha(x_k)) - f(x, \alpha(x)).$$

This disturbance is periodically zero at the sampling time $t = k\delta$, while grows when $t > k\delta$. The growth of this disturbance depends on the "nonlinearity" of f and, obviously, on the value of δ. In particular, when δ is small the disturbance remains small enough to affect greatly the performance of the controller. But in the case in which δ could take (relatively) big values, d can take bigger values. The measure of how big this disturbance value can be is given, again, by the Lipschitz constant L, since

$$||d|| = ||f(x, \alpha(x_k)) - f(x, \alpha(x))|| \leq L||x - x_k|| \leq L \max_{x \in \Omega} ||x - x_k|| = d_{\max}.$$

The effect of this disturbance on the closed loop behavior is to take away x from the desired value. Let x us assume[1] that one desires that x goes asymptotically to $x = 0$, by means of the control $u = \alpha(x)$. This control can be designed[2] making use of a (positive definite) Lyapunov function V such that, when $d = 0$

$$\dot{V} = \frac{\partial V}{\partial x} f(x, \alpha(x))$$

is definite negative. If $\alpha_3(||x||)$ is a $\mathcal{K}$ function[3] this can be expressed saying that

$$\dot{V} = \frac{\partial V}{\partial x} f(x, \alpha(x)) \leq -\alpha_3(||x||).$$

This means that V, which can take the meaning of a (generalized) energy function, is decreasing as time passes. Therefore, asymptotically x tends to the origin. When d is nonzero, namely when the emulated controller is used, what can be ensured[4] is that $\dot{V}$ is negative definite outside a region

$$\dot{V} = \frac{\partial V}{\partial x} f(x, \alpha(x_k)) = \frac{\partial V}{\partial x}\Big[f(x, \alpha(x)) + d\Big] \leq -\alpha_3(||x||) + \alpha_4(||x||)d_{\max}$$

where

$$\left\|\frac{\partial V}{\partial x}\right\| \leq \alpha_4(||x||)$$

and $\alpha_4(||x||)$ is an appropriate $\mathcal{K}$ function[5]. To determine such a region, a ball of the origin of radius μ, one has to "spend" a part of the term $-\alpha_3(||x||)$ (ensuring the asymptotic convergence of x to the origin) to "counteract" the positive term $\alpha_4(||x||)d_{\max}$

$$\dot{V} \leq -(1 - \vartheta)\alpha_3(||x||) + \alpha_4(||x||)d_{\max} - \vartheta\alpha_3(||x||) \leq -(1 - \vartheta)\alpha_3(||x||)$$

for

$$||x|| \geq \mu := \alpha_3^{-1}\Big(\frac{1}{\vartheta} \max_{x \in \Omega} \alpha_4(||x||)d_{\max}\Big)$$

where $\vartheta \in (0, 1)$. As a result of the fact that $\dot{V}$ is negative definite outside the ball of radius μ, the trajectories of the controlled system tend asymptotically to the ball of radius μ and remains in there. This

[1] It is possible to show that the origin can be considered as equilibrium point, after an appropriate change of coordinate.

[2] The controllers designed in [1] have been obtained precisely with a Lyapunov–based technique.

[3] A $\mathcal{K}$ function $\alpha_3(r)$ is a continuous function such that $\alpha_3(0) = 0$, and is strictly increasing, see [8].

[4] A "worst–case scenario is here considered.

[5] It is possible to show that such a function $\alpha_4(||x||)$ does exists.

property is usually called “practical stability” of the origin, since generalizes the property of asymptotic stability to the origin. It is also called “ultimate boundedness” of the trajectories, since the solution x_t of the differential equation will enter, at a certain time instant, the ball of radius μ and will remain in it. The choice of ϑ determines a trade–off between of the dimension of the region about the origin and the convergence velocity of the state trajectories to this ball.

Other aspects can be also be addressed[6]. Notably, the fact that the sampling of the state variables necessary to the control implies that the control law is applied with a time delay δ, so that the system dynamics are more precisely given by

$$\dot{x} = f(x, u_{k-1}).$$

The resolution of this problem, that can be addressed by means of predictors of the form

$$\dot{\xi} = f(\xi,\alpha(\xi_k))$$
$$u_k = \alpha(\xi_k)$$

goes far beyond the scope of the deliverable. In any case, also this effect can be seen as a disturbance acting on the system, since the system dynamics can be written as

$$\dot{x} = f(x, \alpha(x_{k-1})) = f(x, \alpha(x)) + \bar{d}$$

with

$$\bar{d} = d + f(x, \alpha(x_{k-1})) - f(x, \alpha(x_k))$$

and treated in a similar way.

From the previous discussion, it is clear that the effect due to the sampling can be seen as due, mainly, to a persistent perturbation acting on the system. It is also clear that the effect of this disturbance is smaller if smaller is the amplitude of the disturbance which, in turn, diminishes as the sampling time δ is smaller. The aim of the present deliverable is to study criteria ensuring better implementations of control laws and logics on digital programmable devices, with the goal of improving the performance and safety in nuclear plants of novel conception. In particular, more effective digital platform will be studied, which render the implementation of control law more flexible and effective.

1.2 A Mathematical Model of the Primary Circuit of a PWR

In [1] a mathematical model of the primary circuit of a PWR has been considered. The reader can find in [1] the details of this model, given by[7]

[6] Further effects, such as quantizations effects, are not considered here, since considered negligible.

[7] The subindices “*r*”, “*pc*”, “*sg*”, “*pr*” refer to the reactor, primary circuit, steam generator, and pressurizer, respectively.

$$
\begin{aligned}
\dot{N} &= -\frac{p_0 + p_1 v + p_2 v^2}{\mathit{\Lambda}} N + S \\
\dot{M}_{pc} &= m_{in} - m_{out} \\
\dot{T}_{pc} &= \frac{1}{c_{p,pc} M_{pc}} \Big[c_{p,pc} m_{in} (T_{pc,i} - T_{pc}) + c_{p,pc} m_{out} \mathit{\Delta} + c_{\psi} N - n_{sg} k_{t,sg} (T_{pc} - T_{sg}) - W_{loss,pc} \Big] \\
\dot{T}_{sg} &= \frac{1}{c^{l}_{p,sg} M_{sg}} \Big[c^{l}_{p,sg} m_{sg} T_{sg,sw} - c^{v}_{p,sg} m_{sg} T_{sg} - m_{sg} E_{\mathrm{evap},sg} + k_{t,sg} (T_{pc} - T_{sg}) - W_{loss,sg} \Big] \\
\dot{T}_{pr} &= \frac{1}{c_{p,pr} M_{pr}} \Big[- k_{wall} (T_{pr} - T_{pr,wall}) + W_{heat,pr} + \delta_{pr} \Big(c_{p,pc} m_{pr} (T_{pc} + \mathit{\Delta}) - c_{p,pr} m_{pr} T_{pr} \Big) \Big] \\
\dot{T}_{pr,wall} &= \frac{1}{c_{p,wall}} \Big[k_{wall} (T_{pr} - T_{pr,wall}) - W_{loss,pr} \Big]
\end{aligned}
\tag{1}
$$

where the state variables are the neutron flux N (in %), the overall mass in the primary circuit M_{pc} (in kg), the average temperature of the water in the primary circuit T_{pc} (in °C), the average secondary circuit liquid temperature T_{sg} (in °C), the pressurizer water/wall temperature T_{pr}, $T_{pr,wall}$ (in °C), and where

$$
\begin{aligned}
M_{pr} &= M_{pc} - \varphi(T_{pc}) V_{pc,0}, \qquad \varphi(T_{pc}) = c_{\varphi,0} + c_{\varphi,1} T_{pc} - c_{\varphi,2} T_{pc}^2 \\
m_{pr} &= m_{in} - m_{out} - \frac{1}{c_{p,pc} M_{pc}} \frac{\partial \varphi(T_{pc})}{\partial T_{pc}} V_{pc,0} \Big[c_{p,pc} m_{in} (T_{pc,i} - T_{pc}) + c_{p,pc} m_{out} \mathit{\Delta} \\
&\qquad + c_{\psi} N - n_{sg} k_{t,sg} (T_{pc} - T_{sg}) - W_{loss,pc} \Big].
\end{aligned}
$$

In (1) v, m_{in}, $W_{heat,pr}$ are the input variables, while $T_{pc,i}$, m_{out}, m_{sg}, M_{sg}, $T_{sg,sw}$ can be considered as disturbances. The model (1) is hybrid and nonlinear, since the equation of T_{pr} contains the switching term δ_{pr}, which is 1 if $m_{pr} > 0$ and 0 if $m_{pr} \leq 0$. Moreover, it is simple enough for the control purposes but accurate enough to capture the nonlinear, time–varying, switching nature of the plant.

The (measurable) outputs of the systems are the reactor power $W_r(N)$, the steam generator pressure p_{sg} (in kPa), the pressurizer pressure p_{pr} (in kPa), the pressurizer water level l_{pr} (in m)

$$
\begin{aligned}
W_r(N) &= c_{\psi} N \\
p_{sg} &= p_{*,T}(T_{sg}) = c_0 - c_1 T_{sg} + c_2 T_{sg}^2 \\
p_{pr} &= p_{*,T}(T_{pr}) = c_0 - c_1 T_{pr} + c_2 T_{pr}^2 \\
l_{pr}(M_{pc}, T_{pc}) &= \frac{1}{A_{pr}} \left(\frac{M_{pc}}{\varphi(T_{pc})} - V_{pc,0} \right).
\end{aligned}
\tag{2}
$$

The model parameters are reported in Table 1.

1.3 The Pressurizer Inventory and Pressure Controllers

The controller for pressurizer water level and pressure determined in [1] are dynamic controller relying only on the pressurizer wall temperature measurements. Considering a reference level $l_{pr,\mathrm{ref}}$, usually

Reactor			
Neutron flux (state variable)	N	99.3	%
Control rod position (input)	v	0	cm
Reactor power (output)	W_r	13.654×10^8	W
Constant in the reactor power equation	c_ψ	13.75×10^6	W/%
Generation time	Λ	10^{-5}	s
Rod reactivity coefficients	p_0	2.85×10^{-4}	m
	p_1	6.08×10^{-5}	m^{-1}
	p_2	1.322×10^{-4}	m^{-2}
Flux of the constant neutron source	S	2830.5	%/s
Total fraction of delayed neutrons	β	0.0064	
Average half–life	λ	0.1	s^{-1}
Primary circuit			
Overall mass in the primary circuit (state)	M_{pc}	2×10^5	kg
Water average temperature (state)	T_{pc}	281.13	°C
Inlet mass flow rate (input)	m_{in}	1.4222	kg/s
Outlet mass flow rate (disturbance)	m_{out}	2.11	kg/s
Hot leg water temperature	$T_{pc,hl}$	296.13	°C
Cold leg water temperature	$T_{pc,cl}$	266.13	°C
Inlet temperature (disturbance)	$T_{pc,i}$	258.85	°C
Specific heat at 282°C	$c_{p,pc}$	5355	J/kg/K
Heat transfer coefficient	$k_{t,sg}$	9.5296×10^6	W/K
Heat loss	$W_{loss,pc}$	2.996×10^7	W
Water nominal volume	$V_{pc,0}$	242	m^3
Water nominal mass	$M_{pc,0}$	2×10^5	kg
Differences $T_{pc,hl}-T_{pc}=T_{pc}-T_{pc,cl}$	Δ	15	°C
Pressurizer			
Water temperature (state)	T_{pr}	326.57	°C
Heating power (input)	$W_{heat,pr}$	168	kW
Water level (output)	l_{pr}	4.8	m
Pressure (output)	p_{pr}	123×10^2	kPa
Water specific heat at 325°	$c_{p,pr}$	6873.1	J/kg/K
Heat capacity of the wall	$c_{p,wall}$	6.4516×10^7	J/°C
Wall heat transfer coefficient	k_{wall}	1.9267×10^8	W/°C
Heat loss	$W_{loss,pr}$	1.6823×10^5	W
Water mass	M_{pr}	19400	kg
Vessel cross section	A_{pr}	4.52	m^2
Vessel volume	$V_{pr,vessel}$	44	m^3
Steam generator			
Average secondary circuit liquid temperature (state)	T_{sg}	257.78	°C
Secondary circ. water specific heat at 260°	$c^l_{p,sg}$	3809.9	J/kg/K
Secondary circ. vapor specific heat at 260°	$c^v_{p,sg}$	3635.6	J/kg/K
Heat loss	$W_{loss,sg}$	1.8932×10^7	W
Evaporation energy at 260°	$E_{\mathrm{evap},sg}$	1.658×10^6	J/kg
Water mass	M_{sg}	34920	kg
Water level	l_{sg}	1.850	m
Steam pressure (output)	p_{sg}	45.3×10^2	kPa
Secondary water mass flow rate (disturbance)	m_{sg}	119.31	kg/s
Secondary circ. steam mass flow rate	$m_{sg,ss}$	119.31	kg/s
Secondary circ. water mass flow rate	$m_{sg,sw}$	119.31	kg/s
Secondary circ. inlet temperature (disturbance)	$T_{sg,sw}$	220.85	°C
Number of steam generators	n_{sg}	6	
Power transferred to the steam generators	$n_{sg}W_{sg}$	13.351×10^8	W
Functions			
Saturated vapor pressure	$p_{*,T}(T)$		kPa
Coefficients for quadratic approximation	c_0	28884.78	kPa
	c_1	258.01	kPa/°C
	c_2	0.63455	$\mathrm{kPa/°C}^2$
Water density	φ(T)		$\mathrm{kg/m}^3$
Coefficients for quadratic approximation	$c_{\varphi,0}$	581.2	$\mathrm{kg/m}^3$
	$c_{\varphi,1}$	2.98	$\mathrm{kg/m}^3\mathrm{/°C}$
	$c_{\varphi,2}$	0.00848	$\mathrm{kg/m}^3\mathrm{/°C}^2$

Table 1: Model parameters

proportional to a mean value between the cold and the hot leg temperatures, with a drift to give a proper value [16]

$$l_{pr,\mathrm{ref}} = c_{r,1}(T_{pc,cl} + T_{pc,hl}) - c_{r,2} = 2c_{r,1}T_{pc} - c_{r,2}$$

the inventory control for the pressurizer water level l_{pr} is given by

$$\begin{aligned}
\dot{I}_{e_{l_{pr}}} &= l_{pr} - l_{pr,\mathrm{ref}} \\
m_{in} &= \frac{A_{pr}}{\psi(M_{pc}, T_{pc})}\Big[- \big(k_p(l_{pr} - l_{pr,\mathrm{ref}}) + k_i I_{e_{l_{pr}}}\big)\varphi^2(T_{pc}) + m^\circ_{out}\varphi(T_{pc}) \\
&\quad + \frac{1}{c_{p,pc}}\Big(\frac{2c_{r,1}}{M_{pc}}\varphi^2(T_{pc}) + \frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\Big)\Big(c_{p,pc}m^\circ_{out}\varDelta^\circ + c_\psi N - n_{sg}k_{t,sg}(T_{pc} - T_{sg}) - W^\circ_{loss,pc}\Big)\Big]
\end{aligned} \tag{3}$$

with $k_p, k_i > 0$,

$$\begin{aligned}
\psi(M_{pc}, T_{pc}) &= \varphi(T_{pc}) - (T^\circ_{pc,i} - T_{pc})\frac{\partial\varphi(T_{pc})}{\partial T_{pc}} - \frac{2c_{r,1}A_{pr}}{M_{pc}}(T^\circ_{pc,i} - T_{pc})\varphi^2(T_{pc}) \\
\varphi(T_{pc}) &= c_{\varphi,0} + c_{\varphi,1}T_{pc} - c_{\varphi,2}T^2_{pc} \\
\frac{\partial\varphi(T_{pc})}{\partial T_{pc}} &= c_{\varphi,1} - 2c_{\varphi,2}T_{pc}.
\end{aligned}$$

Due to the the nominal disturbance values $\varDelta^\circ$, $T^\circ_{pc,i}$, m°_{out}, $W^\circ_{loss,pc}$, and to their variations $\delta_\varDelta = \varDelta - \varDelta^\circ$, $\delta_{T_{pc,i}} = T_{pc,i} - T^\circ_{pc,i}$, $\delta_{m_{out}} = m_{out} - m^\circ_{out}$, $\delta_{W_{loss,pc}} = W_{loss,pc} - W^\circ_{loss,pc}$, this controller ensures practical exponential stability of the error level, i.e. $e_{l_{pr}}$, $\dot{e}_{l_{pr}}$ tend to a neighborhood of the origin of radius

$$\mu = \frac{\kappa}{\vartheta\lambda^Q_{\min}}||P||\delta_{\max}$$

where $\kappa = \max_t ||\Psi||$,

$$\Psi = \frac{1}{\varphi^2(T_{pc})}\frac{1}{A_{pr}}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}m_{in} & -\varphi(T_{pc}) - \frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\varDelta^\circ & -\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}m^\circ_{out} & \frac{1}{c_{p,pc}}\frac{\partial\varphi(T_{pc})}{\partial T_{pc}} & -\frac{\partial\varphi(T_{pc})}{\partial T_{pc}} \end{pmatrix}.$$

and $\vartheta \in (0, 1)$, P is solution of $PA + A^TP = -2Q$ for a fixed $Q = Q^T > 0$, $\lambda^Q_{\min}$ is the minimum eigenvalue of Q,

$$A = \begin{pmatrix} 0 & 1 \\ -k_i & -k_p \end{pmatrix}$$

and $\delta_{\max}$ is the maximum variation with respect to the nominal disturbance values.

As far as the pressurizer pressure controller is concerned, a first nonlinear dynamic controller can be designed transforming the pressurizer pressure reference $p_{pr,\mathrm{ref}}$ into a pressurizer water reference temperature $T_{pr,\mathrm{ref}}$, inverting the relation obtained from (2)

$$p_{pr,\mathrm{ref}} = c_0 - c_1T_{pr,\mathrm{ref}} + c_2T^2_{pr,\mathrm{ref}}$$

that can be uniquely inverted about the operating point of pressurizer temperature

$$
\begin{aligned}
T_{pr,\mathrm{ref}} &= \frac{c_1 + \sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref}})}}{2c_2} \\
\dot{T}_{pr,\mathrm{ref}} &= \frac{2\dot{p}_{pr,\mathrm{ref}}}{\sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref}})}}.
\end{aligned}
$$

The dynamic controller, which depends on the (measured) temperature $T_{pr,wall}$, but not on the (unmeasured) temperature T_{pr}

$$
\begin{aligned}
W_{heat,pr,\mathrm{ref}} = c_{p,pr}M_{pr}\Big[\dot{T}_{pr,\mathrm{ref}} &+ \frac{k_{wall}}{c_{p,pr}M_{pr}}(T_{pr,\mathrm{ref}} - T_{pr,wall,\mathrm{ref}}) \\
&- \delta_{pr}\Big(\frac{c_{p,pc}m^{\circ}_{pr}}{c_{p,pr}M_{pr}}(T_{pc} + \Delta^{\circ}) - \frac{m^{\circ}_{pr}}{M_{pr}}T_{pr,\mathrm{ref}}\Big)\Big] \\
W_{heat,pr} &= u_{e,W} + W_{heat,pr,\mathrm{ref}}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\dot{\hat{T}}_{pr} &= -\left(\frac{m^{\circ}_{pr}}{M_{pr}} + \frac{k_{wall}}{c_{p,pr}M_{pr}}\right)\hat{T}_{pr} + \frac{k_{wall}}{c_{p,pr}M_{pr}}T_{pr,wall} + \frac{1}{c_{p,pr}M_{pr}}W_{heat,pr} + \frac{c_{p,pc}m^{\circ}_{pr}}{c_{p,pr}M_{pr}}(T_{pc} + \Delta^{\circ}) \\
\dot{T}_{pr,wall,\mathrm{ref}} &= \frac{k_{wall}}{c_{p,wall}}T_{pr,\mathrm{ref}} - \frac{k_{wall}}{c_{p,wall}}T_{pr,wall,\mathrm{ref}} + k_i I_{e_{T_{pr,wall}}} - \frac{1}{c_{p,wall}}W^{\circ}_{loss,pr} \\
\dot{I}_{e_{T_{pr,wall}}} &= T_{pr,wall} - T_{pr,wall,\mathrm{ref}} \\
W_{heat,pr} &= -k_{wall}(T_{pr,wall} - T_{pr,wall,\mathrm{ref}}) - \frac{c_{p,pr}}{c_{p,wall}}k_{wall}M_{pr}B^T P\begin{pmatrix} I_{e_{T_{pr,wall}}} \\ T_{pr,wall} - T_{pr,wall,\mathrm{ref}} \end{pmatrix} \\
&\quad c_{p,pr}M_{pr}\Big[\dot{T}_{pr,\mathrm{ref}} + \frac{k_{wall}}{c_{p,pr}M_{pr}}(T_{pr,\mathrm{ref}} - T_{pr,wall,\mathrm{ref}}) - \delta_{pr}\Big(\frac{c_{p,pc}m^{\circ}_{pr}}{c_{p,pr}M_{pr}}(T_{pc} + \Delta^{\circ}) - \frac{m^{\circ}_{pr}}{M_{pr}}T_{pr,\mathrm{ref}}\Big)\Big]
\end{aligned}
\tag{5}
$$

ensures the practical exponential stability of the error temperatures, with $k_i > 0$, $k_p = \frac{k_{wall}}{c_{p,wall}} > 0$, $T_{pr,wall,\mathrm{ref}}(0) = T_{pr,\mathrm{ref}}(0) - W^{\circ}_{loss,pr}/k_{wall}$, $M_{pr} = M_{pc} - \varphi(T_{pc})V_{pc,0}$,

$$
\begin{aligned}
m^{\circ}_{pr} = m_{in} - m^{\circ}_{out} - \frac{1}{c_{p,pc}M_{pc}}\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}V_{pc,0}\Big[&c_{p,pc}m_{in}(T^{\circ}_{pc,i} - T_{pc}) + c_{p,pc}m^{\circ}_{out}\Delta^{\circ} \\
&+ c_{\psi}N - n_{sg}k_{t,sg}(T_{pc} - T_{sg}) - W^{\circ}_{loss,pc}\Big]
\end{aligned}
$$

and $P = P^T > 0$ solution of the Lyapunov equation

$$
PA + A^T P = -2Q, \qquad A = \begin{pmatrix} 0 & 1 \\ -k_i & -\frac{k_{wall}}{c_{p,wall}} \end{pmatrix}
$$

for a fixed $Q = Q^T > 0$, and $B = (\,0 \quad 1\,)^T$.

An alternative (dynamic switching nonlinear) controller for the pressurizer pressure is given by

$$\begin{aligned}
\dot{I}_{e_{p_{pr}}} &= c_0 - c_1\hat{T}_{pr} + c_2\hat{T}_{pr}^2 - p_{pr,\text{ref}} \\
\dot{\xi} &= \hat{T}_{pr} - T_{pr,wall} - \frac{1}{k_{wall}}W^{\circ}_{loss,pr} - \frac{1}{k}\frac{1}{c_{p,pr}M_{pr}}C_{pr} \\
\hat{T}_{pr} &= \kappa\Big(\frac{c_{p,wall}}{k_{wall}}T_{pr,wall} - \xi\Big) \\
C_{pr} &= \frac{c_{p,pr}M_{pr}}{-c_1 + 2c_2\hat{T}_{pr}}\Big(\dot{p}_{pr,\text{ref}} - K_p\big(c_0 - c_1\hat{T}_{pr} + c_2\hat{T}_{pr}^2 - p_{pr,\text{ref}}\big) - K_i I_{e_{p_{pr}}}\Big) \\
W_{heat,pr} &= k_{wall}(\hat{T}_{pr} - T_{pr,wall}) + C_{pr} + \delta_{pr}\big(c_{p,pr}m^{\circ}_{pr}\hat{T}_{pr} - c_{p,pc}m^{\circ}_{pr}(T_{pc} + \mathit{\Delta}^{\circ})\big)
\end{aligned} \tag{6}$$

where $\kappa, K_p, K_i > 0$, $I_{e_{p_{pr}}}(0) = 0$, $\xi(0) = -\hat{T}_{pr}(0)/k + c_{p,wall}T_{pr,wall}(0)/k_{wall}$ and, as already defined, δ_{pr} is 1 if $m_{pr} > 0$ and 0 otherwise.

1.4 Digital Implementation of the Control Laws

We have already noted that the implementation of control laws by means of digital devices determines a disturbances acting on the feedback system. In this section we consider a further aspect, the digital implementation of derivatives. In fact, in (3), (5), (6) the derivative $\dot{I}_{e_{l_{pr}}}$, $\dot{\hat{T}}_{pr}$, $\dot{T}_{pr,wall,\text{ref}}$, $\dot{I}_{e_{T_{pr,wall}}}$, $\dot{I}_{e_{p_{pr}}}$, $\dot{\xi}$ have to be implemented numerically. Let us consider a feedback system with $u = \alpha(x)$ a certain control law applied to control the system $\dot{x} = \bar{f}(x, u)$. The closed–loop dynamics is hence

$$\dot{x} = \bar{f}(x, \alpha(x)) := f(x)$$

A simple method to approximate by a digital computer the real time solution of this differential equation is to consider the Euler's rule, relying on the definition of derivative

$$\dot{x} = \lim_{\mathit{\Delta} t \to 0} \frac{\mathit{\Delta} x}{\mathit{\Delta} t}.$$

Here $\mathit{\Delta} x$ is the change in x over the time interval $\mathit{\Delta} t$. Even if $\mathit{\Delta} t$ does not tends to zero, the previous relation is approximately true

$$\dot{x}|_{t_k} \simeq \frac{x_{k+1} - x_k}{\delta}$$

where $\dot{x}|_{t_k}$ is the derivative of $x(t)$ calculated at time $t_k = k\delta$, $\delta = t_{k+1} - t_k$ is the sampling interval, $k \in \mathbb{Z}$, and $x_k = x(k\delta)$, $x_{k+1} = x((k+1)\delta)$ are the value of $x(t)$ at $t_k = k\delta$, $t_{k+1} = (k+1)\delta$. This is the so called forward rectangular rule, and leads to the following expression

$$x_{k+1} = x_k + \delta f(x_k).$$

There exists also a backward rectangular rule

$$\dot{x}(k) \simeq \frac{x_k - x_{k-1}}{\delta}$$

leading to

$$x_{k+1} = x_k + \delta f(x_{k+1}).$$

Another method is the trapezoidal rule, where eventually one gets

$$x_{k+1} = x_k + \frac{\delta}{2}\Big(f(x_k) + f(x_{k+1})\Big).$$

These approximations can be used in place of the derivatives that appear in the controller differential equations, to obtain difference equations repetitively solved with time steps of length δ. In the following we will consider the Euler's rule, for the sake of clarity, but the same arguments can be used with the other integration methods.

When dealing with the classical proportional, integral, and derivative control actions

$$u_p(t) = k_p e(t), \qquad u_i(t) = \frac{k_p}{T_i}\int_0^t e(\tau)\,d\tau, \qquad u_d(t) = k_p T_d \dot{e}(t)$$

with k_p the proportional gain, T_i the integral time, T_d the derivative time, they can be approximated with the following algebraic relations, which can be implementable with digital computers

$$u_{p,k} = k_p e_k, \qquad u_{i,k} = u_{i,k-1} + \frac{k_p}{T_i}\delta e_k, \qquad u_{d,k} = k_p T_d \frac{e_k - e_{k-1}}{\delta}$$

where $u_{i,k}$, $u_{d,k}$ are the results of the forward rule of the Euler approximation. Usually, these control actions are used together and their combination need to be done carefully. Hence, considering the Laplace transform for a classical (linear) PID controller

$$G(s) = \frac{u(s)}{e(s)} = k_p\left(1 + \frac{1}{T_i s} + T_d s\right)e(s)$$

so that

$$su(s) = k_p\left(s + \frac{1}{T_i} + T_d s^2\right)e(s)$$

and

$$\dot{u} = k_p\left(\dot{e} + \frac{1}{T_i}e + T_d\ddot{e}\right)$$

the Euler's method, applied twice for $\ddot{e}$, gives

$$u_k = u_{k-1} + k_p\left[\left(1 + \frac{\delta}{T_i} + \frac{T_d}{\delta}\right)e_k - \left(1 - 2\frac{T_d}{\delta}\right)e_{k-1} + \frac{T_d}{\delta}e_{k-2}\right].$$

For linear systems (and controllers) with bandwidths of a few Hz, sample rates are often on the order of 100 Hz, so that δ is on the order of 10^{-2} s, and errors from the approximation is quite small. An empirical rule is that the sample rate should be faster that 30 times the bandwidth in order to assure that the digital controller can be made to closely match the performance of the continuous controller. Except

Shannon rule, regarding the aliasing problem, there is not a theoretic rule to fix δ in order to ensure this close match.

The absence of a systematic rule to choose the sampling period δ is even more problematic for nonlinear systems. In fact, it is well–known that nonlinear systems can experience a finite escape times, which roughly means that the state can go to infinity in finite time intervals [8]. This is a great difference with linear systems, where the state can go to infinity asymptotically, namely when time goes to infinity. As a result, from a conservative point of view, the controllers need to be implemented with the fastest sampling time.

With this important observation in mind, the controllers (3), (5), (6), containing PI terms, can be implemented as follows.

Digital implementation of the pressurizer water level controller (3)

$$\begin{aligned} I_{e_{l_{pr}},k+1} &= I_{e_{l_{pr}},k} + \delta(l_{pr,k} - l_{pr,\mathrm{ref},k}) \\ m_{in,k} &= \frac{A_{pr}}{\psi(M_{pc,k}, T_{pc,k})}\Big[-\big(k_p(l_{pr,k} - l_{pr,\mathrm{ref},k}) + k_i I_{e_{l_{pr}},k}\big)\varphi^2(T_{pc,k}) + m^\circ_{out}\varphi(T_{pc,k}) \\ &\quad + \frac{1}{c_{p,pc}}\Big(\frac{2c_{r,1}}{M_{pc,k}}\varphi^2(T_{pc,k}) + \frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\Big|_k\Big)\Big(c_{p,pc}m^\circ_{out}\Delta^\circ + c_\psi N_k - n_{sg}k_{t,sg}(T_{pc,k} - T_{sg,k}) - W^\circ_{loss,pc}\Big)\Big] \end{aligned} \tag{7}$$

with $k_p, k_i > 0$,

$$\begin{aligned} \psi(M_{pc,k}, T_{pc,k}) &= \varphi(T_{pc,k}) - (T^\circ_{pc,i} - T_{pc,k})\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\Big|_k - \frac{2c_{r,1}A_{pr}}{M_{pc,k}}(T^\circ_{pc,i} - T_{pc,k})\varphi^2(T_{pc,k}) \\ \varphi(T_{pc,k}) &= c_{\varphi,0} + c_{\varphi,1}T_{pc,k} - c_{\varphi,2}T^2_{pc,k} \\ \frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\Big|_k &= c_{\varphi,1} - 2c_{\varphi,2}T_{pc,k}. \end{aligned}$$

Digital implementation of the pressurizer pressure controller (5)

$$
\begin{aligned}
\hat{T}_{pr,k+1} &= \hat{T}_{pr,k} + \delta\Bigg[-\left(\frac{m^{\circ}_{pr,k}}{M_{pr,k}} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}}\right)\hat{T}_{pr,k} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}}T_{pr,wall,k} \\
&\qquad + \frac{1}{c_{p,pr}M_{pr,k}}W_{heat,pr,k} + \frac{c_{p,pc}m^{\circ}_{pr,k}}{c_{p,pr}M_{pr,k}}(T_{pc,k} + \mathit{\Delta}^{\circ})\Bigg] \\
T_{pr,wall,\mathrm{ref},k+1} &= T_{pr,wall,\mathrm{ref},k} + \delta\left(\frac{k_{wall}}{c_{p,wall}}T_{pr,\mathrm{ref},k} - \frac{k_{wall}}{c_{p,wall}}T_{pr,wall,\mathrm{ref},k} + k_i I_{e_{T_{pr,wall}},k} - \frac{1}{c_{p,wall}}W^{\circ}_{loss,pr}\right) \\
I_{e_{T_{pr,wall}},k+1} &= I_{e_{T_{pr,wall}},k} + \delta\left(T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k}\right) \\
W_{heat,pr,k} &= -k_{wall}(T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k}) - \frac{c_{p,pr}}{c_{p,wall}}k_{wall}M_{pr,k}B^{T}P\begin{pmatrix} I_{e_{T_{pr,wall}},k} \\ T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k}\end{pmatrix} \\
&\quad + c_{p,pr}M_{pr,k}\Bigg[\dot{T}_{pr,\mathrm{ref}}\Big|_{k} + \left(\frac{m^{\circ}_{pr,k}}{M_{pr,k}} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}}\right)T_{pr,\mathrm{ref},k} - \frac{k_{wall}}{c_{p,pr}M_{pr,k}}T_{pr,wall,\mathrm{ref},k} \\
&\qquad - \frac{c_{p,pc}m^{\circ}_{pr,k}}{c_{p,pr}M_{pr,k}}(T_{pc,k} + \mathit{\Delta}^{\circ})\Bigg] \\
T_{pr,\mathrm{ref},k} &= \frac{c_1 + \sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref},k})}}{2c_2}, \qquad \dot{T}_{pr,\mathrm{ref}}\Big|_{k} = \frac{2\dot{p}_{pr,\mathrm{ref},k}}{\sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref},k})}}
\end{aligned}
\tag{8}
$$

with $T_{pr,wall,\mathrm{ref},0} = T_{pr,\mathrm{ref},0} - W^{\circ}_{loss,pr}/k_{wall}$, $M_{pr,k} = M_{pc,k} - \varphi(T_{pc,k})V_{pc,0}$,

$$
\begin{aligned}
m^{\circ}_{pr,k} = m_{in,k} - m^{\circ}_{out} - \frac{1}{c_{p,pc}M_{pc,k}}\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\bigg|_{k} V_{pc,0}\Big[&c_{p,pc}m_{in,k}(T^{\circ}_{pc,i} - T_{pc,k}) + c_{p,pc}m^{\circ}_{out}\mathit{\Delta}^{\circ} \\
&+ c_{\psi}N_k - n_{sg}k_{t,sg}(T_{pc,k} - T_{sg,k}) - W^{\circ}_{loss,pc}\Big].
\end{aligned}
$$

Digital implementation of the pressurizer pressure controller (6)

$$
\begin{aligned}
I_{e_{p_{pr}},k+1} &= I_{e_{p_{pr}},k} + \delta\left(c_0 - c_1\hat{T}_{pr} + c_2\hat{T}^2_{pr} - p_{pr,\mathrm{ref}}\right) \\
\xi_{k+1} &= \xi_k + \delta\left(\hat{T}_{pr} - T_{pr,wall} - \frac{1}{k_{wall}}W^{\circ}_{loss,pr} - \frac{1}{k}\frac{1}{c_{p,pr}M_{pr}}C_{pr}\right) \\
\hat{T}_{pr,k} &= \kappa\left(\frac{c_{p,wall}}{k_{wall}}T_{pr,wall,k} - \xi_k\right) \\
C_{pr,k} &= \frac{c_{p,pr}M_{pr,k}}{-c_1 + 2c_2\hat{T}_{pr,k}}\left(\dot{p}_{pr,\mathrm{ref},k} - K_p\left(c_0 - c_1\hat{T}_{pr,k} + c_2\hat{T}^2_{pr,k} - p_{pr,\mathrm{ref},k}\right) - K_i I_{e_{p_{pr}},k}\right) \\
W_{heat,pr,k} &= k_{wall}(\hat{T}_{pr,k} - T_{pr,wall,k}) + C_{pr,k} + \delta_{pr}\left(c_{p,pr}m^{\circ}_{pr,k}\hat{T}_{pr,k} - c_{p,pc}m^{\circ}_{pr,k}(T_{pc,k} + \mathit{\Delta}^{\circ})\right)
\end{aligned}
\tag{9}
$$

where $I_{e_{p_{pr}},0} = 0$, $\xi_0 = -\hat{T}_{pr,0}/\kappa + c_{p,wall}T_{pr,wall,0}/k_{wall}$.

2 Self Triggered Robust Strategies for Optimal Implementations of Control Laws on Digital Devices

The implementation of controllers with digital devices presents many advantages, but at the same time poses some issues. We have already mentioned about the fact that the implementation of control laws with zero order holders, commonly used for the digital implementation of control laws, introduce a delay and hence could bring to unstable behaviors. We have also mentioned the possibility of determining a region of practical stability of the control system. In this section we want to be more specific and consider a technique, called self triggered control, determining the sampling time so that the control system performances are preserved. The obvious property to be preserved is the asymptotic stability, which in turn means that the plant is operating safely.

One important aspect is the design of the digital controller so that the control system recovers, at least in first approximation, the same behavior of a system controlled with a continuous time controller. In fact, many authors propose nonlinear digital controllers reproducing the performances of a certain continuous controller, viz emulating the behavior of the continuous controller [34] [29], [30]. This a very popular technique relies on the simple consideration that when the (fixed) sampling period is short enough, one regains the continuous behavior. Other authors aim at designing the digital controller directly in the digital setting, imposing control performances on the digital model of the system, although nonlinear systems cannot be discretized exactly in closed form, in general. In both cases, a relevant problem arises: the determination of the sampling period. From a theoretical point of view, the sampling period is usually considered constant, namely the new control value is computed periodically at each sampling time. This helps, from a mathematical point of view, the analysis of the sampled nonlinear system, and gives some mathematical tools to solve the design problem. However, it is clear that a better solution should be that of calculating the controller only “when necessary”. This clearly complicates the problem from an analytical point of view. But there are also practical aspects that push to deal with variable sampling. A first aspect is that a constant sampling is quite inefficient (in terms of CPU usage, communication bandwidth, energy, etc.), since it has to be considered the worst–case scenario. In fact, since the system dynamics are nonlinear, one has to ensure good performance for all the operative points. Incidentally, this reveals the need of criteria to fix the sampling time value, that otherwise has to be fixed using (possibly extensive) simulations, or considering empirical rules (20 times the system bandwidth [20]). A further aspect is that many digital controller perform various tasks at the same time. This is quite typical, especially in the case of embedded systems. An example is the electronic central unit in an automobile, which has to manage different tasks, with different priority. Their scheduling is clearly of critical importance to prevent negative coupling effects of lower priority tasks when computing high level tasks, such as attitude control laws. Another important example are the networked systems, where not only the processor time is a resource to be optimized, but also the available communication bandwidth is limited. In wireless sensor networks, furthermore, an important issue is the minimization of the power consumption, in order to augment the life span of the network. In all these applications, the energy consumption is related to the frequency of measurements and transmission over the network. It is clear that in these cases measurement/computation/actuation data transmission should be minimized and should occur only "when necessary".

Among the various techniques proposed to face this problem, the event triggered technique seems to be promising [21], [31], [14], [22]. This technique formalize the statement "when necessary": the measurement/computation/actuation data transmission event occurs when the state of the system assumes certain values. Clearly, this technique requires the continuous measurement of the state. To circumvent this drawback, self–triggered techniques have been proposed. In this case the controller determines its next execution time, and does not require continuous measurements of the state. In particular, when the stabilization of the system origin is considered, this event is triggered only when the asymptotic stability property, as formalized by the Lyapunov approach, can be lost [14], [4], [13], [36], [37], [5], [18], [6]. This approach can be also generalized to a weaker property such as safety [6].

Self triggered control strategies have been introduced in [33], where a heuristic rule is provided to self–trigger the next execution time of a control task on the basis of the last measurement of the state. In [11], [12], a robust self triggered strategy is proposed, which guarantees that the $\mathcal{L}_2$ gain of a linear time invariant system is kept under a given threshold. In [13] a self triggered strategy distributed over a wireless sensor network is proposed for linear time invariant systems.

In [14] sufficient conditions for the existence of a stabilizing event–triggered control strategy are given for nonlinear systems. In [5] the authors propose a self–triggered emulation of the event–triggered control strategy proposed in [14]. In particular a methodology for the computation of the next execution time as a function of the last sample is presented, under a homogeneity condition.

In this deliverable, a methodology for the computation of the next execution time is proposed, based on polynomial approximations of Lyapunov functions, and relying on the assumption that the nonlinear differential equations and the control law are C^ℓ functions, with ℓ sufficiently large. This assumption is verified in the present case. The next sampling time is also computed in the presence of bounded sensing/computation/actuation delays and of norm–bounded parameter uncertainties and disturbances. Moreover, under weaker conditions than those used in [14], it is proved the existence of a self triggered strategy keeping the state in a "safe set" arbitrarily close to the equilibrium point, and a methodology for computing the next execution time is provided. Fixed a δ–ball of the equilibrium point and a disturbance that is upper bounded in norm by a class $\mathcal{K}$ function $\nu(\delta)$, a methodology is presented for the computation of the next execution time that depends on the δ boundary defining the safe set.

2.1 Problem Formulation

Consider a generic nonlinear system

$$\dot{x} = f(x, u, \mu, d) \tag{10}$$

where $x \in \mathcal{D}_x \subset r^n$, $\mathcal{D}_x$ a domain containing the origin, $u \in \mathcal{D}_u \subset r^p$, μ is a parameter uncertainty vector varying in a compact set $\mathcal{D}_\mu \subset r^r$, with $0 \in \mathcal{D}_\mu$, d is an external bounded disturbance vector taking values in a compact set $\mathcal{D}_d \subset r^s$, with $0 \in \mathcal{D}_d$. In the following, we may refer to (10) as the perturbed system. Furthermore, we define the nominal (or "unperturbed") system associated to the "perturbed" system (10) as

$$\dot{x} = f_0(x, u) \doteq f(x, u, 0, 0). \tag{11}$$

Given a state feedback control law $\kappa\colon \mathcal{D}_x \to \mathcal{D}_u$, the closed loop perturbed system is

$$\dot{x} = f(x, \kappa(x), \mu, d) \tag{12}$$

and the closed loop nominal system is

$$\dot{x} = f_0(x, \kappa(x)). \tag{13}$$

We will denote by $x(t)$, $t \geq t_0$, the solution of the closed loop system (12) (or (13), according to the context), with initial condition $x_0 = x(t_0)$. Given a state feedback control law κ it is well–known that, if κ locally stabilizes the origin of system (13) and if $f_0(x, \kappa(x)) \in C^\ell(\mathcal{D}_x)$, $\ell > 1$ integer, then there exists a Lyapunov function $V(x)$ of class $C^1(\mathcal{D}_x)$ such that

$$\begin{aligned} \alpha_1(\|x\|) \leq V(x) &\leq \alpha_2(\|x\|) \\ \frac{\partial V(x)}{\partial x} f_0(x, \kappa(x)) &\leq -\alpha_3(\|x\|) \\ \left\| \frac{\partial V(x)}{\partial x} \right\| &\leq \alpha_4(\|x\|) \end{aligned} \tag{14}$$

with $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathcal{K}$ [8], [10], [35].

Moreover, given a state feedback control law κ, we say that system (12) is safe with respect to the set $\mathcal{S} \subseteq \mathcal{D}_x$ for the time interval $\mathcal{T} \subseteq r_0^+$, if $x(t) \in \mathcal{S}$, $\forall t \in \mathcal{T}$.

The feedback control signal $u(t) = \kappa(x(t))$ requires continuous measurements of the state of the system. We assume that state measurements are available at sampling times t_k, which define a sequence $\mathcal{I} = \{t_k\}_{k \geq 0}$, and that the applied control is

$$u_{\mathcal{I}}(t) = \begin{cases} 0 & \forall t \in [t_0, t_0 + \varDelta_0) \\ \kappa(x_k) & \forall t \in [t_k + \varDelta_k, t_{k+1} + \varDelta_{k+1}),\ k \geq 0 \end{cases} \tag{15}$$

where $\{\varDelta_k\}_{k \geq 0}$ is a sequence of delays, due to the transmission time from the sensor to the controller, the computation time, and the transmission time from the controller to the actuator. On the basis of this assumption, we address the following problems.

Problem 1 *Given a system (11), and a state feedback control law κ such that the origin of (13) is asymptotically stable with region of attraction $\varOmega \subset \mathcal{D}_x$ containing the origin, determine a function $\tau_s : \mathcal{D}_x \to [\tau_{\min}, \infty), \tau_{\min} > 0$ and a maximum allowed delay $\varDelta_{\max} \in [0, \tau_{\min}]$ such that if the sequence of sampling instants $\mathcal{I}$ is inductively defined by*

$$t_{k+1} = t_k + \tau_s(x_k) \tag{16}$$

and if the delays are such that

$$\varDelta_k \in [0, \varDelta_{\max}), \quad \forall\, k \geq 0, \tag{17}$$

then the origin of the closed loop system (13) with control input signal $u_{\mathcal{I}}(t)$ as in (15) is asymptotically stable with region of attraction $\varOmega$. ◇

Problem 2 *Given a system (10) (resp. (11)), and a state feedback control law κ such that the origin of (12) (resp. (13)) is asymptotically stable with region of attraction $\varOmega \subset \mathcal{D}_x$ containing the origin, and an arbitrary safe set $\mathcal{B}_\delta = \{x \in r^n \mid \|x\| < \delta\} \subset \varOmega$, $\delta > 0$, determine τ_s and $\varDelta_{\max}$ as defined in Problem 1 such that if $\mathcal{I}$ is inductively defined by (16) and if $\varDelta_k$ satisfies (17), then the closed loop system (12) with control input signal $u_{\mathcal{I}}(t)$ as in (15) is safe with respect to $\mathcal{B}_\delta$ for the time interval $[t_0, \infty)$.* ◇

In Problems 1 and 2, the function τ_s is used to determine the next sampling instant as a function of the current measurement of the system. The purpose is to obtain a self triggered control system that is robust with respect to delays bounded by a design parameter $\varDelta_{\max}$. By choosing the next sampling instant t_{k+1} as a function of the current measurement at time t_k, we perform sampling only when needed for guaranteeing asymptotic stability or safety. The aim is to determine a sampling instant sequence $\mathcal{I}$

such that the intersampling time $t_{k+1} - t_k$ is as large as possible, in order to reduce transmission power of the sensing and actuation data transmissions, and to reduce the CPU effort due to the computation of the control.

As a comparison of the above definitions with the concepts of Maximally Allowable Transmission Interval (MATI) and Maximally Allowable Delay (MAD) introduced in [7], we can interpret $\varDelta_{\max}$ as the (*global*) MAD, and $t_{k+1} - t_k = \tau_s(x_k) - t_k$ as the (*local*) MATI of the system in the time interval $[t_k, t_{k+1}]$ on the basis of the measurement $x_k = x(t_k)$ of the state $x(t)$ at $t = t_k$.

2.2 Self Triggered Stabilizing Control

The results developed in this section address Problem 1 for system (11), and are based on the following assumptions, which are weaker than those required in [5] (viz., homogeneity of the closed loop dynamics) to compute the next sampling time as a function τ_s of the current state of the system.

Assumption 1 *Assume that*

1. $f_0 \in C^\ell(\mathcal{D}_x \times \mathcal{D}_u)$*, with* ℓ *a positive integer sufficiently large;*
2. *There exists a nonempty set* $\mathcal{U}$ *of state feedback laws* $\kappa\colon \mathcal{D}_x \to \mathcal{D}_u$*, such that* $\kappa \in C^\ell(\mathcal{D}_x)$ *and the origin of (13) is asymptotically stable, with region of attraction a certain compact* $\varOmega \subset \mathcal{D}_x$ *containing the origin;*
3. *The functions* $\alpha_3, \alpha_4 \in \mathcal{K}$ *in (14) are such that* α_3^{-1}, α_4 *are Lipschitz.* ◇

The assumption of existence of a stabilizing control (i.e. non–emptiness of the set $\mathcal{U}$) is not restrictive, since if the nominal system cannot be stabilized using continuous time measurement and actuation, then it is clear that the nominal system cannot be stabilized using a digital control with zero–order holders. The main limitation of Assumption 1 is the Lipschitz condition on $\alpha_3^{-1}(\cdot)$ and $\alpha_4(\cdot)$. We will show how to weaken this assumption in Section 2.3, which will be devoted to safety control.

Theorem 1 *Under Assumption 1, Problem 1 is solvable for system (11), and the function* τ_s *can be iteratively computed as a function of the current state of the system and the maximum allowable delay* $\varDelta_{\max}$. ◇

Proof: We first prove the result for $\varDelta_k = 0$. Since $\mathcal{U}$ is not empty, by Assumption 1, we pick a state feedback control law $\kappa \in \mathcal{U}$. Since $f_0(x, \kappa(x)) \in C^\ell(\mathcal{D}_x)$ with $\ell > 1$, there exists a Lyapunov candidate

$V(x)$ that satisfies (14). Choose $r > 0$ such that the ball $B_r = \{x \in \Omega : \|x\| \leq r\} \subset \Omega$. For $x_k \in B_r$,

$$\begin{aligned}\dot{V} &= \frac{\partial V}{\partial x} f_0(x, \kappa(x_k)) = \frac{\partial V}{\partial x} f_0(x, \kappa(x)) + \frac{\partial V}{\partial x}\Big(f_0(x, \kappa(x_k)) - f_0(x, \kappa(x))\Big) \\ &\leq -\alpha_3(\|x\|) + \alpha_4(\|x\|)\|d_h\|\end{aligned} \tag{18}$$

where

$$d_h = f_0(x(t), \kappa(x_k)) - f_0(x(t), \kappa(x(t)))$$

can be considered as a perturbation due to the holding.

Under Assumption 1, there exists a $\delta_k > 0$ such that $\dot{x} = f(x, \kappa(x_k))$ has a unique solution over $[t_k, t_k + \delta_k]$. Hence, we can expand the components $d_{h,i}$ of d_h in Taylor series. Consider the i^{th} component $d_{h,i}$, $i = 1, \cdots, n$, of the n–dimensional vector d_h. One can expand each component in Taylor series with respect to $t \in [t_k, t_k + \delta_k]$, on the right of t_k, up to the 2^{nd} term, with Lagrange remainder of the 3^{rd} term

$$d_{h,i} = \varphi_{1,i}(x_k)(t - t_k) + \varphi_{2,i}(\bar{x}_i, x_k)(t - t_k)^2 \tag{19}$$

where

$$\varphi_{1,i}(x_k) = d_{h,i}\Big|_{x(t)=x_k}, \ \varphi_{2,i}(\bar{x}_i, x_k) = \frac{1}{2}\left.\frac{\mathrm{d}_+^2 d_{h,i}}{\mathrm{d}t^2}\right|_{x(t)=\bar{x}_i}$$

where $\frac{\mathrm{d}_+^n(\cdot)}{\mathrm{d}t^n}$ denotes the n–th right derivative. According to Taylor theorem with Lagrange remainder, there exists $\bar{t}_i \in [t_k, t]$, with $\bar{x}_i = x(\bar{t}_i)$, $i = 1, \cdots, n$, such that the equality (19) holds. Hence,

$$\|d_h\| \leq \|\varphi_1(x_k)\|(t - t_k) + \|\varphi_2(\bar{x}, x_k)\|(t - t_k)^2$$

where $\bar{x} \doteq (\bar{x}_1, \cdots, \bar{x}_n)$ and

$$\begin{aligned}\varphi_1(x_k) &\doteq \Big(\varphi_{1,1}(x_k), \cdots, \varphi_{1,n}(x_k)\Big)^T \\ \varphi_2(\bar{x}, x_k) &\doteq \Big(\varphi_{2,1}(\bar{x}_1, x_k), \cdots, \varphi_{2,n}(\bar{x}_n, x_k)\Big)^T.\end{aligned}$$

Consider the set $\Omega_{V(x_k)} \doteq \{x \in \Omega : V(x) \leq V(x_k)\}$, and define

$$M_1(x_k) \doteq \|\varphi_1(x_k)\|, \qquad M_2(x_k) \doteq \max_{\bar{x} \in \Omega_{V(x_k)}} \|\varphi_2(\bar{x}, x_k)\|.$$

Since $f, \kappa \in C^\ell$ and $\Omega_{V(x_k)}$ is compact, then $M_1(x_k)$ is finite for any $x_k \in \Omega_{V(x_k)}$, and $M_2(x_k) \in r^+$ exists and is finite for any $x_k \in \Omega_{V(x_k)}$.

Note that there exists a time interval $[t_k, t_{k+1} < t_k + \delta_k]$ such that

$$\alpha_4(\|x\|)\|d_h\| \leq \vartheta\alpha_3(\|x\|) \tag{20}$$

is satisfied for a fixed $\vartheta \in (0, 1)$. In fact, (20) is satisfied if

$$\alpha_3^{-1}\left(\frac{1}{\vartheta}\alpha_4(\|x\|)\Big(M_1(x_k)(t - t_k) + M_2(x_k)(t - t_k)^2\Big)\right) \leq \|x\|.$$

Since α_3^{-1} and α_4 are Lipschitz, then equation (20) is satisfied if

$$\frac{1}{\vartheta} L_{\alpha_3^{-1}} L_{\alpha_4} \|x\| \Big(M_1(x_k)(t - t_k) + M_2(x_k)(t - t_k)^2 \Big) \leq \|x\|$$

where $L_{\alpha_3^{-1}}, L_{\alpha_4} > 0$ are the Lipschitz constants of α_3^{-1}, α_4, respectively. The above equation directly implies that (20) is satisfied if

$$M_1(x_k)(t - t_k) + M_2(x_k)(t - t_k)^2 \leq \frac{\vartheta}{L_{\alpha_3^{-1}} L_{\alpha_4}}. \tag{21}$$

Hence, if we define

$$\tau_s(x_k) \doteq \max \{ t - t_k : \text{ (21) is satisfied for each } t - t_k \in [0, \tau_s(x_k)] \}$$

$$\tau_{\min} \doteq \min_{x_k \in \Omega_{V(x_k)}} \tau_s(x_k)$$

and we choose $t_{k+1} = t_k + \tau_s(x_k)$, then $\dot{V}(t) \leq -(1 - \vartheta)\alpha_3(\|x\|) < 0$ for all $t \in [t_k, t_{k+1}]$ and for all $k \geq 0$. This implies that the origin is asymptotically stable. Equation (21) is a second degree inequality in the form $ay^2 + by \leq c$, where a, b are non–negative and upper bounded for each $x_k \in \mathcal{D}_x$, and c is strictly positive and upper bounded. This trivially implies that $\tau_s(x_k)$ is strictly positive for each $x_k \in \Omega_{V(x_k)}$, and thus $\tau_{\min}$ is strictly positive as well. In this way, $\tau_s(\cdot)$ remains defined iteratively for each $k \geq 0$. This completes the proof for $\Delta_k = 0$.

For the case of $\Delta_k > 0$, following the same reasoning

$$\begin{aligned} \dot{V}(t) &= \frac{\partial V}{\partial x} f_0(x(t), \kappa(x_k)) = \frac{\partial V}{\partial x} f_0(x, \kappa(x)) + \frac{\partial V}{\partial x}(d_h + d_{\Delta_k}) \\ &\leq -\alpha_3(\|x\|) + \alpha_4(\|x\|)\|d_h\| + \alpha_4(\|x\|)\|d_{\Delta_k}\| \end{aligned}$$

for $t \geq t_k + \Delta_k$ where

$$d_h = f_0(x(t), \kappa(x(t_k + \Delta_k))) - f_0(x, \kappa(x))$$

$$d_{\Delta_k} = f_0(x(t), \kappa(x_k)) - f_0(x(t), \kappa(x(t_k + \Delta_k)))$$

can be considered as perturbations due to the holding and to the sensing/computation/actuation delay. Since also the solution $x(t)$ is Lipschitz, as well as f_0 and κ, then

$$\|d_{\Delta_k}\| \leq M_3 \Delta_k, \quad M_3 = L_{f_0} L_\kappa L_x$$

where L_{f_0}, L_κ, L_x are the Lipschitz constants of f_0, κ, x. Proceeding for d_h as in the previous case, we conclude that (20) is satisfied if

$$M_1(x_k)(t - t_k) + M_2(x_k)(t - t_k)^2 + M_3 \Delta_k \leq \frac{\vartheta}{L_{\alpha_3^{-1}} L_{\alpha_4}}. \tag{22}$$

Setting $\vartheta = \vartheta_1 + \vartheta_2$, with $\vartheta_1, \vartheta_2 \in (0, 1)$, equation (22) implies that the stability condition (20) holds if

$$M_1(x_k)(t - t_k) + M_2(x_k)(t - t_k)^2 \leq \frac{\vartheta_1}{L_{\alpha_3^{-1}} L_{\alpha_4}}, \tag{23}$$

and

$$\varDelta_k \leq \varDelta_{\max} \doteq \frac{\vartheta_2}{M_3 L_{\alpha_3^{-1}} L_{\alpha_4}}. \tag{24}$$

Defining

$$\tau_s(x_k) \doteq \max\{t - t_k : \text{ (23) is satisfied for each } t - t_k \in [0, \tau_s(x_k)]\} - \varDelta_{\max}$$

$$\tau_{\min} \doteq \min_{x_k \in \Omega_{V(x_k)}} \tau_s(x_k)$$

and if we choose $t_{k+1} = t_k + \tau_s(x_k)$, then $\dot{V}(t) \leq -(1 - \vartheta)\alpha_3(\|x\|) < 0$ for all $t \in [t_k + \varDelta_k, t_{k+1} + \varDelta_{\max}]$ and for all $k > 0$. This ensures the asymptotic stability of the origin. $\varDelta_{\max}$ is non-negative, and for ϑ_2 sufficiently small $t_{k+1} - t_k = \tau_s(x_k) > \varDelta_{\max} \geq 0$ for each $x_k \in \Omega_{V(x_k)}$. Therefore, $\tau_{\min}$ is strictly positive as well. This completes the proof. ■

Remark 1 *It is worth noting that* $\tau_{\min} > 0$*, as shown in the proof, implies that the time interval between two sampling instants is lower bounded by the minimum sampling time* $\tau_{\min} > 0$*, so that undesired Zeno behaviors are avoided.* ◇

Remark 2 *The choice of* $\vartheta_1 \in (0, 1)$ *corresponds to a simple tradeoff between larger intersampling times* $\tau_s(x_k)$ *and robustness with respect to larger delays* $\varDelta_{\max}$*. As* ϑ_1 *decreases,* $\tau_s(x_k)$ *decreases and* $\varDelta_{\max}$ *increases. This implies that we improve robustness vs delays, paid by stronger sampling requirements.* ◇

Remark 3 *When applying the self triggered rule defined in the above theorem in a real scenario, it is necessary to compute on–line the next sampling time for each time instant* t_k*. This computation corresponds to solving a second degree equality, which is reasonable in an embedded system. On the contrary, the functions* $M_1(\cdot)$ *and* $M_2(\cdot)$ *can be determined off–line, and then (numerically) computed on–line in* x_k*. However,* $M_2(\cdot)$ *might be still difficult to determined in closed form. In this case, one can define*

$$M_2 \doteq \max_{\bar{x}, x_k \in \Omega} \|\varphi_2(\bar{x}, x_k)\|$$

and use it in equation (21) to compute the next sampling time. This new definition clearly implies shorter sampling times. ◇

The above remarks also apply to Theorems 2 and 3 in the following Sections.

2.3 Self Triggered Safety Control

The main limitation of the results developed in Section 2.2 is the Lipschitz continuity assumption of $\alpha_3^{-1}(\cdot)$ and $\alpha_4(\cdot)$. The following example shows that even exponentially stabilizable systems do not always satisfy this assumption.

Example 1 *Consider the system $\dot{x} = Ax + Bu + f(x, u) = f_0(x, u)$ with*

$$f_0(x, u) = \begin{pmatrix} -x_1 + x_2 + x_1^2 \\ (1 + x_1)u \end{pmatrix}.$$

Let $u = \kappa(x) = -x_2 \in \mathcal{U}$. Consider the Lyapunov candidate $V(x) = x^T Px$, with P solution of the Lyapunov equation $PA_c + A_c^T P = -Q$, with $Q = 2I$, I the identity matrix, and $A_c = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix}$. Since $P = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$, then $\lambda^P_{\min} \cong 1.382$ and $\lambda^P_{\max} \cong 3.618$ denote respectively the minimum and the maximum eigenvalue of P. For $\|x\| \leq 2/3$, the time derivative of V satisfies

$$\dot{V} = -\|x\|^2_Q + 2|x_1|^3 + 3|x_1|x_2^2 \leq -2x_1^2 - 2x_2^2 + 2(2/3)x_1^2 + 3(2/3)x_2^2 \leq -\frac{1}{2}\|x\|^2$$

thus the origin is locally exponentially stable, with $\alpha_1(\|x\|) = \lambda^P_{\min}\|x\|^2$, $\alpha_2(\|x\|) = \lambda^P_{\max}\|x\|^2$, $\alpha_3(\|x\|) = \|x\|^2/2$, $\alpha_4(\|x\|) = \lambda^P_{\max}\|x\|$.

It is clear that Assumption 1 is not satisfied, since $\alpha_3^{-1}(\cdot)$ is not Lipschitz. For this reason, one the basis of the previous results, one can not ensure the existence of a stabilizing self triggered strategy. ◇

The main technical problem is that, if $\alpha_3^{-1}(\cdot)$ is not Lipschitz, the next sampling time $\tau_s(x_k)$ goes to zero as x_k approaches the equilibrium point, and this might generate Zeno behaviors. The results developed in this section address Problem 2, both for the nominal system (11) and the generic system (10), and are based on the following assumption, that does not require $\alpha_3^{-1}(\cdot)$ to be Lipschitz.

Assumption 2 *Assume that $f_0 \in C^\ell(\mathcal{D}_x \times \mathcal{D}_u)$, with ℓ a positive integer sufficiently large. Assume that there exists a nonempty set $\mathcal{U}$ of state feedback laws $\kappa\colon \mathcal{D}_x \to \mathcal{D}_u$, such that $\kappa \in C^\ell(\mathcal{D}_x)$ and the origin of the system (13) is asymptotically stable, with region of attraction a certain compact $\Omega \subset \mathcal{D}_x$ containing the origin.* ◇

For system (11) (unperturbed case) we determine a function τ_s to compute the next sampling time as a function of the current state of the system and the maximum allowable delay $\Delta_{\max}$, such that the closed loop system applying a self triggered strategy is safe. On the basis of Assumption 2, in Theorem 2 we provide a different computation of τ_s providing less conservative (less frequent) sampling instants.

For system (10) (perturbed case), given a δ boundary of the equilibrium point and a disturbance that is upper bounded in norm by a class $\mathcal{K}$ function $\nu(\delta)$, we determine a function τ_s to compute the next sampling time as a function of the current state of the system and the maximum allowable delay $\Delta_{\max}$, such that the closed loop system applying a self triggered strategy is safe with respect to δ. We remark that, according to well known results in [8], a locally stable system subject to a bounded disturbance always satisfies a safety property with respect to δ sufficiently small. Nevertheless neither the computation of the function τ_s nor the relation among the safe boundary δ and the disturbance upper bound $\nu(\delta)$ are straightforward from the results in [8].

2.4 Unperturbed Systems

The following theorem states that, if a system (11) is asymptotically stabilizable using a continuous time state feedback control law, then it is always possible to keep the state arbitrarily close to the equilibrium point by applying a digital self triggered strategy. Note that, in order to guarantee that the state is *arbitrarily* close to the equilibrium point, we need the stabilizability assumption.

Theorem 2 *Under Assumption 2, Problem 2 is solvable for system (11), and the function τ_s can be iteratively computed as a function of the current state of the system and the maximum allowable delay $\Delta_{\max}$.* ◇

Proof: Using the same reasoning of Theorem 1 proof, and directly considering the case $\Delta_k > 0$, we conclude that the following inequality

$$\dot{V} \leq -(1-\vartheta)\alpha_3(\|x\|) + \alpha_4(\|x\|)(\|d_h\| + \|d_\Delta\|) - \vartheta\alpha_3(\|x\|) \ \leq -(1-\vartheta)\alpha_3(\|x\|)$$

holds when

$$\alpha_4(\|x\|)\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2 + M_3\Delta_k\Big) \leq \vartheta\alpha_3(\|x\|)$$

with $\vartheta \in (0,1)$, and d_h, d_Δ, $M_1(x_k)$, $M_2(x_k)$, M_3 defined as in Theorem 1. The above inequality holds if

$$\|x\| \geq \eta \doteq \alpha_3^{-1}\Big(\frac{\alpha_4(\delta)}{\vartheta}\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2 + M_3\Delta_k\Big)\Big).$$

This implies, by [8], that there exists $b := \alpha_1^{-1}(\alpha_2(\eta)) > 0$ such that $\|x(\tau)\| \leq b$, $\forall \tau \in [t_k, t]$ if $x_k \in \mathcal{B}_b$ and if the following holds

$$\alpha_4(\delta)\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2 + M_3\Delta_k\Big) \leq \vartheta\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big) \tag{25}$$

where we imposed the constraint $b = \delta$. Equation (25) holds if the following inequalities hold

$$\begin{aligned}
&\alpha_4(\delta)\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2\Big) \leq \vartheta_1\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big)\\
&\alpha_4(\delta)M_3\Delta_k \leq \vartheta_2\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big)
\end{aligned} \tag{26}$$

where we have set $\vartheta = \vartheta_1 + \vartheta_2$, with $\vartheta_1, \vartheta_2 \in (0,1)$ and $\vartheta_1 + \vartheta_2 < 1$. Defining

$$\begin{aligned}
\Delta_{\max} &\doteq \vartheta_2 \frac{\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big)}{\alpha_4(\delta)M_3}\\
\tau_s(x_k) &\doteq \max\{t-t_k : \text{ (26) is satisfied for each } t-t_k \in [0, \tau_s(x_k)]\} - \Delta_{\max}\\
\tau_{\min} &\doteq \min_{x_k \in \mathcal{B}_\delta} \tau_s(x_k)
\end{aligned}$$

and if we choose $t_{k+1} = t_k + \tau_s(x_k)$, then (26) holds for all $t \in [t_k + \varDelta_k, t_{k+1} + \varDelta_{\max}]$ and for all $k \geq 0$, with $\varDelta_{\max}$ non-negative. Since $M_1(x_k)$, $M_2(x_k)$ and M_3 are non–negative and upper bounded for each $x_k \in \mathcal{B}_\delta$, and since α_4, $\alpha_3 \circ \alpha_2^{-1} \circ \alpha_1 \in \mathcal{K}$, then the first of (26) is a second degree inequality in the form $ay^2 + by - c \leq 0$, where a, b are non–negative and bounded and c is strictly positive and bounded. Therefore, for ϑ_2 sufficiently small, $t_{k+1} - t_k = \tau_s(x_k) > \varDelta_{\max} \geq 0$ for each $x_k \in \mathcal{B}_\delta$, and thus $\tau_{\min}$ is strictly positive as well. This completes the proof. ■

systems with a Lyapunov function fulfilling Assumption . Hence, condition (**??**) can be used to determine a time instant $\bar{t}_k > t_k$, while the application of Theorem 2 can be used to determine a value δ_k. Hence, the next sampling time results to be $t'_k = \bar{t}_k + \delta_k$, which may be bigger those determined by applying only Theorem 2.

2.5 Perturbed Systems

that there exists a nonempty set $\mathcal{U}$ of state feedback laws $\kappa\colon \mathcal{D}_x \to \mathcal{D}_u$, such that $\kappa \in C^\ell(\mathcal{D}_x)$ and the origin of the system (12) is asymptotically stable. ◇A generic system (10), subject to disturbances and parameter variations, can be seen as the nominal system (11), perturbed by the term

$$d_g \doteq g(x, u, \mu, d) = f(x, u, \mu, d) - f_0(x, u) \doteq d_g. \tag{27}$$

Hence, (10) can be rewritten as follows

$$\dot{x} = f_0(x, u) + g(x, u, \mu, d). \tag{28}$$

Definition 1 *Under Assumption 2, and given the perturbed system (10) and a safe set $\mathcal{B}_\delta$, $\delta > 0$, we say that the perturbation (27) is δ–admissible if there exists a state feedback control law $\kappa \in \mathcal{U}$ and a constant $\vartheta_g \in (0, 1)$ such that the function $g(x, \kappa(x_0), \mu, d)$ satisfies*

$$\max_{\substack{x, x_k \in \mathcal{B}_\delta \\ d \in \mathcal{D}_d \\ \mu \in \mathcal{D}_\mu}} \|g(x, \kappa(x_k), \mu, d)\| \leq \nu(\delta) \doteq \vartheta_g \frac{\alpha_3\big(\alpha_2^{-1}(\alpha_1(\delta))\big)}{\alpha_4(\delta)} \tag{29}$$

with $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ as in (14). ◇

The δ–admissible perturbations are those for which the safety problem with respect to a ball $\mathcal{B}_\delta$ can be solved using continuous time measurement and actuation, namely it is a necessary condition to achieve safety with respect to $\mathcal{B}_\delta$ using sampled measurements and actuations. Note that in condition (1) the expression of $\nu(\delta)$ can be explicitly computed.

The following theorem states that, if a system is asymptotically stabilizable using a continuous time state feedback control law and the perturbation is δ–admissible, then it is possible to keep the state in a boundary $\mathcal{B}_\delta$ of the equilibrium point by applying a digital self triggered strategy.

Theorem 3 *Under Assumption 2, Problem 2 is solvable for system (10) for any δ–admissible perturbation (27), and the function τ_s can be iteratively computed as a function of the current state of the system and the maximum allowable delay $\varDelta_{\max}$.* ◇

Proof: Using the same reasoning as in the proof of Theorem 2, and since the perturbation is assumed δ–admissible, we conclude that the following inequality

$$\dot{V} \leq -(1-\vartheta)\alpha_3(\|x\|) - \vartheta\alpha_3(\|x\|) + \alpha_4(\|x\|)(\|d_h\| + \|d_\varDelta\| + \|d_g\|)$$

$$\leq -(1-\vartheta)\alpha_3(\|x\|)$$

with d_g defined in (27), and d_h, $d_\varDelta$ defined as in Theorem 1, holds when

$$\begin{aligned} &\alpha_4(\delta)\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2\Big) \leq \vartheta_1\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big) \\ &\alpha_4(\delta)M_3\varDelta_k \leq \vartheta_2\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big) \end{aligned} \tag{30}$$

where $\vartheta = \vartheta_1 + \vartheta_2 + \vartheta_g$, with $\vartheta_1, \vartheta_2, \vartheta_g \in (0,1)$, and $\vartheta_1 + \vartheta_2 < 1 - \vartheta_g$, and $M_1(x_k)$, $M_2(x_k)$, M_3 are as in Theorem 1. Defining

$$\begin{aligned} \varDelta_{\max} &\doteq \vartheta_2 \frac{\alpha_3\Big(\alpha_2^{-1}(\alpha_1(\delta))\Big)}{\alpha_4(\delta)M_3} \\ \tau_s(x_k) &\doteq \max\{t - t_k : \text{ (30) is satisfied for each } t - t_k \in [0, \tau_s(x_k)]\} - \varDelta_{\max} \\ \tau_{\min} &\doteq \min_{x_k \in \mathcal{B}_\delta} \tau_s(x_k) \end{aligned}$$

and if we choose $t_{k+1} = t_k + \tau_s(x_k)$, then (30) holds for all $t \in [t_k + \varDelta_k, t_{k+1} + \varDelta_{\max}]$ and for all $k \geq 0$, with $\varDelta_{\max}$ non-negative. Arguing as in the proof of Theorem 2, for ϑ_2 sufficiently small, $t_{k+1} - t_k = \tau_s(x_k) > \varDelta_{\max} \geq 0$ for each $x_k \in \mathcal{B}_\delta$, and thus $\tau_{\min}$ is strictly positive as well. This completes the proof. ■

As discussed in Section 2.2, the choice of ϑ_1, ϑ_2 and ϑ_g corresponds to a simple tradeoff between larger intersampling times (ϑ_1), and robustness with respect to larger delays (ϑ_2) and perturbations (ϑ_g).

Remark 4 *Theorems 2 and 3 prove the existence of a self triggered strategy characterized by the time sequence $\mathcal{I} = \{t_k\}_{k\geq 0}$, with $t_k \geq \tau_{\min} > 0$ for each $k \geq 0$, such that the closed loop system satisfies a given safety specification. Moreover, they provide a formula to explicitly compute the next sampling time t_{k+1} as a function of the state x_k at time t_k.*

Although the simulation results, illustrated in Section 2.6, show strong benefits of the proposed self triggered strategy with respect to controllers based on constant sampling, the sequence $\mathcal{I}$ might be conservative, in the sense that longer sampling times might be determined, because of the approximations used in the proof. A trivial way to obtain a less conservative sequence $\mathcal{I}$ without introducing more restricting assumptions is the use of Taylor expansions of order higher than 2.

2.6 An Example of Application of the Digital Self Triggered Robust Control

Consider the system defined in Example 1. As already shown, we can not imply the existence of a stabilizing self triggered strategy. However, since Assumption 2 holds, Theorem 2 implies the existence of a self triggered strategy that guarantees safety for an arbitrary small neighborhood of the equilibrium point. In particular, since the origin of the system is locally exponentially stabilizable for $\|x\| \leq 2/3$, we define the safe set $\mathcal{B}_\delta$ with $\delta = 10^{-4} < 2/3$. We performed simulations using Matlab, with initial condition $x_0 = (10^{-5}, 10^{-5})^T \in \mathcal{B}_\delta$.

When a discrete time control law with constant sampling time greater than 2.1 s is used, the closed loop system is unstable.

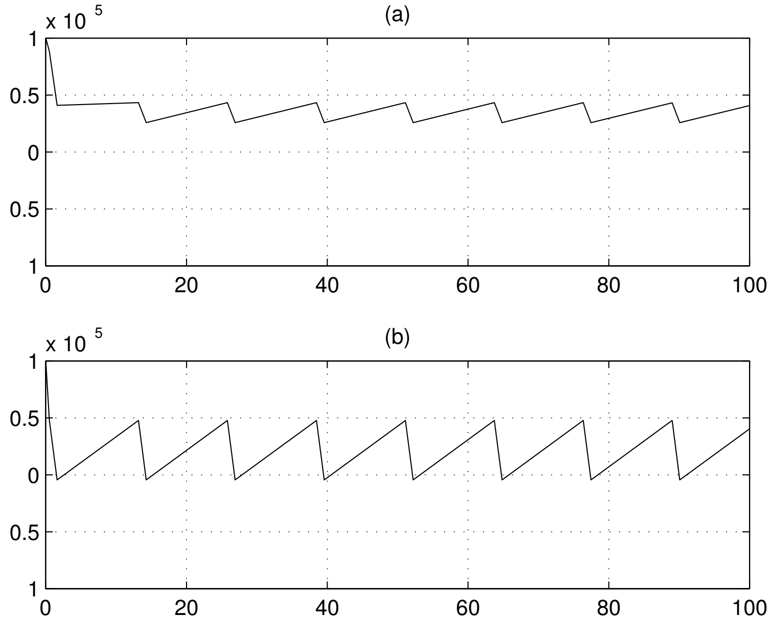


Figure 1: Self triggered control with $\vartheta_1 = 0.99$ and $\vartheta_2 = 0.009$: (a) x_1; (b) x_2 vs time

In Figure 1, the closed loop behavior is illustrated when the proposed self triggered control algorithm is used, with $\vartheta_1 = 0.99$ and $\vartheta_2 = 0.009$. The closed loop system is not asymptotically stable, but is safe with respect to $\mathcal{B}_\delta$ for the time interval $[t_0, \infty)$. It is interesting to remark that the average sampling time is

6.2 s, i.e. more than 295% longer than the constant sampling time of 2.1 s that yields an unstable control loop. Thus, using the proposed self triggered control algorithm, we achieve safety reducing of more than 295% the battery energy consumption, with respect to an unstable control strategy with constant sampling. However, since we have chosen $\vartheta_2 = 0.009$, we can only guarantee robustness with respect to delays bounded by $\varDelta_{\max} = 0.17$ ms.

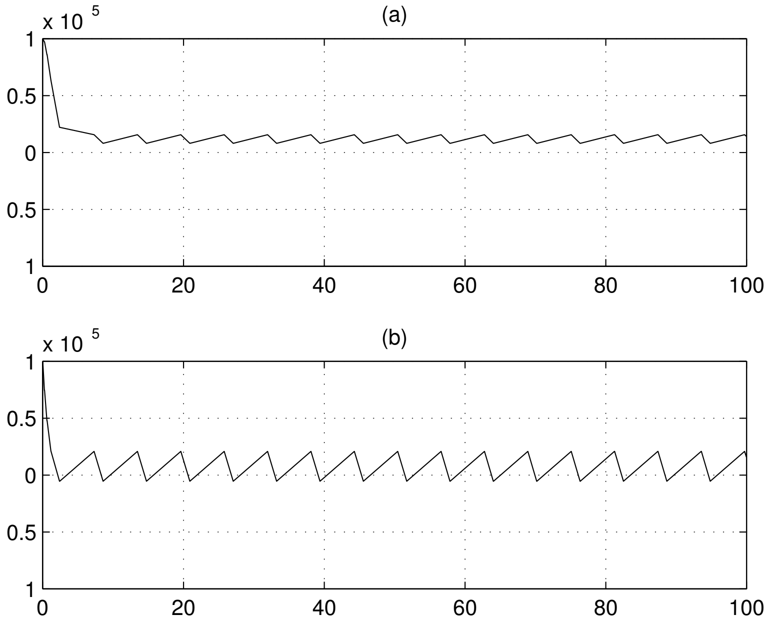


Figure 2: Self triggered control with $\vartheta_1 = 0.5$ and $\vartheta_2 = 0.499$: (a) x_1; (b) x_2 vs time

In Figure 2, the closed loop behavior is illustrated when the proposed self triggered control algorithm law is used, with $\vartheta_1 = 0.5$ and $\vartheta_2 = 0.499$. The closed loop system is not asymptotically stable, but is still safe with respect to $\mathcal{B}_\delta$ for the time interval $[t_0, \infty)$. However, since we have chosen $\vartheta_1 = 0.5$ in order to be robust with respect to delays, the average sampling time 3 s is more conservative with respect to the case $\vartheta_1 = 0.99$. Nevertheless, the average sampling time is almost 50% longer than the constant sampling time of 2.1 s, that yields an unstable control loop. Since we have chosen $\vartheta_2 = 0.009$, we can guarantee robustness with respect to delays bounded by $\varDelta_{\max} = 9$ ms. Thus, using the proposed self triggered control algorithm, we achieve safety reducing of almost 50% the battery energy consumption, with respect to an unstable control strategy with constant sampling, while guaranteeing robustness with respect to delays bounded by $\varDelta_{\max} = 9$ ms.

3 Self Triggered Robust Control of Nonlinear Stochastic Systems

In this section we consider the self–triggered stabilization problem for the class of stochastic systems, where the input generically enters both in the deterministic dynamics and in those affected by noise. This class of system is of particular interest since in practice various disturbances, not measurable, may affect the dynamics. We assume that the state equations are described by an Itô differential equation driven by a Wiener noise [24], [27], [28].

3.1 Problem Formulation

We consider nonlinear stochastic systems of the form

$$dx(t) = f_0(x, u)dt + \sum_{j=1}^{m} g_{0j}(x, u)d\xi_j(t) \tag{31}$$

where $x \in \mathcal{D}_x \subset r^n$, $\mathcal{D}_x$ is a domain containing the origin, $u \in \mathcal{D}_u \subset r^p$, f_0, $g_{0j} : \mathcal{D}_x \times \mathcal{D} \sqcap \to r^n$, $j = 1, \cdots, m$ are sufficiently smooth vector fields, such that $f_0(0, 0) = 0$, $g_{0j}(0, 0) = 0$, $j = 1, \cdots, m$. Moreover, $\{\xi(t) = (\xi_1(t), \cdots, \xi_m(t))^T, t \geq 0\}$ is a standard R^m–valued Wiener process, defined on the usual complete probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, P)$, with $(\mathcal{F}_t)_{t\geq 0}$ the complete right–continuous filtration generated by ξ and $\mathcal{F}_0$ contains all P–null sets. It is worth stressing that in (31) the control appears either in the deterministic or in the stochastic terms [16], [17].

Given a continuous state feedback control law $\kappa \colon \mathcal{D}_x \to \mathcal{D}_u$, the closed loop system is

$$dx(t) = f_0(x, \kappa(x))\, dt + \sum_{j=1}^{m} g_{0j}(x, \kappa(x))\, d\xi_j(t) \tag{32}$$

and we will denote by $x(t)$, $t \geq t_0$, the solution of the closed loop system (32), with initial condition $x_0 = x(t_0)$. It is well–known that if the origin of system (32) is locally asymptotical stable in probability for a certain feedback κ, and if $f_0(x, \kappa(x)) \in C^\ell(\mathcal{D}_x)$, $\ell > 1$ integer, then there exists a Lyapunov function

$V(x)$ of class $C^2(\mathcal{D}_x)$ such that [25], [26]

$$\begin{aligned} \alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \\ \mathcal{L}V(x) \leq -\alpha_3(\|x\|) \\ \left\|\frac{\partial V(x)}{\partial x}\right\| \leq \alpha_4(\|x\|) \\ \left\|\frac{\partial^2 V(x)}{\partial x^2}\right\| \leq \alpha_5(\|x\|) \end{aligned} \tag{33}$$

with $\alpha_i \in \mathcal{K}$, $i = 1, \cdots, 5$. The infinitesimal generator associated to (32), obtain by differentiating V in the sense of Itô, is given by

$$\mathcal{L}V(x) = \frac{\partial V(x)}{\partial x} f_0(x, \kappa(x)) + \frac{1}{2}\sum_{j=1}^{m} \mathrm{Tr}\Big(g_{0j}^T(x, \kappa(x)) \frac{\partial^2 V}{\partial x^2} g_{0j}(x, \kappa(x))\Big). \tag{34}$$

Here, the matrix $\partial^2 V(t, x)/\partial x^2$ is the Hessian matrix of the second order partial derivatives, and $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix.

Clearly, the feedback control signal $u(t) = \kappa(x(t))$ requires continuous measurements of the state of the system. In view of an implementation of $\kappa(x(t))$ by means of digital devices, with variable sampling intervals δ_k, in the following we consider its digital version

$$u(t) = \kappa(x_k), \qquad \forall t \in [t_k, t_{k+1} = t_k + \delta_k),\ k \geq 0 \tag{35}$$

one needs to determine these sampling instants t_k so that the stability property of the origin is preserved in probability. Following the approach developed in [14], [11], [4], [13], [5], [6], the aim is hence to determine on–line a sequence of strictly positive sampling intervals $\delta_k > 0$, i.e. a sequence $\{t_k\}_{k\geq 0}$ of sampling times, such that the origin of

$$dx(t) = f_0(x, \kappa(x_k))dt + \sum_{j=1}^{m} g_{0j}(x, \kappa(x_k))d\xi_j(t) \tag{36}$$

is asymptotically stable in probability.

It is worth noting that to require $\delta_k > 0$ means that there exists a minimum sampling time $0 < \delta_k \leq \tau_{\min}$, $\forall\ k \geq 0$, which in turns will ensure that no Zeno behavior can occur. Hence, the time interval between two sampling instants is lower bounded by $\tau_{\min} > 0$.

The philosophy behind the self–triggered control is obvious: the control is performed only when necessary for guaranteeing the control objectives. This clearly reduces the transmission power of the sensing and actuation data transmissions, as well as the control effort of the digital device computing the control.

3.2 Self Triggered Stabilizing Control

The result developed in this section is based on the following assumption, analogous to the assumptions used in [14], [6] in the case of a deterministic systems.

Assumption 3 *Assume that*

1. $f_0, g_0 \in C^\ell(\mathcal{D}_x \times \mathcal{D}_u)$, *with* ℓ *a positive integer sufficiently large;*

2. *There exists a nonempty set* $\mathcal{U}$ *of state feedback laws* $\kappa \colon \mathcal{D}_x \to \mathcal{D}_u$, *such that* $\kappa \in C^\ell(\mathcal{D}_x)$ *and the origin of (32) is asymptotically stable in probability, with region of attraction a certain compact* $\Omega \subset \mathcal{D}_x$;

3. *The functions* $\alpha_3, \alpha_4, \alpha_5 \in \mathcal{K}$ *in (33) are such that* $\alpha_3^{-1}, \alpha_4, \alpha_5$ *are Lipschitz.* ◇

The assumption of sufficient regularity of the functions f_0, g_0 is required in order to ensure the determination of the next sampling time, making use of a Taylor expansion, analogous to that used in [6]. The assumption of existence of a stabilizing control is not restrictive, since if the nominal system cannot be stabilized in probability using continuous time measurements and actuations, then it is clear that the nominal system cannot be stabilized using a digital control with zero–order holders. Finally, the Lipschitz assumption on $\alpha_3^{-1}, \alpha_4, \alpha_5$ is required to write a simple stability condition, as used in [14], and represents the main limitation of this approach.

Using Assumption 3, one can state the following result.

Theorem 4 *Let us consider the nonlinear stochastic system (31). Under Assumption 3, there exist a piece–wise constant state feedback control law (35), and a sequence of strictly positive sampling intervals* $\delta_k > 0$, *such that the origin of the closed loop system (36) is asymptotically stable in probability.* ◇

Proof: Since $\mathcal{U}$ is not empty, by Assumption 3, we pick a state feedback control law $\kappa \in \mathcal{U}$. Since $f_0(x, \kappa(x)) \in C^\ell(\mathcal{D}_x)$ with $\ell > 1$, there exists a Lyapunov candidate (33). Let us choose $r > 0$ such that the ball $B_r = \{x \in \Omega \mid \|x\| \leq r\} \subset \Omega$.

For $x_k \in B_r$, the infinitesimal generator associated to (36) is given by

$$\mathcal{L}V(x_k) = \frac{\partial V}{\partial x} f_0(x, \kappa(x_k)) + \frac{1}{2} \sum_{j=1}^{m} \mathrm{Tr}\Big(g_{0j}^T(x, \kappa(x_k)) \frac{\partial^2 V}{\partial x^2} g_{0j}(x, \kappa(x_k)) \Big)$$

which can be rewritten as

$$\begin{aligned}\mathcal{L}V(x_k) = &\frac{\partial V}{\partial x} f_0(x,\kappa(x)) + \frac{\partial V}{\partial x}\Big(f_0(x,\kappa(x_k)) - f_0(x,\kappa(x))\Big) + \frac{1}{2}\sum_{j=1}^{m} \mathrm{Tr}\Big(g_{0j}^T(x,\kappa(x))\,\frac{\partial^2 V}{\partial x^2}\,g_{0j}(x,\kappa(x))\Big)\\ &+ \frac{1}{2}\sum_{j=1}^{m}\mathrm{Tr}\Big(\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T \times \frac{\partial^2 V}{\partial x^2}\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]\Big) \qquad (37)\\ &+ \sum_{j=1}^{m}\mathrm{Tr}\Big(g_{0j}^T(x,\kappa(x_k))\,\frac{\partial^2 V}{\partial x^2}\,g_{0j}(x,\kappa(x))\Big) - \sum_{j=1}^{m}\mathrm{Tr}\Big(g_{0j}^T(x,\kappa(x))\,\frac{\partial^2 V}{\partial x^2}\,g_{0j}(x,\kappa(x))\Big).\end{aligned}$$

Note that

$$\begin{aligned}&\frac{1}{2}\mathrm{Tr}\Big(\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T \times \frac{\partial^2 V}{\partial x^2}\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]\Big)\\ &\quad\le \frac{n}{2}\Big\|\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T \times \frac{\partial^2 V}{\partial x^2}\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]\Big\|_\infty\\ &\quad\le \frac{n\sqrt{n}}{2}\Big\|\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T \times \frac{\partial^2 V}{\partial x^2}\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]\Big\|\\ &\quad\le \frac{n\sqrt{n}}{2}\Big\|g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\Big\|^2\Big\|\frac{\partial^2 V}{\partial x^2}\Big\|\end{aligned}$$

for all $j = 1,\cdots,m$. Similarly, the sum of the two last terms of (37) are such that

$$\begin{aligned}&\mathrm{Tr}\left(\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T\frac{\partial^2 V}{\partial x^2}g_{0j}(x,\kappa(x))\big]\right)\\ &\qquad\le n\Big\|\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T\frac{\partial^2 V}{\partial x^2}g_{0j}(x,\kappa(x))\Big\|_\infty\\ &\qquad\le n\sqrt{n}\Big\|\big[g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\big]^T \times \frac{\partial^2 V}{\partial x^2}g_{0j}(x,\kappa(x))\Big\|\\ &\qquad\le n\sqrt{n}\Big\|g_{0j}(x,\kappa(x_k)) - g_{0j}(x,\kappa(x))\Big\| \times \Big\|g_{0j}(x,\kappa(x))\Big\|\Big\|\frac{\partial^2 V}{\partial x^2}\Big\|.\end{aligned}$$

Using these bounds, one obtains

$$\begin{aligned}\mathcal{L}V(x_k) \le -\alpha_3(\|x\|) + \alpha_4(\|x\|)\|d_{h,f}\| + \frac{n\sqrt{n}}{2}\alpha_5(\|x\|)\sum_{j=1}^{m}\|d_{h,g_j}\|^2\\ + n\sqrt{n}\alpha_5(\|x\|)\sum_{j=1}^{m}\|d_{h,g_j}\|\|g_{0j}(x,\kappa(x))\|\end{aligned} \qquad (38)$$

where

$$d_{h,f} = f_0(x(t),\kappa(x(t_k))) - f_0(x(t),\kappa(x(t)))$$

$$d_{h,g_j} = g_{0j}(x(t),\kappa(x(t_k))) - g_{0j}(x(t),\kappa(x(t)))$$

$j = 1,\cdots,m$, are terms that can be regarded as perturbations, due to the holding, and acting on the control system (32).

Under Assumption 3, there exists a time interval $[t_k, t_k+\varepsilon_k]$ such that (36) has a unique solution $x(t)$. Hence, it is possible to expand in Taylor series the i^{th} components $d_{h,f,i}, d_{h,g_j,i}$ of $d_{h,f}, d_{h,g_j}$, $j = 1, \cdots, m$, with respect to $t \in [t_k, t_k+\varepsilon_k]$, on the right of t_k, with the Lagrange remainder. Denoting by $\mathrm{d}_+^n(\cdot)/\mathrm{d}t^n$ the n–th right derivative, and proceeding as in [6], one works out

$$\begin{aligned} d_{h,f,i} &= \varphi_{1,i}(x_k)(t-t_k) + \varphi_{2,i}(\bar{x}_i, x_k)(t-t_k)^2 \\ d_{h,g_j,i} &= \varphi_{3j,i}(x_k)(t-t_k) + \varphi_{4j,i}(\bar{x}_i, x_k)(t-t_k)^2 \end{aligned} \tag{39}$$

for $j = 1, \cdots, m$, where we have defined

$$\begin{aligned} \varphi_{1,i}(x_k) &= d_{h,f,i}\Big|_{x(t)=x_k} \\ \varphi_{2,i}(\bar{x}_i, x_k) &= \frac{1}{2}\left.\frac{\mathrm{d}_+^2 d_{h,f,i}}{\mathrm{d}t^2}\right|_{x(t)=\bar{x}_i} \\ \varphi_{3j,i}(x_k) &= d_{h,g_j,i}\Big|_{x(t)=x_k} \\ \varphi_{4j,i}(\bar{x}_i, x_k) &= \frac{1}{2}\left.\frac{\mathrm{d}_+^2 d_{h,g_j,i}}{\mathrm{d}t^2}\right|_{x(t)=\bar{x}_i} . \end{aligned}$$

The Taylor theorem with the Lagrange remainder ensures the existence of time instants $\bar{t}_i \in [t_k, t]$, with $\bar{x}_i = x(\bar{t}_i)$, $i = 1, \cdots, n$, such that the equalities (39) hold. Denoting by $\bar{x} = (\bar{x}_1, \cdots, \bar{x}_n)$ the corresponding point, one obtains

$$\begin{aligned} \|d_{h,f}\| &\leq \|\varphi_1(x_k)\|(t-t_k) + \|\varphi_2(\bar{x}, x_k)\|(t-t_k)^2 \\ \|d_{h,g_j}\| &\leq \|\varphi_{3j}(x_k)\|(t-t_k) + \|\varphi_{4j}(\bar{x}, x_k)\|(t-t_k)^2 \end{aligned} \tag{40}$$

where

$$\begin{aligned} \varphi_p(x_k) &= \Big(\varphi_{p,1}(x_k), \cdots, \varphi_{p,n}(x_k)\Big)^T \\ \varphi_q(\bar{x}, x_k) &= \Big(\varphi_{q,1}(\bar{x}_1, x_k), \cdots, \varphi_{q,n}(\bar{x}_n, x_k)\Big)^T \end{aligned}$$

for $p = 1, 3j$ and $q = 2, 4j$, $j = 1, \cdots, m$. Moreover, let us consider the level set $\Omega_{V(x_k)}$, and define

$$\begin{aligned} M_p(x_k) &= \|\varphi_p(x_k)\|, & p &= 1, 3j \\ M_q(x_k) &= \max_{\bar{x}\in\Omega_{V(x_k)}} \|\varphi_q(\bar{x}, x_k)\|, & q &= 2, 4j. \end{aligned}$$

Since $f_0, g_{0j}, \kappa \in C^\ell$ and $\Omega_{V(x_k)}$ is compact, then $M_p(x_k)$ is finite for any $x_k \in \Omega_{V(x_k)}$, and $M_q(x_k) \in r^+$ exists and is finite for any $x_k \in \Omega_{V(x_k)}$.

Finally, one can introduce the terms

$$C_{g_{0j}} = \max_{x\in\Omega_{V(x_k)}} \left\|g_{0j}(x, \kappa(x))\right\|, \quad j = 1, \cdots, m$$

which exist and are finite on the compact set $\Omega_{V(x_k)}$, so that the infinitesimal generator (38) associated to (36) can be written as follows

$$\begin{aligned}\mathcal{L}V(x_k) &\leq -\alpha_3(\|x\|) + \alpha_4(\|x\|)\|d_{h,f}\| + \frac{n\sqrt{n}}{2}\alpha_5(\|x\|)\sum_{j=1}^{m}(\|d_{h,g_j}\| + 2C_{g_{0j}}\|d_{h,g_j}\|)\\ &= -\alpha_3(\|x\|) + \alpha_4(\|x\|)\|d_{h,f}\| + \frac{n\sqrt{n}}{2}\alpha_5(\|x\|)\sum_{j=1}^{m}\Big((\|d_{h,g_j}\| + C_{g_{0j}})^2 - C_{g_{0j}}^2\Big).\end{aligned} \tag{41}$$

In what follows we will show that, for each term in (41) which is not negative definite, one can consider a negative definite term that ensures the negativity of the whole $\mathcal{L}V$, at least for small (but nonzero) time intervals. For, let us consider $\vartheta = \sum_{j=0}^{m}\vartheta_j < 1$, with $\vartheta_j \in (0,1)$, $j = 0,\cdots,m$, and let us require that

$$\begin{aligned}\alpha_4(\|x\|)\|d_{h,f}\| &\leq \vartheta_0\alpha_3(\|x\|)\\ \frac{n\sqrt{n}}{2}\alpha_5(\|x\|)\Big((\|d_{h,g_j}\| + C_{g_{0j}})^2 - C_{g_{0j}}^2\Big) &\leq \vartheta_j\alpha_3(\|x\|)\\ j &= 1,\cdots,m\end{aligned} \tag{42}$$

are satisfied. Conditions (42) will determine a time instant $t_{k+1} = t_k + \delta_k < t_k + \varepsilon_k$, and hence a positive time interval δ_k in which the infinitesimal generator is negative definite. In fact, using (40), equations (42) are satisfied if

$$\begin{aligned}\alpha_3^{-1}\Big(\frac{1}{\vartheta_0}\alpha_4(\|x\|)\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2\Big)\Big) &\leq \|x\|\\ \alpha_3^{-1}\Big(\frac{n\sqrt{n}}{2\vartheta_j}\alpha_5(\|x\|)\Big[(M_{3j}(x_k)(t-t_k) + M_{4j}(x_k)(t-t_k)^2 + C_{g_{0j}})^2 - C_{g_{0j}}^2\Big]\Big) &\leq \|x\|\end{aligned}$$

for $j = 1,\cdots,m$. Since α_3^{-1}, α_4 and α_5 are Lipschitz, then equations (42) are satisfied if

$$\begin{aligned}\frac{1}{\vartheta_0}L_{\alpha_3^{-1}}L_{\alpha_4}\|x\|\Big(M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2\Big) &\leq \|x\|\\ \frac{n\sqrt{n}}{2\vartheta_j}L_{\alpha_3^{-1}}L_{\alpha_5}\|x\|\Big[\Big(M_{3j}(x_k)(t-t_k) + M_{4j}(x_k)(t-t_k)^2 + C_{g_{0j}}\Big)^2 - C_{g_{0j}}^2\Big] &\leq \|x\|\end{aligned}$$

for all $j = 1,\cdots,m$, where $L_{\alpha_3^{-1}}, L_{\alpha_4}, L_{\alpha_5} > 0$ are the Lipschitz constants of $\alpha_3^{-1}, \alpha_4, \alpha_5$, respectively. These equations imply that (42) are satisfied under the sufficient conditions

$$\begin{aligned}M_1(x_k)(t-t_k) + M_2(x_k)(t-t_k)^2 &\leq \frac{\vartheta_0}{L_{\alpha_3^{-1}}L_{\alpha_4}}\\ M_{3j}(x_k)(t-t_k) + M_{4j}(x_k)(t-t_k)^2 &\leq \left(\sqrt{1 + \frac{2\vartheta_j}{n\sqrt{n}}\frac{1}{C_{g_{0j}}^2 L_{\alpha_3^{-1}}L_{\alpha_5}}} - 1\right)C_{g_{0j}}\end{aligned} \tag{43}$$

for $j = 1,\cdots,m$. Defining

$$\delta_k = \min\max\Big\{t - t_k \mid \text{ (43) are satisfied, } j = 1,\cdots,m\Big\}$$

and choosing $t_{k+1} = t_k + \delta_k$, then

$$\mathcal{L}V(x) \leq -(1 - \vartheta)\alpha_3(\|x\|)$$

for all $t \in [t_k, t_{k+1}]$ and for all $k \geq 0$. This implies that the origin is asymptotically stable in probability.

Equations (43) are second degree inequalities in the form $a(x_k)y^2 + b(x_k)y \leq c$, where $a(x_k), b(x_k)$ are non-negative and upper bounded for each $x_k \in \mathcal{D}_x$, and c is strictly positive and upper bounded. This trivially implies that δ_k, $\forall\ k \geq 0$, are strictly positive for each $x_k \in \Omega_{V(x_k)}$, and thus a minimum dwell time does exists, so ensuring that δ_k does not go to zero as $k \to \infty$. ■

Remark 5 *From the proof of the previous result, it is clear that the main difference between the deterministic and the stochastic case consists of the fact that the sampling period has to satisfy extra conditions [6]. In fact, while in the deterministic case one can determine a sampling sequence $\{\delta_k\}$ solving only the first of conditions (43), in the stochastic case one needs to satisfy m additional conditions given by the second of (43). Therefore, in the stochastic case the self–triggered control strategy will determine, in general, more restrictive (shorter) sampling times.* ◇

3.3 Self Triggered Safety Control

The main limitation of the results developed in Section 3.2 is the Lipschitz continuity assumption of $\alpha_3^{-1}(\cdot)$. In fact, if $\alpha_3^{-1}(\cdot)$ is not Lipschitz, the next sampling time $t_k + \delta_k$ goes to zero as x_k approaches the equilibrium point, and this might generate Zeno behaviors. Hence, in the spirit of the self triggered safety control addressed in [6], in this section we will show that it is possible to keep the state arbitrarily close to the equilibrium point by applying a self triggering strategy. The solution of this problem will not require the Lipschitz assumption on α_3^{-1}.

In the following definition, an invariant property is used to define that a system is almost surely (a.s.) safe with respect to a given subset of the state space.

Definition 2 *Given a state feedback control law κ, system (32) is a.s. safe with respect to the set $\mathcal{S} \subseteq \mathcal{D}_x$ for the time interval $\mathcal{T} \subseteq r^+$, if $x(t) \in \mathcal{S}$, $\forall t \in \mathcal{T}$ a.s.* ◇

Given the system (31), a stabilizing state feedback control law κ, and an arbitrary safe set $\mathcal{B}_\delta = \{x \in r^n \mid \|x\| < \delta\} \subset \mathcal{D}_x$, the objective is to determine a sequence of strictly positive sampling intervals $\delta_k > 0$ and a piece–wise constant state feedback control law, as in the previous section, such that the closed loop system (32) is a. s. safe with respect to $\mathcal{B}_\delta$, for the time interval $[t_0, \infty)$. ◇

The results developed in this section are based on the following.

Assumption 4 *Assume that $f_0, g_{0j} \in C^\ell(\mathcal{D}_x \times \mathcal{D}_u)$, $j = 1, \cdots, m$, with ℓ a positive integer sufficiently large. Assume that there exists a nonempty set $\mathcal{U}$ of state feedback laws $\kappa: \mathcal{D}_x \to \mathcal{D}_u$, such that $\kappa \in C^\ell(\mathcal{D}_x)$ and the origin of the system (32) is asymptotically stable in probability.* ⋄

The following theorem states that if a system is almost surely asymptotically stabilizable, using a continuous time state feedback control law, then it is always possible to keep the state arbitrarily close to the equilibrium point by applying a digital self triggering strategy. Note that, in order to guarantee that the state is arbitrarily close to the equilibrium point, we still need the stabilizability assumption.

Theorem 5 *Given the system (31) and a safe set $\mathcal{B}_\delta$, $\delta > 0$, under Assumption 4 there exist piece–wise constant state feedback control law (35) and a sequence of strictly positive sampling intervals $\delta_k > 0$ such that the closed loop system (32), (35) is almost surely safe with respect to $\mathcal{B}_\delta$, for the time interval* $[t_0, \infty)$. ⋄

Proof: The proof of Theorem (5) follows the same arguments of Theorem (4).

3.4 A Simple Illustrative Example

In order to illustrate the propose approach, let us consider the following system

$$dx = (Ax + Bu + f(x,u))dt + Cxdw$$

$$= f_0(x,u)dt + g_0(x,u)dw$$

with

$$f_0(x,u) = \begin{pmatrix} -x_1 + x_2 + x_1^2 \\ (1 + x_1)u \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Let us consider the continuous control $u = \kappa(x) = -x_2 \in \mathcal{U}$, and the Lyapunov candidate $V(x) = x^T P x$, with P solution of the Lyapunov equation

$$PA_c + A_c^T P + C^T PC = -R$$

with R a symmetric positive definite matrix, and

$$A_c = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix}.$$

If

$$R = \begin{pmatrix} -2 & 0 \\ 0 & -3 \end{pmatrix}$$

the matrix

$$P = \begin{pmatrix} 3 & 1 \\ 1 & 4 \end{pmatrix}$$

is a solution of the Lyapunov equation, with $\lambda^P_{\min} \cong 2.382$, $\lambda^P_{\max} \cong 4.618$ the minimum and the maximum eigenvalue of P, respectively.

The infinitesimal generator associated to the previous system, for $\|x\| \leq 1/3$ satisfies

$$\begin{aligned} \mathcal{L}V &\leq -2x_1^2 - 3x_2^2 + 6|x_1|^3 + 8|x_1|x_2^2 \\ &\leq -2x_1^2 - 3x_2^2 + 6(1/3)x_1^2 + 8(1/3)x_2^2 \leq -\frac{1}{3}\|x\|^2. \end{aligned}$$

Thus, the origin of the system is almost surely exponentially stable in probability, with

$$\begin{aligned} &\alpha_1 = \lambda^P_{\min}\|x\|^2, \qquad \alpha_2 = \lambda^P_{\max}\|x\|^2, \quad \alpha_3 = \|x\|^2/3 \\ &\alpha_4 = \lambda^P_{\max}\|x\|, \qquad \alpha_5 = \|2P\|. \end{aligned}$$

It is clear that Assumption 3 is not satisfied, since α_3^{-1} is not Lipschitz. For this reason, we can not imply the existence of a stabilizing self triggered strategy. However, Theorem 5 implies the existence of a self triggered strategy that guarantees almost surely safety for an arbitrary small neighborhood of the equilibrium point. Since the origin is locally stabilizable in probability for $\|x\| \leq 1/3$, we can define the safe set as the ball $\mathcal{B}_\delta$ with $\delta = 10^{-4} < 1/3$.

4 Implementation Features about Control Algorithms in Digital Logic Devices

In the digital logic system scenario, there are many architectures suitable for realizing different kinds of control algorithms considering performances in terms of timing, power consumption and resources availability.

In this section a brief introduction about microprocessors and custom devices for digital processing is presented, in order to obtain an evaluation about benefits and drawbacks. DSP processors are described in their features and logic functioning, focusing on how they tend to be integrated in wider systems including FPGAs, memories, peripherals, etc. Moreover, FPGA technology is depicted in the control environment explaining how designers are trying to exploit their potentialities in applications of parallel computation, control and digital signal processing.

4.1 General Description of DSP Systems

The informal definition of digital signal processing is the application of mathematical operations to digitally represent and elaborate signals. Often, samples are obtained from physical signals (for example, audio signals) through the use of transducers (such as microphones) and analog-to-digital converters. After mathematical processing, digital signals may be converted back to physical signals via digital-to-analog converters. In some systems, the use of DSP is crucial for the operation of the system. For example, modems and digital cellular telephones rely very heavily on DSP technology. In other products, the use of DSP is less central, but often offers important competitive advantages in terms of features, performance, and costs. For example, manufacturers of primarily analog consumer electronics devices like audio amplifiers are beginning to employ DSP technology to provide new features.

4.1.1 DSP Advantages

Digital signal processing enjoys several advantages over analog signal processing. The most significant of these is that DSP systems are able to accomplish tasks inexpensively that would be difficult or even impossible using analog electronics. Examples of such applications include speech synthesis, speech recognition, and high-speed control techniques. DSP systems also enjoy two additional advantages over analog systems:

1. Low sensitivity to the environment. Digital systems, by their nature, are considerably less sensitive to environmental conditions than analog systems. For example, an analog circuit's behavior depends strictly on its temperature. In contrast, barring catastrophic failures, a DSP system's delivers the same response for little temperature changes.
2. Insensitivity to component tolerances. Analog components are manufactured to particular tolerances-a resistor, i.e., might be guaranteed to have a resistance within 1 percent of its nominal value. The overall response of an analog system depends on the actual values of all of the analog components used. As a result, two analog systems of exactly the same design will have slightly different responses due to slight variations in their components. In contrast, correctly functioning digital components always produce the same outputs given the same inputs.

These two advantages combine synergistically to give DSP systems an additional benefit over analog systems:

3. Predictable, repeatable behavior. Because a DSP system's output does not vary due to environmental factors or component variations, it is possible to design systems having exact, known responses.

Finally, some DSP systems may also have two other advantages over analog systems:

4. Reprogrammability. If a DSP system is based on programmable processors or programmable logic devices (PLD) in general, it can be reprogrammed, even in the field, to perform other tasks. In contrast, analog systems require physically different components to perform different tasks.
5. Size. The size of analog components varies related to their values.

These advantages, coupled with the fact that DSP can take advantage of the rapidly increasing density of digital integrated circuit manufacturing processes, increasingly make DSP the solution of choice for signal processing.

4.1.2 DSP Systems Features

In this section a number of characteristics common to all DSP systems are described, such as algorithms, sample rate, clock rate, and arithmetic types.

1. Algorithms

   DSP systems are often characterized by algorithms. The algorithm specifies the arithmetic operations to be performed but does not specify how that arithmetic is to be implemented. It might be implemented in software on an ordinary microprocessor or in a programmable signal processor, or it might be implemented in custom integrated circuits. The selection of an implementation technology is determined in part by the required speed and arithmetic precision.

2. Sample Rates

   A key characteristic of a DSP system is its sample rate: the rate at which samples are consumed, processed, or produced. Combined with the complexity of the algorithms, the sample rate determines the required speed of the implementation technology. A familiar example is the digital audio compact disc (CD) player, which produces samples at a rate of 44.1 kHz on two channels. Of course, a DSP system may use more than one sample rate; such systems are said to be multirate DSP systems. An example is a converter from the CD sample rate of 44.1 kHz to the digital audio tape (DAT) rate of 48 kHz. Because of the awkward ratio between these sample rates, the conversion is usually done in stages, typically with at least two intermediate sample rates. Another example of a multirate algorithm is a filter bank, used in applications such as speech, audio, and video encoding and some signal analysis algorithms. Filter banks typically consist of stages that divide the signal into high and low frequency portions. These new signals are then downsampled (i.e., their sample rate is lowered by periodically discarding samples) and divided again. In multirate applications, the ratio between the highest and the lowest sample rates in the system can become quite large, sometimes exceeding 100,000. The range of sample rates encountered in signal processing systems is huge. Sample rates for applications range over 12 orders of magnitude. Only at the very top of that range is digital implementation rare. This is because the cost and difficulty of implementing a given algorithm digitally increases with the sample rate. DSP algorithms used at higher sample rates tend obviously to be simpler than those used at lower sample rates. Many DSP systems must meet extremely rigorous speed goals, since they operate on lengthy segments of real world signals in real–time. Where other kinds of systems (like databases) may be required to meet performance goals on average, real–time DSP systems often must meet such goals in every instance. In such systems, failure to maintain the necessary processing rates is considered a serious malfunction. Such systems are often said to be subject to hard realtime constraints.

3. Clock Rates

Digital electronic systems are often characterized by their clock rates. The clock rate usually refers to the rate at which the system performs its most basic unit of work. In mass-produced, commercial products, clock rates of up to 100 MHz are common, with faster rates found in some high–performance products. For DSP systems, the ratio of system clock rate to sample rate is one of the most important characteristics used to determine how the system will be implemented. The relationship between the clock rate and the sample rate partially determines the amount of hardware needed to implement an algorithm with a given complexity in real–time. As the ratio of sample rate to clock rate increases, so does the amount and complexity of hardware required to implement the algorithm.

4. Numeric Representations

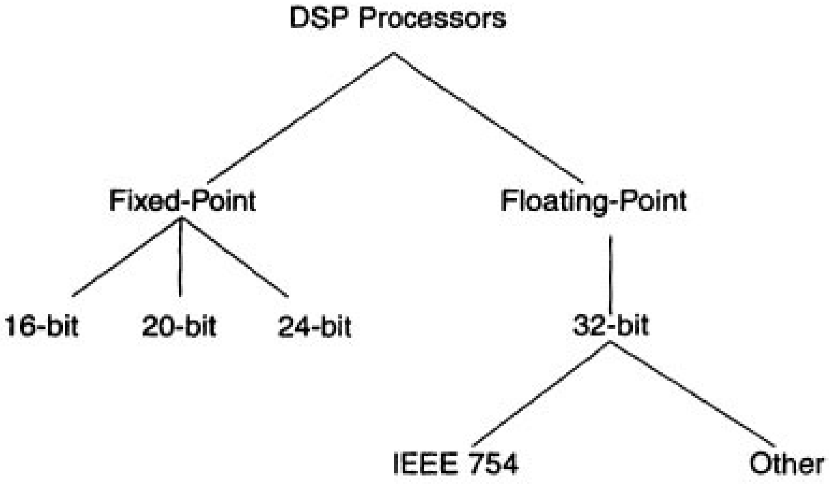


Figure 3: Numerical representation in DSP processors

Arithmetic operations such as addition and multiplication are at the heart of DSP algorithms and systems [39]. As a result, the numeric representations and type of arithmetic used can have a profound influence on the behavior and performance of a DSP system. The most important choice for the designer is between fixed–point and floating–point arithmetic. Fixed-point arithmetic represents numbers in a fixed range (e.g., −1.0 to +1.0) with a finite number of bits of precision (called the word width). For example, an eight-bit fixed–point number provides a resolution of 1/256 of the range over which the number is allowed to vary. Numbers outside of the specified range cannot be represented; arithmetic operations that would result in a number outside this range either saturate (that is, are limited to the largest positive or negative representable value) or wrap around (that is, the extra bits resulting from the arithmetic operation are discarded).

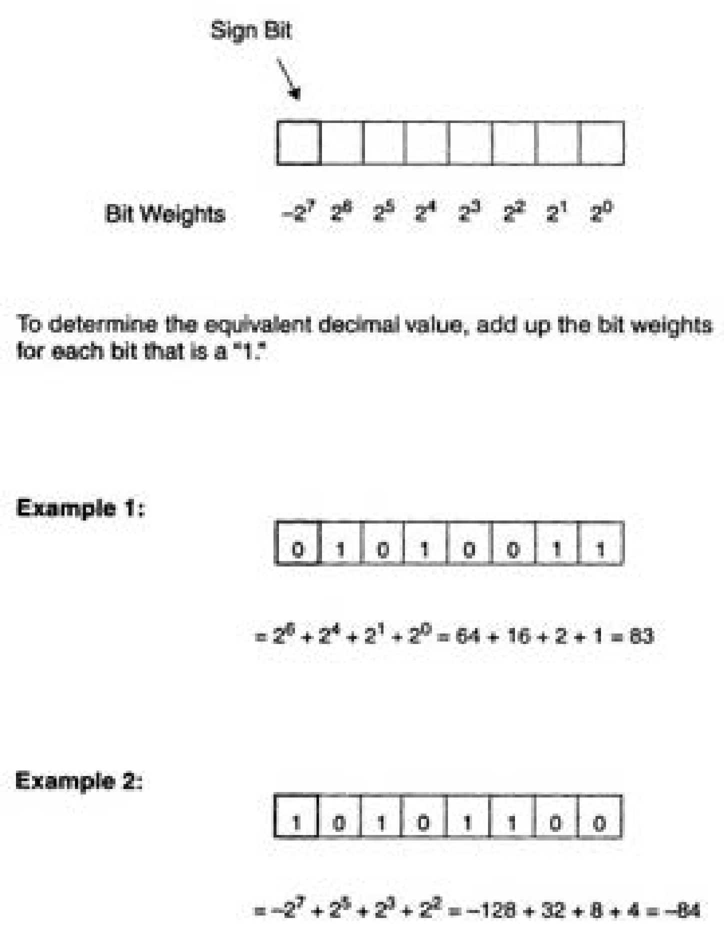
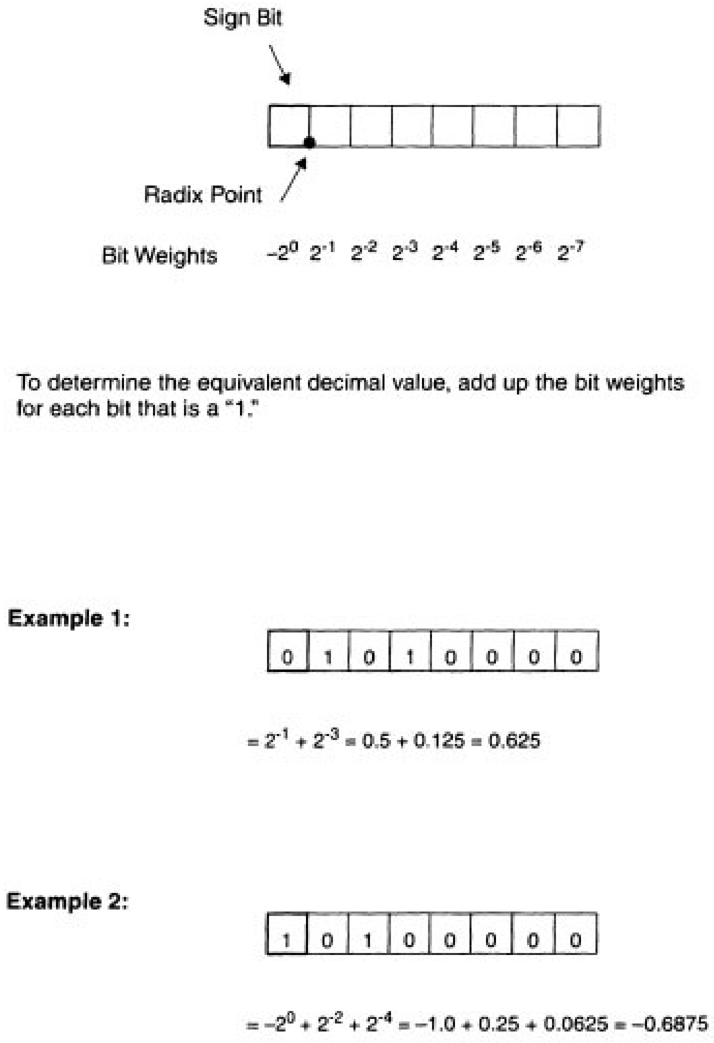


Figure 4: Simple binary integer representation

Floating–point arithmetic greatly expands the representable range of values. Floating–point arithmetic represents every number in two parts: a mantissa and an exponent. The mantissa is, in effect, forced to lie between -1.0 and $+1.0$, while the exponent keeps track of the amount by which the mantissa must be scaled (in terms of powers of two) in order to create the actual value represented.

That is: value = mantissa × base$^{\text{exponent}}$. Floating–point arithmetic provides much greater dynamic range (that is, the ratio between the largest and smallest value that can be represented) than fixed–point arithmetic. Because it reduces the probability of overflow and the necessity of scaling, it can considerably simplify algorithm and software design. Unfortunately, floating–point arithmetic is generally slower and more expensive than fixed–point arithmetic, and is more complicated to implement in hardware than



To determine the equivalent decimal value, add up the bit weights for each bit that is a "1."

$= 2^{-1} + 2^{-3} = 0.5 + 0.125 = 0.625$

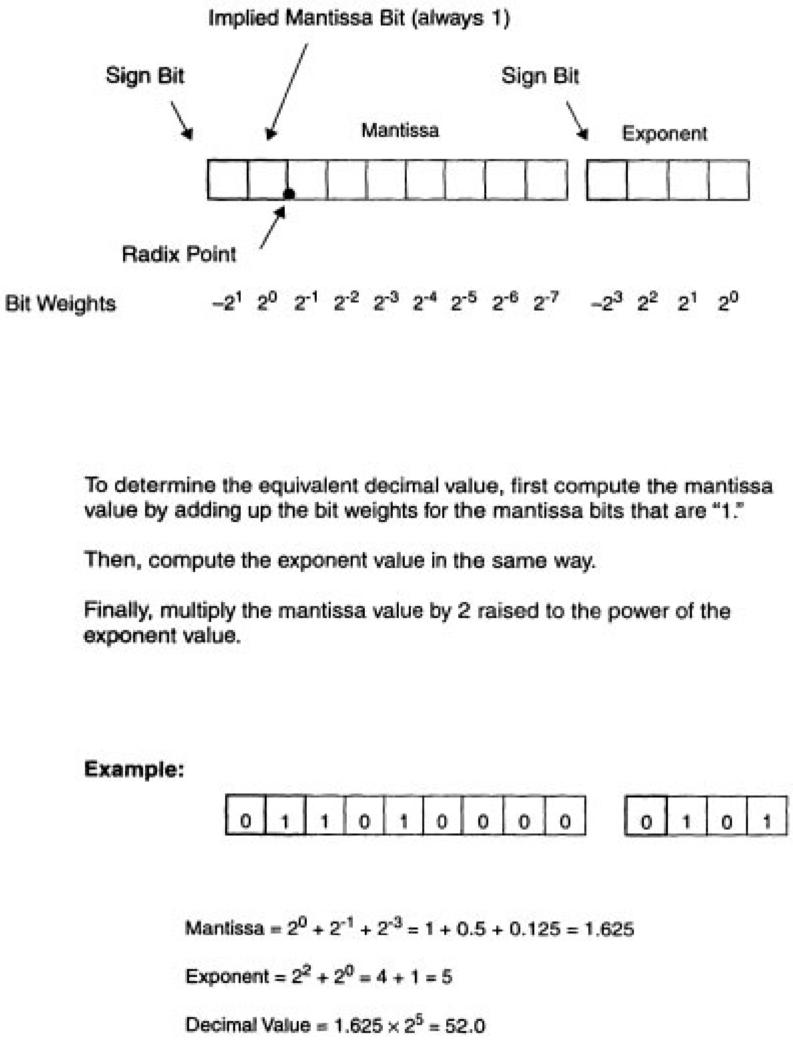
$= -2^0 + 2^{-2} + 2^{-4} = -1.0 + 0.25 + 0.0625 = -0.6875$

Figure 5: Simple binary fractional representation

fixed point arithmetic.

4.1.2.1 Fixed–Point Versus Floating–Point

The earliest DSP processors used fixed–point arithmetic, and in fact fixed–point DSPs still dominate today. The algorithms and hardware used to implement fractional arithmetic are virtually identical to those used for integer arithmetic. The main difference between integer and fractional arithmetic has to do with how the results of multiplication operations are handled. In practice, most fixed–point DSP processors support fractional arithmetic and integer arithmetic. The former is most useful for signal processing algorithms, while the latter is useful for control operations, address calculations, and other



To determine the equivalent decimal value, first compute the mantissa value by adding up the bit weights for the mantissa bits that are "1."

Then, compute the exponent value in the same way.

Finally, multiply the mantissa value by 2 raised to the power of the exponent value.

Mantissa = $2^0 + 2^{-1} + 2^{-3} = 1 + 0.5 + 0.125 = 1.625$

Exponent = $2^2 + 2^0 = 4 + 1 = 5$

Decimal Value = $1.625 \times 2^5 = 52.0$

Figure 6: Simplified binary floating–point representation, comprised of a mantissa (fraction part) and an exponent

operations that do not involve signals. With the floating–point representation instead, system designers have access to wider dynamic range (the ratio between the largest and smallest numbers that can be represented) and in many cases better precision.

Our definition of precision is based on the idea of the quantization error. This is the numerical error introduced when a longer numeric format is converted to a shorter one. The greater the possible quantization error relative to the size of the value represented, the less precision is available. For a fixed–point format, we define the maximum available precision to be equal to the number of bits in the format. For example, a 16–bit fractional format provides a maximum 16 bits of precision. This definition is based on computing the ratio of the size of the value represented to the size of the maximum quantization error that could be suffered when converting from a more precise representation via rounding. Formally stated

$$\text{maximum precision (in bits)} = \log_2 \text{(maximum value /maximum quantization error)}.$$

For a 16–bit fractional representation, the largest-magnitude value that can be represented is -1.0. When converting to a 16–bit fractional format from a more precise format via rounding, the maximum quantization error is 2^{-16}. Using the relation above, we can compute that this format has a maximum precision of $\log_2(1/2^{16})$, or 16 bits, the same as the format's overall width.

Note that if the value being represented has a smaller magnitude than the maximum, the precision obtained is less than the maximum available precision. This underscores the importance of careful signal scaling when using fixed–point arithmetic. Scaling is used to maintain precision by modifying the range of signal values to be near the maximum range of the numeric representation used.

Using this same definition for a floating–point format, the maximum available precision is the number of bits in the mantissa, including the implied integer bit. Because floating–point processors automatically scale all values so that the implied integer bit is equal to 1, the magnitude of the mantissa is restricted to be at least 1.0. This guarantees that the precision of any floating–point value is no less than half of the maximum available precision. Thus, floating–point processors maintain very good precision with no extra effort on the part of the programmer.

In practice, floating–point DSPs generally use a 32–bit format with a 24–bit mantissa and one implied integer bit, providing 25 bits of precision. Most fixed–point DSPs use a 16–bit format, providing 16 bits of precision. So, while in theory the choice between fixed and floating–point arithmetic could be independent of the choice of precision, in practice floating–point processors usually provide higher precision.

As mentioned above, dynamic range is defined as the ratio between the largest and smallest number representable in a given data format. It is in this regard that floating–point formats provide their key advantage. So, while using the same number of bits as the fixed–point format, the floating–point format provides dramatically higher dynamic range. In applications, dynamic range translates into a range of signal magnitudes that can be processed while maintaining sufficient fidelity. Different applications have different dynamic range needs. For telecommunications applications, dynamic range in the neighborhood of 50 dB is usually sufficient. For high-fidelity audio applications, 90 dB is a common benchmark. It's often helpful, though, if the processor's numeric representation and arithmetic hardware have somewhat more dynamic range than the application demands, as this frees the programmer from some of the painstaking scaling that may otherwise be needed to preserve adequate dynamic range.

Floating–point DSP processors are generally costlier than their fixed–point cousins, but easier to program. The increased cost results from the more complex circuitry required within the floating–point processor, which implies a larger chip. In addition, the larger word sizes of floating–point processors often means that off-chip buses and memories are wider, raising overall system costs.

The ease–of–use advantage of floating–point processors is due to the fact that in many cases the programmer does not have to be concerned about dynamic range and precision. On a fixed–point processor, in contrast, programmers often must carefully scale signals at various stages of their programs to ensure adequate numeric performance with the limited dynamic range and precision of the fixed–point processor.

Most high–volume, embedded applications use fixed–point processors because the priority is low cost. Programmers and algorithm designers determine the dynamic range and precision needs of their application, either analytically or through simulation, and then add scaling operations into the code if necessary. For applications that are less cost–sensitive, or that have extremely demanding dynamic range and precision requirements, or where ease of programming is paramount, floating–point processors have the advantage.

4.1.2.2 Native Data Word Width

The native data word width of a processor is the width of data that the processor's buses and data path can manipulate in a single instruction cycle. The size of the data word has a major impact on processor cost because it strongly influences the size of the chip and the number of package pins required as well as the size and number of external memory devices connected to the DSP. Therefore, designers try to use the chip with the smallest word size that their application can tolerate.

As with the choice between fixed–point and floating–point chips, there is often a trade-off between word size and development complexity. An application that appears to require 24–bit data for adequate performance can sometimes be coaxed into a 16–bit processor at the cost of more complex algorithms and/or programming.

4.1.2.3 Extended Precision

Extended precision means the use of data representations that provide higher precision than that of a processor's native data format. Extended precision can be obtained in two ways. First, many fixed and floating–point processors provide built-in support for an extended precision format for operations taking place exclusively within the data path of the processor. This means that as long as a series of arithmetic operations is carried out exclusively within the processor's data path and does not involve transferring intermediate results to and from memory, a data word width larger than the native data word width is available.

This allows a series of arithmetic operations to be performed using extra precision and/or dynamic range, with a final rounding operation performed when the result is stored to memory. Second, it's generally possible, though often painful, to perform multiprecision arithmetic by constructing larger data words out of sequences of native-width data words. For example, with a 16–bit fixed–point processor, a programmer can form 32–bit data words by stringing together pairs of 16–bit words. The programmer can implement multiprecision arithmetic operations by using the appropriate sequences of single-precision instructions. Of course, because each multiprecision arithmetic operation requires a sequence of single–precision instructions, multiprecision arithmetic is much slower than single-precision.

However, some processors provide features that ease multiprecision arithmetic. These include the ability to preserve the carry bit resulting from a single-precision addition operation for use as an input into a subsequent addition, and the ability to treat multiplication operands as signed or unsigned under program control. If the bulk of an application can be handled with single-precision arithmetic, but higher precision is needed for a small section of the code, then the selective use of multiprecision arithmetic may make sense. If most of the application requires higher precision, then a processor with a larger native data word size may be a better choice, if one is available.

4.1.2.4 Floating–Point Emulation and Block Floating–Point

Even when using a fixed–point processor, it is possible to obtain the precision and dynamic range of general-purpose floating–point arithmetic by using software routines that emulate the behavior of a floating–point processor. Some processor manufacturers provide a library of floating–point emulation routines for their fixed–point processors. If a library is not available, then the emulation routines must be written by the user. Floating–point routines are usually very expensive to execute in terms of processor cycles. This implies that floating–point emulation may be appropriate if only a very small part of the arithmetic computations in a given application require floating–point. If a significant amount of floating–point arithmetic is needed, then a floating–point processor is usually the appropriate choice. Another approach to obtaining increased precision and dynamic range for selected data in a fixed–point processor implementation is a block floating–point representation. With block floating–point, a group of numbers with different mantissas but a single, common exponent is treated as a block of data. Rather than store the exponent within part of each data word as is done with general purpose floating–point, the shared exponent is stored in its own separate data word. For example, a block of eight data values might share a common exponent, which would be stored in a separate data word. In this case, storage of an entire block of eight data values would require nine memory locations (eight for the mantissas and one for the exponent). Block floating–point is used to maintain greater dynamic range and precision than can be achieved with the processor's native fixed–point arithmetic formats. The conversion between the processor's native fixed–point format and block floating–point format is performed explicitly by the programmer through software. Some processors have hardware features to assist in the use of block floating–point formats. The most common of these is an "exponent detect" instruction. This instruction computes the shift needed to convert a high-precision intermediate result (for example, a value in an accumulator) to block floating–point format.

4.1.2.5 IEEE–754 Floating–Point

In 1985, the Institute of Electrical and Electronics Engineers released IEEE Standard 754 [IEE85], which defines standard formats for floating–point data representations and a set of standard rules for floating–point arithmetic. The rules specify, for example, the rounding algorithms that should be provided in a floating–point processor and how the processor should handle arithmetic exception conditions, such as divide by zero or overflow. A few commercial DSP processors provide partial hardware support for IEEE–754 floating–point formats and arithmetic. The Motorola DSP96002 features hardware support for single precision floating–point arithmetic as specified in IEEE–754. The Analog Devices ADSP–210xx family processors provide nearly complete hardware support for single-precision floating–point arithmetic as specified in the standard. Some other floating–point processors, such as the ATT DSP32xx, do not internally conform to IEEE–754, but do provide special hardware for fast conversion of numbers between the processor's internal floating–point representation and IEEE–754 representation. Hardware support for format conversion can be important in applications that require a non-IEEE–754-compliant DSP to interface with other processors that use the IEEE–754 representation. Without hardware conversion support, the noncompliant floating–point DSP must use software routines to convert between the different floating–point formats, and this software conversion can be quite time consuming. Therefore, developers of applications that require a DSP to interface with other processors that use the IEEE–754 representation should evaluate the practicality of software conversion carefully, or choose a processor with hardware conversion capabilities (or one that uses IEEE floating–point formats internally).

4.2 Custom Hardware

There are two important reasons why custom-developed hardware is sometimes a better choice than a DSP processor-based implementation [38]: performance and production costs. Just as DSP processors are more cost-effective for DSP applications than general-purpose processors because of their specialization, custom hardware has the potential to be even more cost-effective due to its more specialized nature. In applications with high sampling rates (for example, higher than 1/100th of the system clock rate), custom hardware may be the only reasonable approach.

For high volume products, custom hardware may also be less expensive than a DSP processor. This is because a custom implementation places in hardware only those functions needed by the application, whereas a DSP processor requires every application to pay for the full functionality of the processor, even if it uses only a small subset of its capabilities. Of course, developing custom hardware has some serious drawbacks in addition to these advantages. Most notable among these drawbacks are the effort and expense associated with custom hardware development, especially for custom chip design.

Custom hardware can take many forms. It can be a simple, small printed circuit board using off-the-shelf components, or it can be a complex, multiboard system, incorporating custom integrated circuits.

One of the most common approaches for custom hardware for DSP applications is to design custom printed circuit boards that incorporate a variety of off-the-shelf components. These components may include standard logic devices, fixed-function or configurable arithmetic units, field-programmable gate arrays (FPGAs), and function or application-specific integrated circuits (FASICs). As their name implies, FASICs are chips that are designed to perform a specific function, perhaps for a single application. Examples of FASICs include configurable digital filter chips, which can be configured to work in a range of applications, and facsimile modem chips, which are designed specifically to provide the signal processing functions for a fax modem and aren't useful for anything else.

As tools for creating custom chips improve and more engineers become familiar with chip design techniques, more companies are developing custom chips for their applications. Designing a custom chip provides the ultimate flexibility, since the chip can be tailored to the needs of the application, down to the level of a single logic gate. Of course, the benefits of custom chips and other hardware-based implementation approaches come with important trade-offs. Perhaps most importantly, the complexity and cost of developing custom hardware can be high, and the time required can be long. In addition, if the hardware includes a custom programmable processor, new software development tools will be required.

It is important to point out that the implementation options discussed here are not mutually exclusive. In fact, it is quite common to combine many of these design approaches in a single system, choosing different techniques for different parts of the system. One such hybrid approach, DSP core-based ASICs, was mentioned above. Others, such as the combination of an off-the-shelf DSP processor with custom ICs, FPGAs, and a general-purpose processor, are very common.

4.2.1 FPGA Architecture and Technology

FPGAs are a group of digital and user–programmable blocks (Gate Array), which are programmable in their functionalities and routing. As the acronym suggests, FPGAs grow out from gate arrays: which represent a particular digital technology that allows designers to realize tailored circuits on the basis of their needs, beginning from a standard architecture. This kind of technology called semi-custom differs from the full-custom one (like ASICs) where every single element is user-defined. Gate arrays are composed by a uniform logic gate matrix. Designers act on the final circuit, realized on the gate array, editing the last metal levels that link the defined logic gates. An FPGA device maintains some of the gate array features but its programming technique is totally different, indeed they are in-system programmable by users, directly on their workbench. High performances, low costs and a limited development time have decreed FPGA success in many applications. Most of FPGA applications reside in military projects, image processing, high–performance Digital Signal Processing (DSP) and other vector or matrix processing. The final result is a circuit suitable for designer necessities whose performances are very close to an ASIC development.

Figure 7: Example of an FPGA–based application

A first classification of FPGAs is done on the specific distribution of programmable elements and on the routing resources and logic: symmetrical FPGAs have the logic block distributed in a matrix and the routing logic passes horizontally and vertically between the programmable blocks. Otherwise row-based FPGAs are organized in parallel rows, with the logic gates along them, and the routing logic that horizontally crosses programmable block rows. Lastly, another group called 'hierarchical' is organized connecting wide programmable blocks through routing resources.

Blocks and connections are carried out using advanced VLSI technologies, so that reliability issues are pre-emptively solved. The lapse of time needed to implement a first prototype is very short, because it requires only a software programming. The FPGA design flow is not trivial and it is very similar to design flows for VLSI technology, but with the benefit of being short and stable. One of the fundamental advantages is the possibility of modifying design errors in a little while because of the different test simulations for every design layer thus resulting in a simplification in designers' work to verify immediately the efficiency of a particular solution. Digital circuits production, implemented through programmable devices, exhibits an economic benefit: in fact when an application has been validated and released on the market, every FPGA realization cost is constant for the producer, only the software development environment expense is amortized. FPGA and PLD design is very profitable in every kind of project that foresees few units production. On the other hand, the expense for an ASIC device is amortized when many units are implemented, because photolithographic masks are produced only once. For these reasons, the industry of prototypes is the main beneficiary of the FPGA' economic benefits: it is worthwhile to realize a first prototype using an FPGA and moving towards the production through an ASIC device.

4.2.2 Advantages of FPGAs

In recent years a particular improvement in FPGA size and performance has been noted thanks to a number of factors, including technological advancement of finer chip geometries, higher level of integration, the use of faster serial and communication links, specialized cores, enhanced logic and innovative design. Meanwhile, the overall performance growth curve of traditional microprocessors has flattened because of power density hurdles. At the same time, the number of processor cores has increased bringing the new issue of coping with optimal use of parallelism between processes while operating systems and automatic parallelization tools lag far behind. FPGAs' first benefit derives from their ability to handle massively parallel processing. FPGAs are able to operate at a modest clock rate as hundreds of megahertz, but they can realize tens of thousands of computations per clock cycle while operating in tens of watts range of power. A similar microprocessor may run at 1-2 GHz, but it would be very limited in the number of clock-cycle operations, typically four or eight operations per cycle. This means that an FPGA can provide a 50 to 100 times the performance per watt of power consumed by a microprocessor. Nevertheless FPGA computational strength, there are three key factors that determine the utility of FPGAs for a specific application. These factors are algorithm suitability, floating point vs. fixed point number representation and general FPGA software difficulty.

The first mention to raise is for which kind of algorithms FPGAs are best suited. FPGAs are thought to be used in problems that can be easily and efficiently divided into many parallel, often repetitive, computational tasks. On the other hand, there are some specific cases where FPGAs are not well exploited, such as target classification and moving target indication problems. Indeed repetitive operations are FPGAs strong points and generally they are used in predictable and static problems.

A second issue is the fact that FPGAs are not suited to floating point calculations, which microprocessors on the contrary address with well developed vector math engines. FPGAs are able to cope with this kind of calculation, but they require an undue amount of logic to implement: this causes a limit in calculation density of the FPGA and decrease its main computational benefits.

The last key factor is the degree of technical capabilities involved in software development and the talent and resources available for the work. On one hand the challenges of designing with traditional microprocessors are well established and familiar. On the other hand even if FPGA development tools have improved significantly over the last few years, it always takes a skilled user to develop code for an FPGA. This aspect is on the overtaking: in fact more and more often a lot of development environments provide a simple way of FPGA programming. This is the case of LabVIEW programming language that does not require a hardware description language (HDL) to develop FPGA set-up.

In addition to these three main arguments, other considerations can be taken into account for example which sensor interfaces are required by an application, that can be directed towards FPGAs or microprocessors. Often FPGAs cover all the different kind of communication standards, not thinkable for microprocessor limitations.

4.2.2.1 A Forward–Looking Architecture

In the past, systems tended to be homogeneous that is composed of one type of processing element. Today, designers have a wide range of products to choose from and can often mix–and–match computing components to get just the right mix of computing and I/O to meet their application specs. Using a hybrid FPGA-microprocessor architecture, it is possible to customize a system, providing FPGA components where they are useful with more DSPs or general-purpose microprocessors for the portions of application for which they are more suitable.

As FPGAs have grown larger and faster over the last decade, they have assumed a more central role in embedded processing applications. System–On–Chip is becoming a complete novel discipline able to let designers create new system solutions using the state of the art about what markets propose. An embedded system has some designed tasks well–known during its development, these will be executed through a hardware and software combination studied for that specific application. This is an important

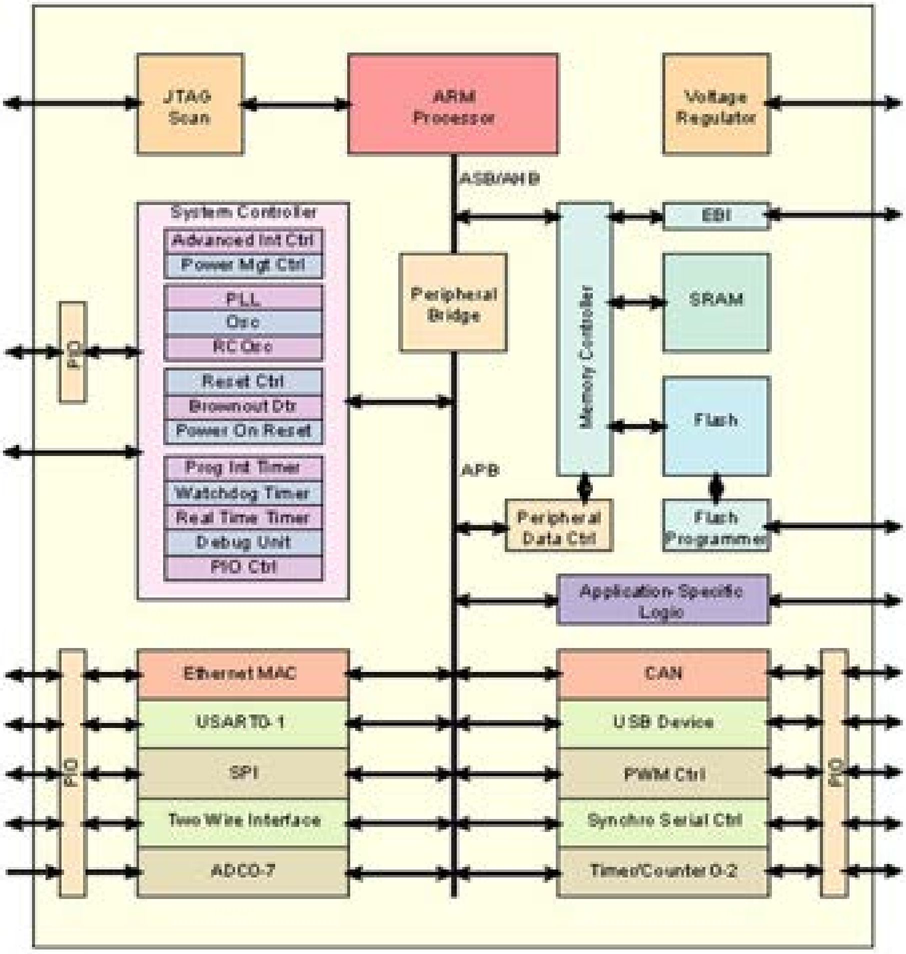


Figure 8: Base structure of a System–On–Chip architecture

benefit because the hardware resources can be heavily reduced to optimize the circuit occupation, consumption and costs. Furthermore the software counterpart execution is often real–time to allow users to reach a deterministic control of the system evolution.

A SoC implementation in FPGA technology permit to include the processors and a DSP cores inside the FPGA architecture, in order to manage a complete integrated instrument able to satisfy every requirement in several applications. The most important vendors of FPGA provide hard-core and soft-core devices easy to be introduced in the FPGA programming citeart:Minev-2007. The former are real physical areas where a microprocessor or a DSP core resides that are realized as a layout level, and it depends on the FPGA technology; the latter are a general hardware decription totally indipendent from the adopted technology, that are recognized by the FPGA programmer tool and automatically synthesized and inserted during the place and route phase.

Designers customize these soft cores and the surrounding logic to the task at hand. Recently, FPGA

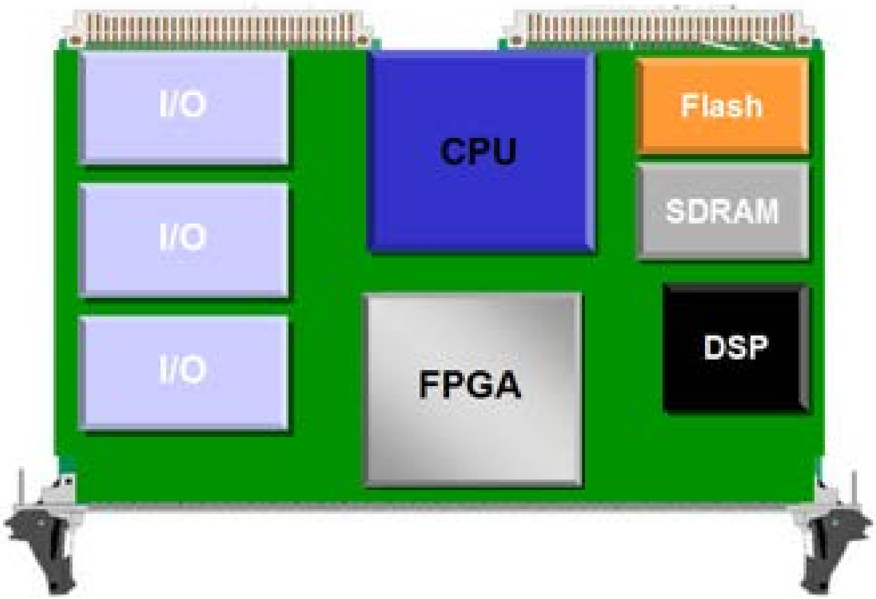


Figure 9: Plethora of components in a typical industrial PCB

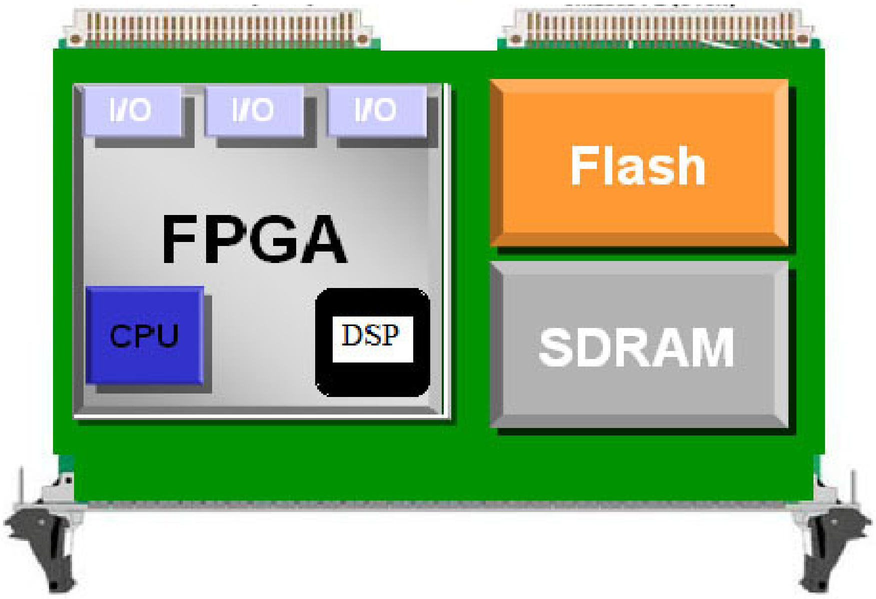


Figure 10: Novel FGPAs including DSP and CPU cores

vendors have taken this idea a step further, developing ICs that now include full ARM processing subsystems along with hardened peripherals, memory controllers, etc. all tightly integrated with the FPGA fabric. This marriage of hardened processor cores within the fabric gives designers the ease of programming with a familiar real time operating system (RTOS), yet has opened up new doors for customization of the overall processing system, with much tighter linkage between data and controlling processing.

4.3 FPGA in Control Schemes

Embedded control systems are found in a wide range of applications such as consumer electronics, medical equipment, robotics, automotive products, and industrial processes. For such systems, control algorithms are implemented as software programs that execute on a fixed architecture hardware processor.

The question that we must answer before we proceed is, with a plethora of embedded devices available for digital control: 'Why must one go in for embedded control using FPGA'? This can be answered by looking at the following advantages that FPGA possess. Most of computations in control involves the use of 2 operations. The first one being the Multiply operation and the second one being the accumulate operation. Together these operations are called Multiply ACcumulate (MAC) operations. The computational overhead is the maximum when any kind of digital controller is performing these operations. Hence the sampling rate and hence speed is limited by the rate at which the device performs these computations. In a general purpose microprocessor the processors resources are held up while it is busy performing these MAC operations and the speed or the sampling rate is decided by the latency of these instructions. In addition to this important feature, FPGAs can exploit other benefits in order to offer excellent parallelism, reconfigurable configuration and rapid prototyping. FPGAs are also fundamentals to implement PWM generators whose signals of high frequencies and precise duty–cycle resolution.

4.4 FPGA Versus DSP Processor Developing Control Loops

In order to ensure fair and square comparison between FPGA and general purpose processors, let us examine the operation of implementing a digital filter. It is a well known fact that many of the controllers that are designed are ultimately implemented as digital filters. Hence in order to illustrate the power of the FPGA, let us look at the specific implementation of a 256 tap filter on a typical DSP processor and an FPGA. The conventional DSP processor is a general purpose programming device that typically has 1–4 MAC units along with barrel shifters and other circuits optimized for efficient computations.

The conventional DSP is a serial device. Let us assume that it has got a single MAC unit. A 256 tap filter involves 256 MAC operations per sample. Hence with a single MAC unit, it takes 256 clock cycles for the output to be computed in a typical DSP processor. In order to improve the system throughput, we have to look at other options like using a high frequency clock generator. This increases the system complexity and the cost. Moreover the chances for clock skew occurring with high frequency clocks is also high. On the other hand, let us look at the same filter implemented on a typical FPGA.

Let us consider the most important feature of a FPGA–parallelism. The FPGA contains a large number of gates and millions of transistors. Hence we can implement the filter in a parallel manner.

The implementation consists of 256 registers and 256 multiplier units along with the addition of the final partial product. Hence what took 256 clock cycles in a DSP can be completed in a single clock cycle in an FPGA. This results in a tremendous improvement in the latency of each instruction.

Now let us look at some of the other features that FPGA based embedded control offers to us. The speed of a control system impacts its performance, stability, robustness and disturbance rejection characteristics. Faster control systems are typically more stable, easier to tune, and less susceptible to changing conditions and disturbances. To provide stable and robust management, a control system must be able to measure the process variable and set an actuator output command within a fixed period of time. The computational performance of the FPGA is so fast that the control loop rate is limited only by the sensors, actuators, and I/O modules. This is a stark contrast to traditional control systems, where the processing performance was typically the limiting factor. One of the most important parameters that is involved in performance measurement of digital control systems is loop cycle time. Loop cycle time is the time taken to execute one cycle of the control loop. It is the time that elapses between sampling the output, computing the controller output according to the control algorithm and sending the control signal to the actuator. Because of the inherent parallelism present in the FPGA, very low loop cycle times are possible.

Another common measure of control system performance and robustness is jitter, which is a measure of the variation of the actual loop cycle time from the desired loop cycle time. In general purpose operating systems such as Windows, the jitter is unbounded so closed loop control system stability cannot be guaranteed. Processor-based control systems with real–time operating systems are commonly able to guarantee control loop jitter of less than 100 microseconds. In FPGA based systems the control loop does not need to share hardware resources with other tasks and control loops can be precisely timed using the FPGA clock. The jitter for FPGA–based control loops depends on the accuracy of the FPGA clock source. It typically ranges in the order of picoseconds. The FPGA can effectively be used as a prototyping device in order to get the control algorithms fine tuned and running correctly. The wide of design tools available for FPGA's make it very easy in order to build a prototype of the control algorithm that we wish to implement and understand and refine the various issues like timing and signal integrity. One can even design the controller in a control systems design package like MATLAB or LabVIEW environment and use the VHDL or Verilog descriptions of the controller thus generated to fuse it on to the FPGA prototyping board. The FPGA thus plays a very important role in prototyping the controller even if the ultimate goal is the creation of an Application Specific Integrated Circuit (ASIC) controller for the application at hand. FPGA has another advantage in the fact that the design cycle time for the controller is less in an FPGA rather than an ASIC. In some cases it may be economical for the controller to be implemented in a FPGA rather than an ASIC. The FPGA also consumes lesser power than the microprocessor based or ASIC based controllers. The FPGA design now consists of the steps of creation, simulation, verification, synthesis, placement and routing of the design. A lot of computer based tools are available for this purpose, which is yet another argument in FPGA's favor. Hence we can safely arrive at a justification for the use of FPGA in control applications.

4.4.1 FPGA–Based PID Controller

In this section we give an example of how a controller can be implemented using FPGAs. The PID controller is the most used algorithm in industry. Also the controllers (8), (9) have PI terms. In a continuous domain the output is computed as follows

$$u(t) = k_p \left[e\,(t) + \frac{1}{T_i} \int_0^t e\,(t)\,dt + T_d \frac{de\,(t)}{dt} \right]$$

where k_p is the proportional gain, T_i is the reset time and T_d is the derivative time. In the FPGA technology, it is possible to realize two different typologies of PID controller, a serial design or a parallel one. In this paragraph a first comparison between parallel and serial structure has been done considering the resource utilization, speed and power consumption. The former equation is discretized, obtaining

$$u_k = k_p e_k + k_i \sum_{j=0}^{k-1} e_j + k_d (e_k - e_{k-1}) \tag{44}$$

where $\kappa_i = \kappa_p T/T_i$ is the integral coefficient and $\kappa_d = \kappa_p T_d/T$ is the derivative coefficient. This form is known as the position form of the PID algorithm. An alternative would be to compute u_k based on past output u_{k-1} and correction term $\mathit{\Delta} u_k$. This approach is often called as the velocity form of the PID algorithm. The first step in this regard would be to calculate u_{k-1} based on equation (44)

$$u_{k-1} = k_p e_{k-1} + k_i \sum_{j=0}^{k-1} e_j + k_d (e_{k-1} - e_{k-2}).$$

Then, one calculates the correction term as

$$\mathit{\Delta} u_k = u_k - u_{k-1} = k_0 e_{k-1} + k_1 e_{k-2} + k_2 e_{k-3}$$

where

$$k_0 = k_i + k_p + k_d, \qquad k_1 = -k_p - 2k_d, \qquad k_2 = k_d.$$

Hence, the current control output is calculated as

$$u_k = u_{k-1} + \mathit{\Delta} u_k = u_{k-1} + k_0 e_k + k_1 e_{k-1} + k_2 e_{k-2}.$$

The above equation is decomposed into its basic operations. Here p and p_d refers to the controlled variable and its desired value (set point) respectively. Moreover, p_0, p_1, p_2, s_1, s_2 are temporary variables.

$e_k = p + (-p_d)$

$p_0 = k_0 e_k$

$p_1 = k_1 e_{k-1}$

$p_2 = k_2 e_{k-2}$

$s_1 = p_0 + p_1$

$s_2 = p_2 + u_{k-1}$

$u_k = s_1 + s_2.$

For parallel design, each basic operation has got its own arithmetic unit either an adder or a multiplier. In serial design, which is mainly composed of sequential logic. All operations share only one adder and one multiplier.

4.4.2 Parallel Design

The parallel implementation uses 4 adders and 3 multipliers corresponding to the basic operations. The architecture diagram is shown in the following figure. The other circuitry includes registers for latching initial and intermediate values of error and output signals. The implementation also includes value limitation logic that keeps the signals generated by the control logic within limits that the physical device can bear.

4.4.3 Serial Design

In order to minimize the area and the resources consumed for the design, the serial design consists of only one adder and one multiplier. The other parts in the implementation include registers, multiplexers and circuits for arithmetic operations. They are commonly refereed to as the data–path circuits. Registers are used to store intermediate results. Because of the fact that the single adder multiplier unit is used in a time shared manner, there is the necessity of a control unit which is a finite state machine that sets the select lines of the multiplexers; thereby changing the input to the circuits. The results of those tests that have relevance to the problem are presented.

1. Resource Utilization: it was found that the serial implementation consumed far less resources on the FPGA than the parallel implementation. Even though the serial implementation includes a

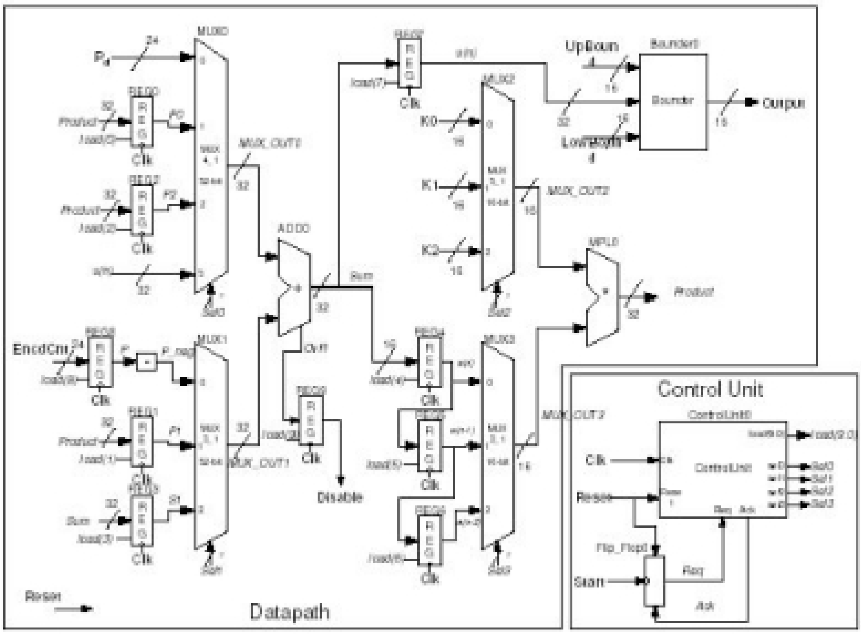


Figure 11: Serial Implementation of PID in FPGA

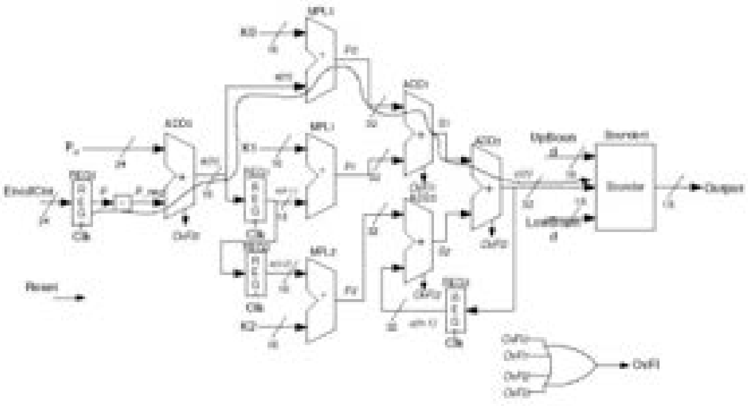


Figure 12: Parallel Implementation of PID in FPGA

control unit, it was found to consume far lesser number of CLBs to implement.

2. Speed: Test have been led with the Xilinx timing analyzer and it was found that in each design there were two timing concerns. The first one was the control clock frequency. This controlled the timing cycles of the PID algorithm. The next is the sampling frequency. This corresponds to the rate at which the control algorithm generates control signals; this is dependent on whether the implementation is a serial one or a parallel one. For the parallel implementation which is essentially

a combinational logic implementation, the sampling frequency and the control clock frequency are the same. This is a result of the inherently parallel nature of such an implementation. On the other hand, the serial algorithm requires four clock cycles to compute all the four basic operations specified in equations (3.9) – (3.14). Hence the sampling frequency for the serial implementation would be 1/4 of the control clock frequency.

3. Power Dissipation: The power dissipation increased as the sampling frequency was increased. At reasonable sampling frequencies, there wa no difference between the parallel and serial designs, even though the parallel design was expected to be more power efficient because of much lower sampling frequency.

4.4.4 A More Efficient PID

In the previous section we had looked at an implementation of a PID controller based on multipliers and adders. But when we are implementing PID controllers in LUT rich FPGA's, any design that does not make use of the memory rich characteristics of the FPGA is not an optimal implementation. It should however, be mentioned that this type of PID implementation is more efficient only in those kinds of FPGA that are rich in LUT's. An improved implementation of a PID Controller is based on Distributed Arithmetic (DA) concepts. The continuous PID equation (3.1) is modified as follows in order to avoid problems of spikes in the output because of the derivative term. These spikes occur when the user tries to change the set point abruptly. If the derivative term acts on the set point, then a sudden change in the set point would result in spikes in the output.

$$U(s) = K\left[bU_c(s) - Y(s) + \frac{1}{sT_i}(U_c(s) - Y(s)) - \frac{sT_d}{1 + \frac{sT_d}{N}}Y(s)\right]. \tag{45}$$

In (45), it is advantageous to allow only a fraction of the command signal act on the proportional part. Here k_i is the integral gain, k_d is the derivative gain, K is the proportional gain, U_c is the set point and Y is the process value. U is the controller output. Discretizing equation (45) by using the forward differences for the derivative term and backward differences for the integral term one has

$$u(kT) = P(kT) + I(kT) + D(kT)$$

where k denotes k–th sampling instant and

$$\begin{aligned}
P(kT) &= K(bu(kT) - y(kT)) \\
I(kT) &= I((k-1)T) + \frac{kT}{T_i}u((k-1)T) - y((k-1)T) \\
D(kT) &= \frac{T_d}{T_d + NT}(D(k-1)T) - \frac{KT_dN}{T_d + NT}(y(kT) - y((k-1)T))
\end{aligned}$$

where $y_k = y(kT)$ is the output at the current instant, $y_{k-1} = y((k-1)T)$ is the output at the previous instant, u_c is the desired output of the system, $I((k-1)T$ is the value of the integral term at the previous instant, $D((k-1)T)$ is the value of the derivative at the previous instant, K, b, T_i, T_d, N are controller parameters, T is the sampling time. The direct implementation of the above equation requires 5 multipliers, 5 adder subtractors and 4 delay elements. The multiplier based design is not efficient for FPGA implementation because of the fact that the FPGA has got limited number of CLB's for implementing the above logic circuits. A better implementation would be the DA Based implementation. Assuming that $u(kT)$, $u((k-1)T)$, $y(kT)$, $y((k-1)T)$ are m bit numbers and $[j]$ represents the j^{th} bit of these numbers, we obtain the following equations

$$P(kT) = \sum_{j=0}^{m-1}(k_b * u(kT)[j] - k * y(kT)[j]) * 2^j$$

$$I(kT) = \sum_{j=0}^{m-1}(I((k-1)T)[j] + \frac{kT}{T_i}(u((k-1)T)[j] - y(((k-1)T)[j]) * 2^j$$

$$D(kT) = \sum_{j=0}^{m-1}(\frac{T_d}{T_d + NT}D((k-1)T)[j] - \frac{kT_dN}{T_d + NT}((y(kT)[j] - y((k-1)T)[j])) * 2^j.$$

The results of

$$(k_b * u(kT)[j] - k * y(kT)[j])$$

$$(I((k-1)T)[j] + kT/T_i(u((k-1)T)[j] - y(((k-1)T)[j])$$

$$(T_d/T_d + NTD((k-1)T)[j] - kT_dN/T_d + NT((y(kT)[j] - y((k-1)T)[j]$$

are precomputed and stored in various look up tables. Using the three LUT's and corresponding shift add accumulators. The $P(kT)$, $D(kT)$, $I(kT)$ terms can be computed in m clock cycles. The main advantage of this method is the fact that it utilizes the LUT rich feature of the FPGA for computing the control effort.

The DA implementation for this particular implementation will consists of four delay blocks, 3 LUT's, 3 accumulators, 2 adders. Delay blocks are used to obtain $U((k-1)T)$ and $y(k-1)T$ respectively, whereas delay blocks are used to compute $D(k-1)T$ and $I(k-1)T$. Three LUT's and ACC's are used to provide the terms $P(kT)$, $I(kT)$, $D(kT)$ respectively. The ACC consists of an accumulator and an adder subtractor pair. Finally two adders produce the sum of $P(kT)$, $I(kT)$, $D(kT)$. The throughput of this implementation is $m + 1$ clock cycles, i.e. m clock cycles to compute U and one more clock cycle to update $I((k-1)T)$ and $D((k-1)T)$. Thus we find that the DA based implementation consumes far less number of logic resources than the parallel multiplier based design. Hence the design using DA would require 14 clock cycles to implement in comparison to the design based on multipliers that would take just a single clock cycle. Since power saving is dependent upon the clock frequency, the reduction in power consumption and the reduction in clock frequency would be advantageous in those applications which can tolerate the increased loop cycle time, resulting form the predominantly serial implementation of the DA based controller.

Conclusions

In this deliverable some aspects of the digital implementation of a control law on a physical device have been studied in order to reduce the deterioration of the control performances once implemented on a digital device, possibly bringing to unstable behaviors. Two aspects have been studied. The first deals with a self–triggered implementation, determining the sampling times necessary to implement the controller preserving the desired performance. The second deals with the physical device on which the control is implemented. Among the various possible solutions, we concentrated on the FPGA technology, which shows to be particularly interesting, especially in terms of parallel computation, control and digital signal processing.

References

[1] S. Di Gennaro and B. Castillo–Toledo, Comparative Study of Controllers for the Supervision, Control and Protection Systems in Pressurized Water Reactors of Evolutive Generation, Deliverable 2, PAR2010 Project, 2011.

[2] S. Di Gennaro and B. Castillo–Toledo, Performance Study of the Control Systems in the presence of Faults and/or Reference Accidents in Pressurized Water Reactors of Evolutive Generation, Deliverable 2, PAR2010 Project, 2011.

[3] I. F. Akyildiz, and I. H. Kasimoglu, *WirelessHART: Wireless sensor and actor networks: research challenges, Ad Hoc Networks*, Vol. 2, No. 4, pp. 351–367, 2004.

[4] A. Anta, and P. Tabuada, Self–Triggered Stabilization of Homogeneous Control Systems, *Proceedings of the 2008 American Control Conference – ACC 2008*, pp. 4129–4134, 2008.

[5] A. Anta, and P. Tabuada, To Sample or not to Sample: Self–Triggered Control for Nonlinear Systems, *IEEE Transactions on Automatic Control*, Vol. 55, No. 9, 2010.

[6] M. D. Di Benedetto, S. Di Gennaro, and A. D'Innocenzo, Digital Self Triggered Robust Control of Nonlinear Systems, *Proceeding of the* 50^{th} *Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, pp. 1674–1679, 2011.

[7] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw and D. Neöic. Networked Control Systems with Communication Constraints: Tradeoffs between Transmission Intervals, Delays and Performance. *IEEE Transactions on Automatic Control*, Vol. 55, No. 8, pp. 1781-1796, 2010.

[8] H. K. Khalil, *Nonlinear Systems*, Third Edition, Prentice Hall, Upper Saddle River, New Jersey, U.S.A., 2002.

[9] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.

[10] J. Kurzweil, On the Inversion of Liapunov's Second Theorem on Stability of Motion, *Translation of American Mathematical Society*, Vol. 24, pp. 19–77, 1963. Originally appeared on *Czechoslovak Mathematica Journal*, Vol. 81, pp. 217–259, 1956.

[11] M. Lemmon, T. Chantem, X. Hu, and M. Zyskowski, On Self–Triggered Full Information H–infinity Controllers, in *Hybrid Hybrid Systems: Computation and Control*, 2007.

[12] X. Wang, and M. Lemmon, Self–Triggered Feedback Control Systems With Finite–Gain $\mathcal{L}_2$ Stability, *IEEE Transactions on Automatic Control*, No. 3, Vol. 54, pp. 452-467, 2009.

[13] M. Mazo, and P. Tabuada, On Event–Triggered and Self–Triggered Control over Sensor/Actuator Networks, *Proceedings of the* 47^{th} *Conference on Decision and Control*, Cancun, Mexico, pp. 435–440, 2008.

[14] P. Tabuada, Event–Triggered Real–Time Scheduling of Stabilizing Control Tasks, *IEEE Transactions on Automatic Control*, Vol. 52, No. 9, pp. 1680–1685, 2007.

[15] M. Velasco, P. Marti, and J. Fuertes, The Self Triggered Task Model for Real–Time Control Systems, *Work–in–Progress Session of the* 24^{th} *IEEE Real–Time Systems Symposium – RTSS03*, pp. –, 2003.

[16] W. Aggoune, Contribution to the Stabilization of Stochastic Nonlinear Systems with Time Delays, *Proceedings of the* 18^{th} *IFAC World Congress* Milano, Italy, August 28–September 2, pp. 3885–3890, 2011

[17] W. Aggoune, On Feedback Stabilization of Stochastic Nonlinear Systems with Discrete and Distributed Delays, *Proceedings of the* 50^{th} *IEEE Conference on Decision and Control, and European Control Conference (CDC–ECC)*, Orlando, FL, USA, December 12–15, pp. 6296–6301, 2011.

[18] A. Anta and P. Tabuada, Exploiting Isochrony in Self–Triggered Control, *Submitted for publication, Preprint available on arXiv:1009.5208*, September 2010.

[19] L. Arnold, *Stochastic Differential Equations: Theory and Applications*, Wiley, 1972.

[20] K. J. Åström and B. Wittenmark, *Computer Controlled Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1990.

[21] K. E. Årzén, A Simple Event Based PID Controller, *Proceedings of the* 14^{th} *IFAC World Congress*, Vol. 18, pp. 423–428, 1999.

[22] W. P. Heemels, J. H. Sandee, and P. P. Bosch, Analysis of Event–Driven Controllers for Linear Systems, *International Journal of Control*, Vol. 81, No. 4, pp. 571–590, 2008.

[23] R. Z. Khasminskii, *Stochastic Stability of Differential Equations*, Sijthoff and Noordhoff, Alphen aan den Rijn, 1980.

[24] V.B. Kolmanovskii and A. Myshkis, *Applied Theory of Functional Differential Equations, Mathematics and Its Application*, Kluwer Academic Publishers, Dordrecht, 1992.

[25] H. J. Kushner, *Stochastic Stability and Control*, Academic Press, 1967.

[26] H. J. Kushner, Converse Theorem for stochastic Lyapunov functions, *SIAM Journal of Control and Optimization*, Vol. 5, pp. 228–233, 1967.

[27] X. Mao, *Stochastic Differential Equations and Applications*, Horwood, 1997.

[28] S.–E.A. Mohammed, *Stochastic Functional Differential Equations*, Longman, 1986.

[29] D. Nesic and L. Gruene, Lyapunov Based Continuous–Time Controller Redesign for Sampled–Data Implementation, *Automatica*, Vol. 41, No. 7, pp. 1143Ð1156, 2005.

[30] D. Nesic and A. R. Teel, Stabilization of Sampled–Data Nonlinear Systems via Backstepping on their Euler Approximate Model, *Automatica*, Vol. 42, No. 10, pp. 1801Ð1808, 2006.

[31] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, Using Deadbands to Reduce Communication in Networked Control Systems, *Proceedings of the American Control Conference 2002*, Vol. 4, pp. 3015–3020, 2002.

[32] P.E. Protter, *Stochastic Integration and Differential Equations*, 2^{nd} Ed., Springer, 2003.

[33] M. Velasco, J. Fuertes, and P. Marti, The Self Triggered Task Model for Real–Time Control Systems, *Proceedings of the Real–Time Systems Symposyum – RTSS03*, Work Progress Track, pp. 67–70, 2003.

[34] Y. Yamamoto, B. D. O. Anderson, and M. Nagahara, Approximating Sampled–Data Systems with Applications to Digital Redesign, *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, 2002, Vol. 4, pp. 3724Ð3729, 2002.

[35] T. Yoshizawa, *On the Stability of Solutions of a System of Differential Equations*, Memoirs of the College of Sciences, University of Kyoto, Ser. A, Vol. 29, pp. 27–33, 1955.

[36] X. Wang, and M. Lemmon, State Based Self–triggered Feedback Control Systems with L_2 Stability, *Proceedings of the* 17^{th} *IFAC World Congress*, pp. 15238–15243, 2008.

[37] X. Wang, and M. Lemmon, Self–triggered Feedback Control Systems with Finite–Gain L_2 Stability, *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, pp. 452-467, March 2009.

[38] S. Edwards, Microprocessor or FPGAs? Making the Right Choice, *RTC Magazine*, 2011.

[39] D. Strenski, P. Sundararajan, and Ralph Wittig, The Expanding Floating–Point Performance Gap Between FPGAs and Microprocessors, *HPC Wire*, 2010.

[40] P. B. Minev, and V. Stoianova Kukenska Implementation of Soft–Core Processors in FPGAs, *UNITECH'07*, 2007.

ALL. 3: Development of a Simulation Environment for the Analysis of Control System Performance for Performance and Safety Improvements of Novel Nuclear Plants

Center of Excellence DEWS

Department of Electrical and Information Engineering

University of L'Aquila, V. G. Gronchi 18, 67100, L'Aquila, Italy

Deliverable 2

Development of a Simulation Environment for the Analysis of Control System Performance for Performance and Safety Improvements of Novel Nuclear Plants

Sviluppo di un Sistema di Simulazione per l'Analisi delle Prestazione di Sistemi di Controllo per il Miglioramento della Prestazioni e della Sicurezza di Impianti Nucleari di Nuova Concezione

Authors:
Stefano Di Gennaro, Bernardino Castillo–Toledo, Fabrizio Memmi

Principal Investigator:
Prof. Stefano Di Gennaro

Project PAR 2011

July 31, 2012

Abstract
In this deliverable, the digital controllers designed in [3] have been tested in the Simulink© simulation environment, to check their performance, and to verify if the behavior of the controlled system is correct and satisfies the control specifications, for various values of the sampling time. To better check the real behavior of the closed loop system, the controllers designed and tested on the basis of such a simulation environment need to be further checked inserting some hardware in the loop. In particular, it has been considered the implementation of the control law from real process measurements. It has been analyzed an appropriate real–time environment where is possible to acquire and store data, compute the control algorithm, and apply the control action to actuators. Even though not certified for nuclear applications, it has been analyzed the popular LabVIEW© as a potential solution to interface Simulink with real–time environments. This solution allows checking the methodological steps for the real–time prototyping of the controllers, and can be used for future non–nuclear industrial applications, while for nuclear applications more costly nuclear–certified softwares, ensuring the same real–time performances of LabVIEW, have to be considered. Finally, it will be presented an experimental set–up used to show the benefit of the FPGA features and that has to be integrated with the Simulink/LabVIEW simulation environment, in order to better validate the designed control laws.

Riassunto
In questo documento sono stati testati i controllori digitali progettati in [3] nell'ambiente di simulazione Simulink©, per controllare la loro prestazione e verificare se il comportamento del sistema controllato è corretto e soddisfa le specifiche di controllo, per vari valori del tempo di campionamento. Per meglio controllare il reale comportamento del sistema a ciclo chiuso, i controllori progettati e testati sulla base di tale ambiente di simulazione devono essere ulteriormente testati inserendo nell'anello dei dispositivi fisici. In particolare è stata considerata l'implementazione della legge di controllo a partire da misure di un processo reale. È stato analizzato un ambiente a tempo reale appropriato ove è possibile acquisire e immgazzinate dati, calcolare l'algoritmo di controllo, e applicare l'azione di controllo agli attuatori. Sebbene non certificato per applicazioni nucleari, è stato analizzato il popolare LabVIEW© come potenziale soluzione per interfacciare Simulink con ambienti a tempo–reale. Questa soluzione permette di controllare i passi metodologici per una prototipizzazione a tempo–reale dei controllori, e può essere usata per future applicazioni industriali non nucleari, mentre per applicazioni nucleari devono essere considerati più costosi programmi certificati in campo nucleare, che assicurino le stesse prestazioni in tempo reale di LabVIEW. Infine sarà presentato uno schema sperimentale, usato per mostrare i vantaggi delle caratteristiche degli FPGA e che deve essere integrato con l'ambiente di simulazione in Simulink/LabVIEW, per meglio validare le leggi di controllo progettate.

1 A Simulation Environment for Performance Evaluation of Digital Controllers

The digital controllers designed in [3] need to be tested in simulation environment to check their performance, to verify if the behavior of the controlled system is correct and if it satisfies the control specifications. This environment is Matlab$^{©}$ (Matrix Laboratory), also used in [1] to test the continuous time controllers. More specifically, the toolbox Simulink$^{©}$ has been used, which has the advantage of an easy graphical visualization.

1.1 Some Recalls on The Simulink Environment

There exists a variety of softwares able to simulate dynamical systems, both commercial and non commercial. One of the most popular choice for simulating control systems, also in the industrial context, is Matlab$^{©}$ (Matrix Laboratory) of Mathworks, and its toolbox Simulink$^{©}$.

Simulink is an environment for multi–domain simulation and Model–Based Design for dynamic and embedded systems. Simulink provides an interactive graphical environment and a customizable set of block libraries that allow the design, simulation, implementation and test of a large number of systems arising in communication, control, signal, video and image processing, just to mention a few fields.

Simulink provides an extensive and expandable library of predefined blocks and a graphical editor for arranging these intuitive blocks into block diagrams. The user composes the block diagram of the system to be simulated by means of the interconnections among the elementary blocks, and Simulink automatically generates the implementation code.

Simulink is capable of interacting with Matlab, enabling full access to Matlab workspace for analyzing and visualizing results, customizing the modeling environment, as well as defining signal, parameter, and test data. Moreover, Matlab Function blocks fully exploit the powerful Matlab algorithms. These characteristics allow describing the behavior of complex dynamics thanks to control statements, cycles and other facilities, thus making the model design easier.

Additional key features are:

1. Model Explorer to navigate, create, configure, and search all signals,parameters, properties, and

generated code associated with the model;

2. Application Programming Interfaces (APIs) that allow the connection with other simulation programs and incorporate hand–written codes;

3. Simulation modes (Normal, Accelerator, and Rapid Accelerator) for running simulations interpretively or at compiled C–code speeds using fixed– or variable–step solvers;

4. Graphical debugger and profiler to examine simulation results and then diagnose performance and unexpected behavior in the model;

5. Model analysis and diagnostics tools to ensure model consistency and identify modeling errors.

Simulink provides a graphical user interface (GUI) for building models as block diagrams. The interactive graphical environment simplifies the modeling process, making the formulation of differential and difference equations dispensable.

In developing complex dynamical systems is often convenient, if not necessary, to split models into hierarchies of designed components, and make them communicate through input and output. Simulink models are hierarchical, thus being perfectly suited to such an approach. In this way, the models may be analyzed at different levels and according to their structural organization.

1.2 Implementation in Simulink of the Digital Controllers

A mathematical model of the primary circuit of a PWR has been implemented in Simulink in [1], where the reader can find the details of this implementation. In this section, the implementation of the digital controllers developed in [3] is described. As already discussed in [3], most of the modern controllers are realized by microprocessor–based digital circuits, or process–control computers. These devices can be characterized by discrete operations, where the control algorithms are implemented on a digital device.

The values of the sampled variables of the model have been used to implement the digital controllers for the pressurizer level and pressure. Figure 1 shows the Simulink diagram block representing the digital control system.

We first introduce and briefly describe the Simulink blocks used to build the control simulation system. Then, the single blocks composing the control system are analyzed and commented.

The library blocks used in the system are

1. Embedded Matlab Function: is the main library used in the model, it allows you to implement a MATLAB function (with input and output) in the Simulink environment.

2. Integrator: foundation library for the simulation of dynamic models. Returns the integral of the output signal which receives in input, at the time instant current. One can use variety of methods of numericalintegration for the calculation of the output.

3. Unit Delay: it delays its input by the specified sample period. This block is equivalent to the z-1 discrete-time operator. The block accepts one input and generates one output, which can be either both scalar or both vector. If the input is a vector, all elements of the vector are delayed by the same sample period.

4. Subsystem block: represents a subsystem of the system that contains it. The Subsystem block can represent a virtual subsystem or a nonvirtual subsystem.

5. Zero–order hold: this block samples and holds the input for the sample period you specify. The block accepts one input and generates one output. Each signal can be scalar or vector. If the input is a vector, the block holds all elements of the vector for the same sample period.

6. From: this block picks up the signal from the relative block ?Goto? and passes it on in output. Allows the passage of signals between blocks without connecting them.

7. Goto: transfers the input signal to the corresponding block From.

8. Input port: creates a port for subsystem or external inputs. It represents a link inside and outside of the system.

9. Output port: creates an output port for subsystem or external output. It represents a link between inside and outside the system.

10. Switch: this block performs a switching of out between the first and the third signal input places, via the directives of the control signal (2nd signal).

11. Signal to Workspace: writes the data obtained from simulations, within a structure in the main MATLAB workspace.

12. Scope: graphics the variable value at the time of simulation. Allows viewing multi-axis with respect to the same time range.

With respect to the simulation environment considered in [1], the scheme of figure 1 presents the following differences

- A digital inventory mass controller;

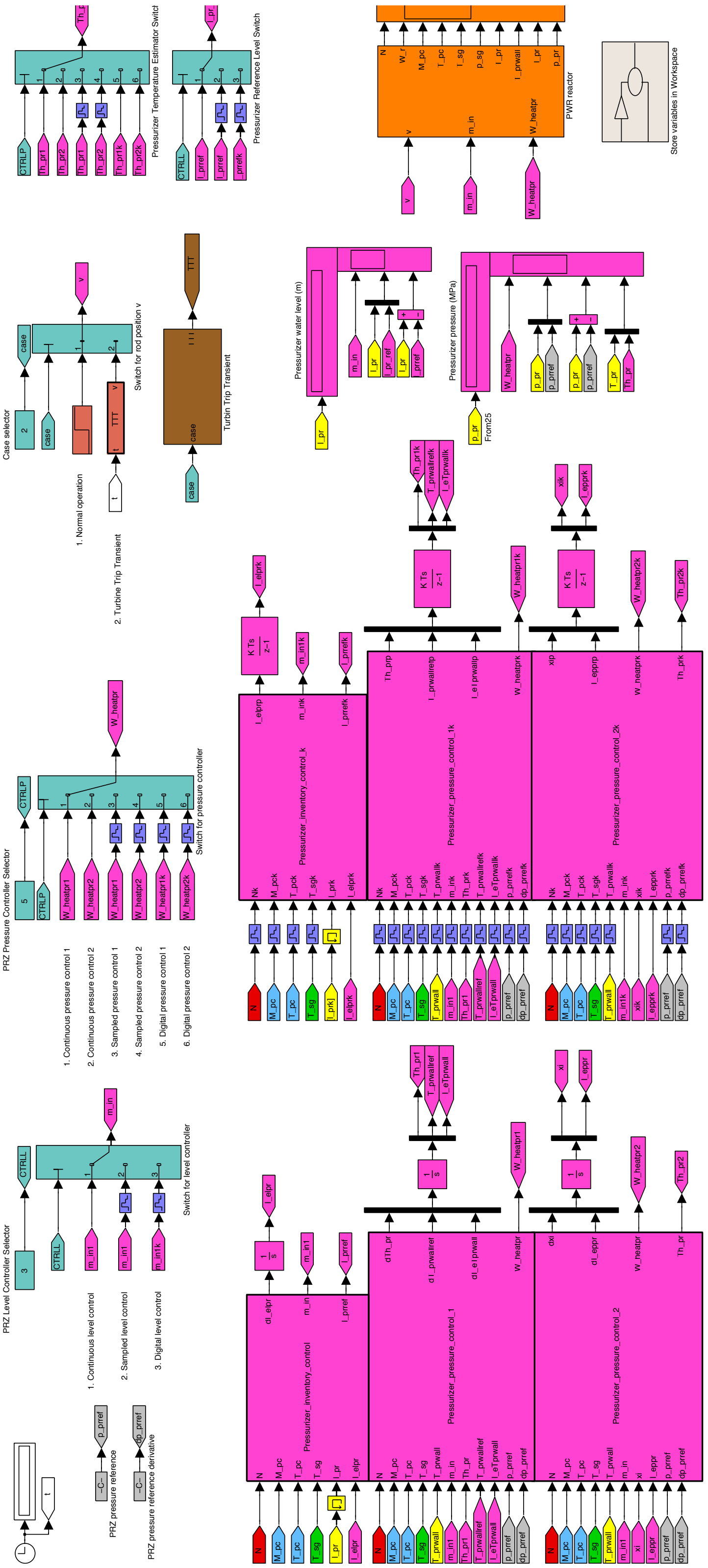

PRZ Level Controller Selector
PRZ Pressure Controller Selector
Case selector
Switch for level controller
Switch for pressure controller
Switch for rod position v
Pressurizer Temperature Estimator Switch
Pressurizer Reference Level Switch
1. Continuous level control
2. Sampled level control
3. Digital level control
1. Continuous pressure control 1
2. Continuous pressure control 2
3. Sampled pressure control 1
4. Sampled pressure control 2
5. Digital pressure control 1
6. Digital pressure control 2
1. Normal operation
2. Turbine Trip Transient
Turbin Trip Transient
PRZ pressure reference
PRZ pressure reference derivative
Pressurizer_inventory_control
Pressurizer_pressure_control_1
Pressurizer_pressure_control_2
Pressurizer_inventory_control_k
Pressurizer_pressure_control_1k
Pressurizer_pressure_control_2k
Pressurizer water level (m)
Pressurizer pressure (MPa)
From25
PWR reactor
Store variables in Workspace
K Ts
z−1
1
s

Figure 1: Digital control scheme for the primary circuit of a NPP

- Two digital pressurizer pressure controllers;
- Switches and displays.

An exhaustive description of the previous simulation environment is given in [1], which analyzes in detail the blocks corresponding to the PWR reactor dynamics, the continuous–time pressurizer water level control, the continuous–time pressurizer pressure controllers, the rod position control, and the various function performed by switches and displays visible in the scheme. Here we will describe the new blocks added to implement the digital controllers. Figure 5 shows the block `Pressurizer_inventory _control_k`, which represents the implementation of the digital inventory mass controller

$$
\begin{aligned}
I_{e_{l_{pr}},k+1} &= I_{e_{l_{pr}},k} + \delta(l_{pr,k} - l_{pr,\mathrm{ref},k}) \\
m_{in,k} &= \frac{A_{pr}}{\psi(M_{pc,k},T_{pc,k})}\Bigg[-\Big(k_p(l_{pr,k} - l_{pr,\mathrm{ref},k}) + k_i I_{e_{l_{pr}},k}\Big)\varphi^2(T_{pc,k}) + m^{\circ}_{out}\varphi(T_{pc,k}) \\
&\quad + \frac{1}{c_{p,pc}}\left(\frac{2c_{r,1}}{M_{pc,k}}\varphi^2(T_{pc,k}) + \left.\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\right|_k\right)\Big(c_{p,pc}m^{\circ}_{out}\Delta^{\circ} + c_{\psi}N_k - n_{sg}k_{t,sg}(T_{pc,k} - T_{sg,k}) - W^{\circ}_{loss,pc}\Big)\Bigg]
\end{aligned} \tag{1}
$$

with $k_p, k_i > 0$,

$$
\begin{aligned}
\psi(M_{pc,k},T_{pc,k}) &= \varphi(T_{pc,k}) - (T^{\circ}_{pc,i} - T_{pc,k})\left.\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\right|_k - \frac{2c_{r,1}A_{pr}}{M_{pc,k}}(T^{\circ}_{pc,i} - T_{pc,k})\varphi^2(T_{pc,k}) \\
\varphi(T_{pc,k}) &= c_{\varphi,0} + c_{\varphi,1}T_{pc,k} - c_{\varphi,2}T^2_{pc,k} \\
\left.\frac{\partial\varphi(T_{pc})}{\partial T_{pc}}\right|_k &= c_{\varphi,1} - 2c_{\varphi,2}T_{pc,k}.
\end{aligned}
$$

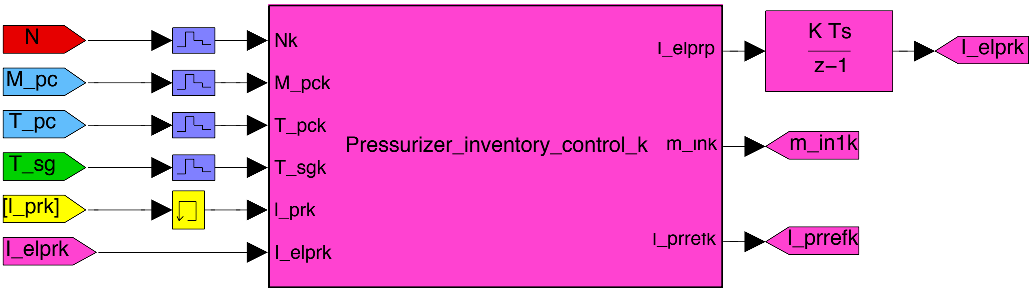


Figure 2: Pressurizer Discrete Inventory Mass Control

The corresponding EMF code is given in Table 1.

Table 1 – Digital Inventory Control (EMF code)

```
function [I_elprp,m_ink,l_prrefk]=Pressurizer_inventory_control_k(Nk,M_pck,
                                                  T_pck,T_sgk,l_prk,I_elprk)
%#eml
%--------------------------------------------------------------------
% Initialization of the variables

% System parameters
Delta=0;
A_pr=0;
c_ppc=0;
c_psi=0;
c_phi0=0;
c_phi1=0;
c_phi2=0;
n_sg=0;
k_tsg=0;

% Reference parameters
c_r1=0;
c_r2=0;

% Perturbation parameters
m_out0=0;
T_pci0=0;
Delta0=0;
W_losspc0=0;

% Controller parameters
k_p=0;
k_i=0;
```

```
% Actuator parameter
minmax=0;

% Sampling period
delta=0;

%--------------------------------------------------------------------
% Loading current parameters from workspace
eml.extrinsic('evalin');
%eml.extrinsic('assignin');

% Load parameters from workspace
% System parameters
Delta=evalin('base','Delta');
A_pr=evalin('base','A_pr');
c_ppc=evalin('base','c_ppc');
c_psi=evalin('base','c_psi');
c_phi0=evalin('base','c_phi0');
c_phi1=evalin('base','c_phi1');
c_phi2=evalin('base','c_phi2');
n_sg=evalin('base','n_sg');
k_tsg=evalin('base','k_tsg');

% Reference parameters
c_r1=evalin('base','c_r1');
c_r2=evalin('base','c_r2');

% Perturbation parameters
m_out0=evalin('base','m_out0');
T_pci0=evalin('base','T_pci0');
Delta0=evalin('base','Delta0');
W_losspc0=evalin('base','W_losspc0');
```

```
% Controller parameters
k_p=evalin(’base’,’k_p’);
k_i=evalin(’base’,’k_i’);

% Actuator parameter
minmax=evalin(’base’,’minmax’);

% Sampling period
delta=evalin(’base’,’delta’);

%------------------------------------------------------------------
% Digital Pressurizer Level Control
% Cold and hot leg temperatures
T_pcclk=T_pck-Delta;
T_pchlk=T_pck+Delta;

% Level reference (see Pisa’s report)
lprrefk=c_r1*(T_pcclk+T_pchlk)-c_r2;
if lprrefk<0
    l_prrefk=0;
else
    l_prrefk=lprrefk;
end

% Function \phi and its derivative
phik=c_phi0+c_phi1*T_pck-c_phi2*T_pck^2;
dphik=c_phi1-2*c_phi2*T_pck;

% \psi function
psik=phik-(T_pci0-T_pck)*dphik-2*c_r1*A_pr*(T_pci0-T_pck)*phik^2/M_pck;

% Integral term
I_elprp=l_prk-l_prrefk;
```

```
% Input m_{in}
mink=A_pr*((2*c_r1*phik^2/M_pck+dphik)*(c_ppc*m_out0*Delta0+c_psi*Nk
     -n_sg*k_tsg*(T_pck-T_sgk)-W_losspc0)/c_ppc-(k_p*(l_prk-l_prrefk)
     +k_i*I_elprk)*phik^2+m_out0*phik)/psik;
if mink<=0,
    m_ink=0;
elseif mink>=minmax;
    m_ink=minmax;
else
    m_ink=mink;
end
```

Table 1 – Digital Inventory Control (EMF code)

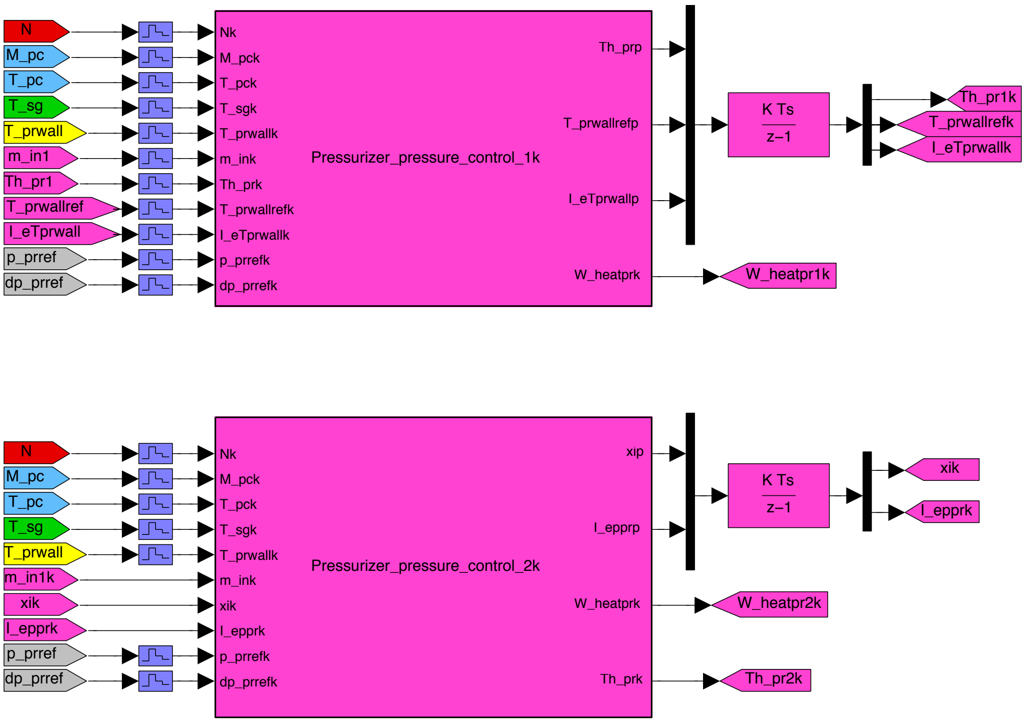


Figure 3: Digital pressurizer pressure controllers

In Figure 3 the `Pressurizer_pressure_controller_1k` and the `Pressurizer_pressure` _controller_2k blocks are shown. These blocks simulate the implementation of the digital controllers

$$
\begin{aligned}
\hat{T}_{pr,k+1} &= \hat{T}_{pr,k} + \delta\Bigg[-\left(\frac{m^{\circ}_{pr,k}}{M_{pr,k}} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}}\right)\hat{T}_{pr,k} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}} T_{pr,wall,k} \\
&\qquad + \frac{1}{c_{p,pr}M_{pr,k}} W_{heat,pr,k} + \frac{c_{p,pc}m^{\circ}_{pr,k}}{c_{p,pr}M_{pr,k}}(T_{pc,k} + \mathit{\Delta}^{\circ})\Bigg] \\
T_{pr,wall,\mathrm{ref},k+1} &= T_{pr,wall,\mathrm{ref},k} + \delta\left(\frac{k_{wall}}{c_{p,wall}} T_{pr,\mathrm{ref},k} - \frac{k_{wall}}{c_{p,wall}} T_{pr,wall,\mathrm{ref},k} + k_i I_{e_{T_{pr,wall}},k} - \frac{1}{c_{p,wall}} W^{\circ}_{loss,pr}\right) \\
I_{e_{T_{pr,wall}},k+1} &= I_{e_{T_{pr,wall}},k} + \delta\left(T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k}\right) \\
W_{heat,pr,k} &= -k_{wall}(T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k}) - \frac{c_{p,pr}}{c_{p,wall}} k_{wall} M_{pr,k} B^T P \begin{pmatrix} I_{e_{T_{pr,wall}},k} \\ T_{pr,wall,k} - T_{pr,wall,\mathrm{ref},k} \end{pmatrix} \\
&\quad + c_{p,pr}M_{pr,k}\Bigg[\dot{T}_{pr,\mathrm{ref}}\Big|_k + \left(\frac{m^{\circ}_{pr,k}}{M_{pr,k}} + \frac{k_{wall}}{c_{p,pr}M_{pr,k}}\right) T_{pr,\mathrm{ref},k} - \frac{k_{wall}}{c_{p,pr}M_{pr,k}} T_{pr,wall,\mathrm{ref},k} \\
&\qquad - \frac{c_{p,pc}m^{\circ}_{pr,k}}{c_{p,pr}M_{pr,k}}(T_{pc,k} + \mathit{\Delta}^{\circ})\Bigg] \\
T_{pr,\mathrm{ref},k} &= \frac{c_1 + \sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref},k})}}{2c_2}, \qquad \dot{T}_{pr,\mathrm{ref}}\Big|_k = \frac{2\dot{p}_{pr,\mathrm{ref},k}}{\sqrt{c_1^2 - 4c_2(c_0 - p_{pr,\mathrm{ref},k})}}
\end{aligned}
$$

and

$$
\begin{aligned}
I_{e_{p_{pr}},k+1} &= I_{e_{p_{pr}},k} + \delta\left(c_0 - c_1\hat{T}_{pr} + c_2\hat{T}^2_{pr} - p_{pr,\mathrm{ref}}\right) \\
\xi_{k+1} &= \xi_k + \delta\left(\hat{T}_{pr} - T_{pr,wall} - \frac{1}{k_{wall}} W^{\circ}_{loss,pr} - \frac{1}{k}\frac{1}{c_{p,pr}M_{pr}} C_{pr}\right) \\
\hat{T}_{pr,k} &= \kappa\left(\frac{c_{p,wall}}{k_{wall}} T_{pr,wall,k} - \xi_k\right) \\
C_{pr,k} &= \frac{c_{p,pr}M_{pr,k}}{-c_1 + 2c_2\hat{T}_{pr,k}}\left(\dot{p}_{pr,\mathrm{ref},k} - K_p\left(c_0 - c_1\hat{T}_{pr,k} + c_2\hat{T}^2_{pr,k} - p_{pr,\mathrm{ref},k}\right) - K_i I_{e_{p_{pr}},k}\right) \\
W_{heat,pr,k} &= k_{wall}(\hat{T}_{pr,k} - T_{pr,wall,k}) + C_{pr,k} + \delta_{pr}\left(c_{p,pr}m^{\circ}_{pr,k}\hat{T}_{pr,k} - c_{p,pc}m^{\circ}_{pr,k}(T_{pc,k} + \mathit{\Delta}^{\circ})\right)
\end{aligned}
$$

respectively. Their EMF codes are reported in Tables 2 and 3.

Table 2 – Digital pressurizer pressure control 1k (EMF code)

```
function [Th_prp,T_prwallrefp,I_eTprwallp,W_heatprk]
=Pressurizer_pressure_control_1k(Nk,M_pck,T_pck,T_sgk,T_prwallk,m_ink,Th_prk,
                                  T_prwallrefk,I_eTprwallk,p_prrefk,dp_prrefk)
%#eml
```

```
%--------------------------------------------------------------------------
%Inizialization of variables

% System parameters
c_ppr=0;
c_ppc=0;
c_psi=0;
c_phi0=0;
c_phi1=0;
c_phi2=0;
c0=0;
c1=0;
c2=0;
n_sg=0;
k_tsg=0;
c_pwall=0;
k_wall=0;
V_pc0=0;

% Perturbation parameters
W_losspr0=0;
W_losspc0=0;
m_out0=0;
T_pci0=0;
Delta0=0;

% Actuator parameter
Wheatmax=0;

% Controller parameter
k_di1=0;
Pd=zeros(2,2);
```

```
% Sampling period
delta=0;

%--------------------------------------------------------------------------
% Loading current parameters from workspace
eml.extrinsic('evalin');
%eml.extrinsic('assignin');

% Load parameters from workspace
% System parameters
c_ppr=evalin('base','c_ppr');
c_ppc=evalin('base','c_ppc');
c_psi=evalin('base','c_psi');
c_phi0=evalin('base','c_phi0');
c_phi1=evalin('base','c_phi1');
c_phi2=evalin('base','c_phi2');
c0=evalin('base','c0');
c1=evalin('base','c1');
c2=evalin('base','c2');
n_sg=evalin('base','n_sg');
k_tsg=evalin('base','k_tsg');
c_pwall=evalin('base','c_pwall');
k_wall=evalin('base','k_wall');
V_pc0=evalin('base','V_pc0');

% Perturbation parameters
W_losspr0=evalin('base','W_losspr0');
W_losspc0=evalin('base','W_losspc0');
m_out0=evalin('base','m_out0');
T_pci0=evalin('base','T_pci0');
Delta0=evalin('base','Delta0');

% Actuator parameter
```

```
Wheatmax=evalin(’base’,’Wheatmax’);

% Controller parameter
k_di1=evalin(’base’,’k_di1’);
Pd=evalin(’base’,’Pd’);

% Sampling period
delta=evalin(’base’,’delta’);

%---------------------------------------------------------------------------
% Digital Pressurizer Pressure Controller 1
T_prrefk=(c1+sqrt(c1^2-4*c2*(c0-p_prrefk)))/(2*c2);
dT_prrefk=2*dp_prrefk/(sqrt(c1^2-4*c2*(c0-p_prrefk)));
density_pck=c_phi0+c_phi1*T_pck-c_phi2*T_pck^2;
derdensity_pck=c_phi1-2*c_phi2*T_pck;
M_prk=M_pck-density_pck*V_pc0;
m_pr0k=m_ink-m_out0-derdensity_pck*V_pc0*(c_ppc*m_ink*(T_pci0-T_pck)
      +c_ppc*m_out0*Delta0+c_psi*Nk-n_sg*k_tsg*(T_pck-T_sgk)
      -W_losspc0)/(c_ppc*M_pck);
if m_pr0k>0,
    dprk=1;
else
    dprk=0;
end
W_heatprrefk=c_ppr*M_prk*dT_prrefk+k_wall*(T_prrefk-T_prwallrefk)
           -dprk*(c_ppc*m_pr0k*(T_pck+Delta0)-c_ppr*m_pr0k*T_prrefk);
BPdxk=[0 1]*Pd*[I_eTprwallk;T_prwallk-T_prwallrefk];
W_hk=-k_wall*(T_prwallk-T_prwallrefk)-c_ppr*k_wall*M_prk*BPdxk+W_heatprrefk;
if W_hk<0
    W_heatprk=0;
elseif W_hk>Wheatmax,
    W_heatprk=Wheatmax;
else
```

```
    W_heatprk=W_hk;
end
Th_prp=(-k_wall*(Th_prk-T_prwallk)+W_heatprk+dprk*(c_ppc*m_pr0k*(T_pck+Delta0)
       -c_ppr*m_pr0k*Th_prk))/(c_ppr*M_prk);
T_prwallrefp=k_wall*(T_prrefk-T_prwallrefk)/c_pwall+k_di1*I_eTprwallk
       -W_losspr0/c_pwall;
I_eTprwallp=T_prwallk-T_prwallrefk;
```

Table 2 – Digital pressurizer pressure control 1k (EMF code)

Table 3 – Digital pressurizer pressure control 2k (EMF code)

```
function [xip,I_epprp,W_heatprk,Th_prk]=Pressurizer_pressure_control_2k(Nk,
          M_pck,T_pck,T_sgk,T_prwallk,m_ink,xik,I_epprk,p_prrefk,dp_prrefk)
%#eml
%-------------------------------------------------------------------------
%Inizialization of variables


% System parameters
c_ppr=0;
c_ppc=0;
c_psi=0;
c_phi0=0;
c_phi1=0;
c_phi2=0;
c0=0;
c1=0;
c2=0;
n_sg=0;
k_tsg=0;
c_pwall=0;
k_wall=0;
```

```
V_pc0=0;

% Perturbation parameters
W_losspr0=0;
W_losspc0=0;
m_out0=0;
T_pci0=0;
Delta0=0;

% Controller parameters
Kdp=0;
Kdi=0;
kd=0;

% Actuator parameter
Wheatmax=0;

% Sampling period
delta=0;

%-------------------------------------------------------------------------
% Loading current parameters from workspace
eml.extrinsic('evalin');
%eml.extrinsic('assignin');

% Load parameters from workspace
% System parameters
c_ppr=evalin('base','c_ppr');
c_ppc=evalin('base','c_ppc');
c_psi=evalin('base','c_psi');
c_phi0=evalin('base','c_phi0');
c_phi1=evalin('base','c_phi1');
c_phi2=evalin('base','c_phi2');
```

```
c0=evalin(’base’,’c0’);
c1=evalin(’base’,’c1’);
c2=evalin(’base’,’c2’);
n_sg=evalin(’base’,’n_sg’);
k_tsg=evalin(’base’,’k_tsg’);
c_pwall=evalin(’base’,’c_pwall’);
k_wall=evalin(’base’,’k_wall’);
V_pc0=evalin(’base’,’V_pc0’);

% Perturbation parameters
W_losspr0=evalin(’base’,’W_losspr0’);
W_losspc0=evalin(’base’,’W_losspc0’);
m_out0=evalin(’base’,’m_out0’);
T_pci0=evalin(’base’,’T_pci0’);
Delta0=evalin(’base’,’Delta0’);

% Controller parameters
Kdp=evalin(’base’,’Kdp’);
Kdi=evalin(’base’,’Kdi’);
kd=evalin(’base’,’kd’);

% Actuator parameter
Wheatmax=evalin(’base’,’Wheatmax’);

% Sampling period
delta=evalin(’base’,’delta’);

%--------------------------------------------------------------------------
% Digital Pressurizer Pressure Controller 2
density_pck=c_phi0+c_phi1*T_pck-c_phi2*T_pck^2;
derdensity_pck=c_phi1-2*c_phi2*T_pck;
M_prk=M_pck-density_pck*V_pc0;
m_pr0k=m_ink-m_out0-derdensity_pck*V_pc0*(c_ppc*m_ink*(T_pci0-T_pck)
        +c_ppc*m_out0*Delta0+c_psi*Nk-n_sg*k_tsg*(T_pck-T_sgk)
        -W_losspc0)/(c_ppc*M_pck);
if m_pr0k>0,
    dprk=1;
else
    dprk=0;
end
Th_prk=kd*(c_pwall*T_prwallk/k_wall-xik);
ph_prk=c0-c1*Th_prk+c2*Th_prk^2;
Dhk=-c1+2*c2*Th_prk;
Cprk=c_ppr*M_prk*(dp_prrefk-Kdp*(ph_prk-p_prrefk)-Kdi*I_epprk)/Dhk;
xip=Th_prk-T_prwallk-W_losspr0/k_wall-Cprk/(kd*c_ppr*M_prk);
I_epprp=ph_prk-p_prrefk;
W_hk=k_wall*(Th_prk-T_prwallk)+Cprk+dprk*(c_ppr*m_pr0k*Th_prk
     -c_ppc*m_pr0k*(T_pck+Delta0));
if W_hk<0
    W_heatprk=0;
elseif W_hk>Wheatmax,
    W_heatprk=Wheatmax;
else
    W_heatprk=W_hk;
end
```

Table 3 – Digital pressurizer pressure control 2k (EMF code)

Numerical and analogical displays show the behavior of the controlled variables, make possible their check during the simulation, as long as further variables of interest, such as the tracking errors.

Switches allow the selection of the desired control law. They have various possible choices, for the various possible control laws (continuous–time, sampled, digital).

1.3 Simulations Results

In this section, the results obtained in the simulations of the model will be presented and discussed. The Simulink scheme, in Figure 1, allows varying several simulation conditions, such as the operation

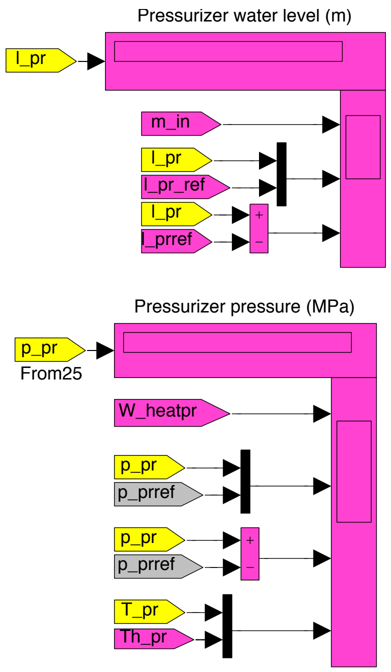


Figure 4: Displays for level and pressure in the pressurizer

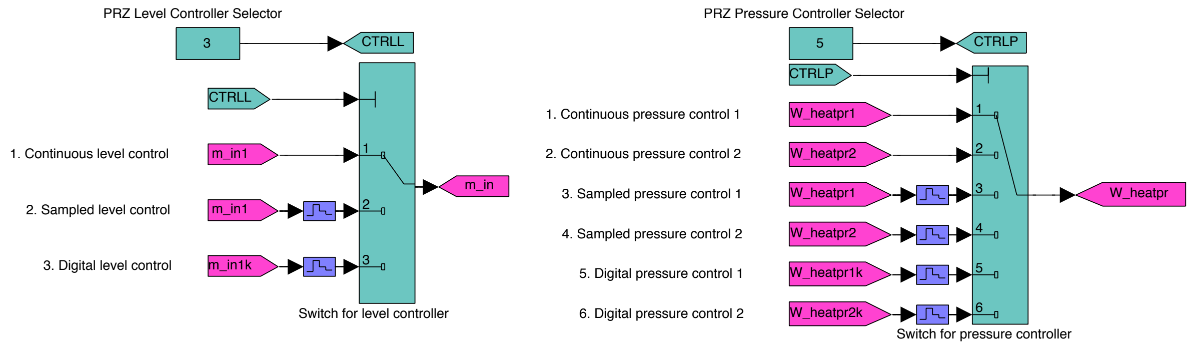

PRZ Level Controller Selector
3
CTRLL
CTRLL
1. Continuous level control
m_in1
2. Sampled level control
m_in1
3. Digital level control
m_in1k
1
2
3
m_in
Switch for level controller
PRZ Pressure Controller Selector
5
CTRLP
CTRLP
1. Continuous pressure control 1
W_heatpr1
2. Continuous pressure control 2
W_heatpr2
3. Sampled pressure control 1
W_heatpr1
4. Sampled pressure control 2
W_heatpr2
5. Digital pressure control 1
W_heatpr1k
6. Digital pressure control 2
W_heatpr2k
1
2
3
4
5
6
W_heatpr
Switch for pressure controller

Figure 5: Switches for control selection

condition of the plant (normal conditions, turbine trip transient), the pressurizer inventory and pressure controllers.

The parameters influencing the controller's performance, such as the sampling time for the sampled and the digital controllers, or those describing the turbine trip transient, can be changed modifying the values in the initialization data file (see Table 4).

For the sake of conciseness, we report here the most interesting case, namely that corresponding to the digital inventory control `Pressurizer_inventory_control_k` and the first digital pressure controller `Pressurizer_pressure_controller_1k` during a stop valve fault, and the consequent turbine trip transient. In fact, this event allows checking whether the digital control laws are robust with respect to faults. The simulation study have been conducted considering growing sampling times, as further test of robustness of the control laws with respect to delays. Indeed, the sampling time is one of the most important factor when dealing with implementations on digital electronics devices. The performance of these devices are growing quickly, but are still limited in speed. Finally, in the simulations parameter perturbations have been considered in order to achieve more realistic conditions.

We consider a turbine trip due to faulty closure of the turbine stop valve. During normal operation, the main steam flows from the steam generators through parallel pipes to a header, from where the steam is led to main steam stop and control valves by individual pipes of the high pressure turbine. Branch lines provide the possibility to bypass the turbine during transient operations. In normal conditions, the bypass station remains closed and the steam passes through the main stop and control valves and expands in the high pressure turbine.

The main steam stop valves have dual function. They isolate the turbine from the main steam line or from the steam generator. They rapidly interrupt the supply of steam to the turbine after being triggered by monitors if a dangerous condition arises. Therefore they have been designed for quick closing and maximum reliability. The control valves, on the other hand, regulate the flow of steam to the turbine according to the prevailing and provide a second means of isolation for the turbine in case of emergency. The control valve is operated by the piston of the servo–motor which is subjected to the spring force in the closing direction and the pressure of the control fluid in the opening direction. The position of the valve is determined by the secondary fluid pressure which is controlled by the governor.

In case of undue operating conditions within the turbine–generator plant the turbine trip system is released by means of protective devices for turbine and generator (turbine protection system). Hereby the main steam stop and control valves are closed. The steam produced in the steam generators is bypassed via bypass stop and control valves and dumped into the condenser. When turbine trip is initiated, the pressure drop in the trip oil circuit also causes the secondary fluid pressure to collapse because it is fed from it. The result is that both stop valve and the control valve close rapidly. The time for closing the stop valve is about 150 ms and for the control valve about 200 ms. In events like excessive load reductions, load rejection or turbine trip, the main steam maximum pressure limitation opens valves in the main steam bypass station and the main steam is passed into the condenser. The main steam pressure in the header is used as actual value for the control. The set–point is a few bars above the main steam operating pressure. Main steam relief station may also be used for controlling the main steam pressure.

Besides manual trip or spurious actuation, turbine trip initiation may be caused by steam turbine protection system components, like overspeed protection, overspeed trip selection, high condenser pressure protection, thrust bearing trip, low lube oil pressure trip, fire protection, main steam minimum pressure signalin, electrical or mechanical generator protection.

After the turbine stop valves have closed, main steam pressure increases challenging the steam generator secondary side heat removal capability. Coolant temperature and pressure will increase and, unless adequate countermeasures are timely provided, heat removal from the core may be challenged too.

On turbine trip, after the turbine stop valves have closed, secondary side heat removal is abruptly interrupted leading to a sudden increase in the main steam pressure. However, the pressure excursion is rather limited by the prompt response of the turbine bypass station. The main steam bypass valves open immediately because the MS pressure goes above the maximum pressure set–point, which is reduced on turbine trip and then raised to a maximum at which it is held. In the first 10 s after the turbine trip, degraded heat removal conditions in the secondary side of the steam generators cause an increase in the coolant average temperature and a consequent expansion of coolant volume. Because of the high reactor minimum load, the overall temperature changes in coolant and moderator are rather small. Consequently, the volumetric changes are also limited, as can be seen in the behavior of the pressurizer water level. Closed–loop control and limitation systems are called upon to deal with the effects of power mismatch between reactor and the electric generator, keeping process variables within acceptable limits. Partial rod dropping is initiated by comparing reactor and generator power and, after 1.1 s delay time, the reactor is run back to a minimum load of approximately 80%, as a consequence of rod movement. A minor transient is observed in the coolant pressure which demands the intervention of pressurizer heating power. With the available main steam bypass system, the secondary relief station stays closed. Due to the main steam pressure rise, the steam generator water level initially slightly decreases and it is brought back to normal by the main feed–water control. Due to the effective response of control and limitation systems, promptly reducing the reactor power and early opening of the main steam bypass station, the process variables differ only insignificantly from their set–points, keeping reasonable margins to the limits of the reactor protection system, which are not reached.

Figures 6–36 show the dynamics associated to the principal variables describing the primary circuit dynamics, the reference values of the controlled variables and output, as well as the controlled input. As already commented, the simulations have been carried out for different values of the sampling time: $\delta = 0.0001$ s, $\delta = 0.001$ s, $\delta = 0.01$ s, $\delta = 0.1$ s, and $\delta = 1$ s. The simulation results that follow will show the satisfactory behavior, although for $\delta = 1$ the control input starts to be too active.

Table 4 – Initialization data file

```
clear all, clc
disp('Loading simulation data ...')
disp('(see help for details)')
disp(' ')

%=========================================================================
% Reactor parameters
%-------------------------------------------------------------------------
Lambda=1e-5;         % generation time; s
S=2830.05;           % flux of the constant neutron source; %/s
p0=2.85e-4;          % rod reactivity coefficients; m
p1=6.08e-5;          %                              m^(-1)
p2=1.322e-4;         %                              m^(-2)

%-------------------------------------------------------------------------
% Primary Ciruit parameters
%---------------------------
c_ppc=5355;          % specific heat at 280 C; J/(kg*K)
c_psi=13.75e6;       % power reactor constant; W/%
n_sg=6;              % number of steam generatots in Paks Nuclear Power Plant

% Perturbations
m_out0=2.11;         % nominal outlet mass flow rate; kg/s
m_out=2.0678;        % real outlet mass flow rate: -2% of m_out0; kg/s
T_pci0=258.85;       % nominal inlet temperature; C
T_pci=256.2615;      % real inlet temperature: -1% of T_pci0; C
W_losspc0=2.996e7;   % nominal heat loss; J/s
W_losspc=3.07976e7;  % real heat loss: +3% of W_losspc0; J/s
Delta0=15;           % nominal difference between T_pc and T_pc,cl; C
Delta=15.6;          % real difference between T_pc and T_pc,cl: +4% of Delta0; C

%-------------------------------------------------------------------------
% Steam Generator parameters
%---------------------------
m_sg=119.31;         % inlet secondary water mass flow rate = outlet secondary
```

```
                                                   steam mass flow rate; kg
c_psgl=3809.9;       % second. circuit liquid water specific heat at 260 C;
                                                                   J/(kg K)
c_psgv=3635.6;       % second. circuit  steam water specific heat at 260 C;
                                                                   J/(kg K)
T_sgsw= 220.85;      % second. circuit inlet temperature; C
E_evapsg=1.658e6;    % evaporation energy at 260 C; J/kg
k_tsg=9.5296e6;      % steam generator heat transfer coefficient; J/(K s)
M_sg=34920;          % water mass; kg

% Perturbations
W_losssg0=1.8932e7;  % nominal heat loss; J/s
W_losssg=1.9689e7;   % real heat loss: +4% of W_losssg0; J/s

%--------------------------------------------------------------------------
% Pressurizer parameters
%----------------------------
c_ppr=6873.1;        % specific heat of the water; J/(kg*K)
V_pc0=242;           % water nominal volume; m^3
c_phi0=581.2;        % coefficients of the density quadratic function; []
c_phi1=2.98;
c_phi2=0.00848;
c0=28884.78;         % coefficients of the saturated vapor; kPa
c1=258.01;           %                                      kPa/C
c2=0.63455;          %                                      kPa/C^2
A_pr=4.52;           % vessel cross section; m^2
k_wall=1.9267e8;     % wall heat transfer coefficient; W/C
c_pwall=6.4516e7;    % wall heat capacity; J/C

% Perturbations
W_losspr0=1.6823e5;  % nominal heat loss; J/s
W_losspr=1.7159e5;   % real heat loss: +2% of W_losspr; J/s
```

```
%-------------------------------------------------------------------------
% Nominal inputs
%--------------------------
v0=0;               % input: nominal rod position; cm
m_in0=2.11;         % input: nominal inlet mass flow rate; kg/s
W_heatpr0=168000;   % input: nominal heating power; W

%-------------------------------------------------------------------------
% Steady state conditions
%--------------------------
% Reactor initial Condition
N0=Lambda*S/p0;     % =99.3 The neutron flux N is measured in percent

% Primary Ciruit initial conditions
M_pc0=2e5;          % water mass in the primary circuit; kg

% Primary circuit and Steam generator initial conditions
A=[c_ppc*m_in0+n_sg*k_tsg   -n_sg*k_tsg;
    -k_tsg             m_sg*c_psgv+k_tsg];
B=[c_ppc*m_in0*T_pci+c_ppc*m_out*Delta+c_psi*N0-W_losspc;
   m_sg*(c_psgl*T_sgsw-E_evapsg)-W_losssg];
C=inv(A)*B;
T_pc0=C(1,1);
T_sg0=C(2,1);
clear A B C

% Pressurizer initial conditions
T_pr0=326.51;                     % C
T_prwall0=T_pr0-W_losspr/k_wall;

%-------------------------------------------------------------------------
% Pressurizer water level reference parameters
%--------------------------
```

```
% pressurizer water level at nominal conditions
l_pr0=(M_pc0/(c_phi0+c_phi1*T_pc0-c_phi2*T_pc0^2)-V_pc0)/A_pr;

c_r1=0.093;                       % m/C
c_r2=2*c_r1*T_pc0-l_pr0;          % m

%-------------------------------------------------------------------------
% Turbine trip transient (TTT)
%---------------------------
tTTT=2;               % instant of occurrence of the turbine trip transient; s
DeltaTTTN=1.1;        % delay for reducing reactor power; s

a=p2;                % determination of rod position correspondig to
b=p1;                % N= 80% of N0
c=p0-Lambda*S/(N0*0.80);
vTTT=(-b+sqrt(b^2-4*a*c))/(2*a);
clear a b c

%-------------------------------------------------------------------------
% Pressurizer water level controller parameters
%---------------------------
k_p=100;
k_i=50;

% Initial condition (integral action)
I_elpr0=0;

% Actuator parameter (saturation)
minmax=20;                               % kg/s

%-------------------------------------------------------------------------
% Pressurizer pressure controller
%---------------------------
```

```
% Pressure reference and derivative
p_prref=12300;                          % kPa
dp_prref=0;

% Actuator parameter (saturation)
Wheatmax=3.6e6;                         % W

% Temperature observer initial conditions
Th_pr0=324;                             % C

%-----------------------------------------------------------------------------
% Continuous controller 1
%--------------------------
% Initial conditions
T_prref0=326.51;                        % C
T_prwallref0=T_prref0-W_losspr0/k_wall;  % C
I_eTprwall0=0;                          % C s (integral action)

% Controller parameter
k_i1=200;                               % Integral gain

% Lyapunov matrix equation
q1=1e-5;
q2=1e-10;
Q=[q1 0; 0 q2];
A=[0 1; -k_i1 -k_wall/c_pwall];
P=lyap(A',Q);  % P=lyap(A',Q) solves the Lyapunov matrix equation: P*A + A'*P = -Q
clear A Q

%-----------------------------------------------------------------------------
% Continuous controller 2
%--------------------------
% Parameters
```

```
zeta=0.707;                              % damping
wn=3e3;                                  % natural frequency
Kp=2*zeta*wn;
Ki=wn^2;

% Observer gain
k=20;

% Integrator initial conditions
xi0=-Th_pr0/k+c_pwall*T_prwall0/k_wall;
I_eppr0=0;

%-------------------------------------------------------------------------
% Digital Controllers
%---------------------------
% Sampling time
delta=.1;                                % s

%-------------------------------------------------------------------------
% Digital controller 1
%---------------------------
% Initial conditions as for the continuous controller

% Controller parameter
k_di1=k_i1;                             % Integral gain

% Lyapunov matrix equation for the digital case
qd1=q1;
qd2=q2;
Qd=[qd1 0; 0 qd2];
Ad=[0 1; -k_di1 -k_wall/c_pwall];
Pd=lyap(Ad',Qd);  % Pd=lyap(Ad',Qd) solves the Lyapunov matrix equation:
                                                    Pd*Ad + Ad'*Pd = -Qd
```

```
clear Ad Qd

%-------------------------------------------------------------------------
% Digital controller 2
%---------------------------
% Initial conditions as for the continuous controller

% Parameters
zetad=0.707;                            % damping
wnd=50;                                 % natural frequency
Kdp=2*zetad*wnd;
Kdi=wnd^2;

% Observer gain
kd=20;

%=========================================================================
disp('... data loaded!')
disp('Starting simulation.')
```

Table 4 – Initialization data file

1.3.1 Digital Inventory `Pressurizer_inventory_controller_1k` and Pressure Control `Pressurizer_pressure_controller_1k` with Sampling Time $\delta = 10^{-3}$ s

The simulation results are summarized in Figures 6–12.

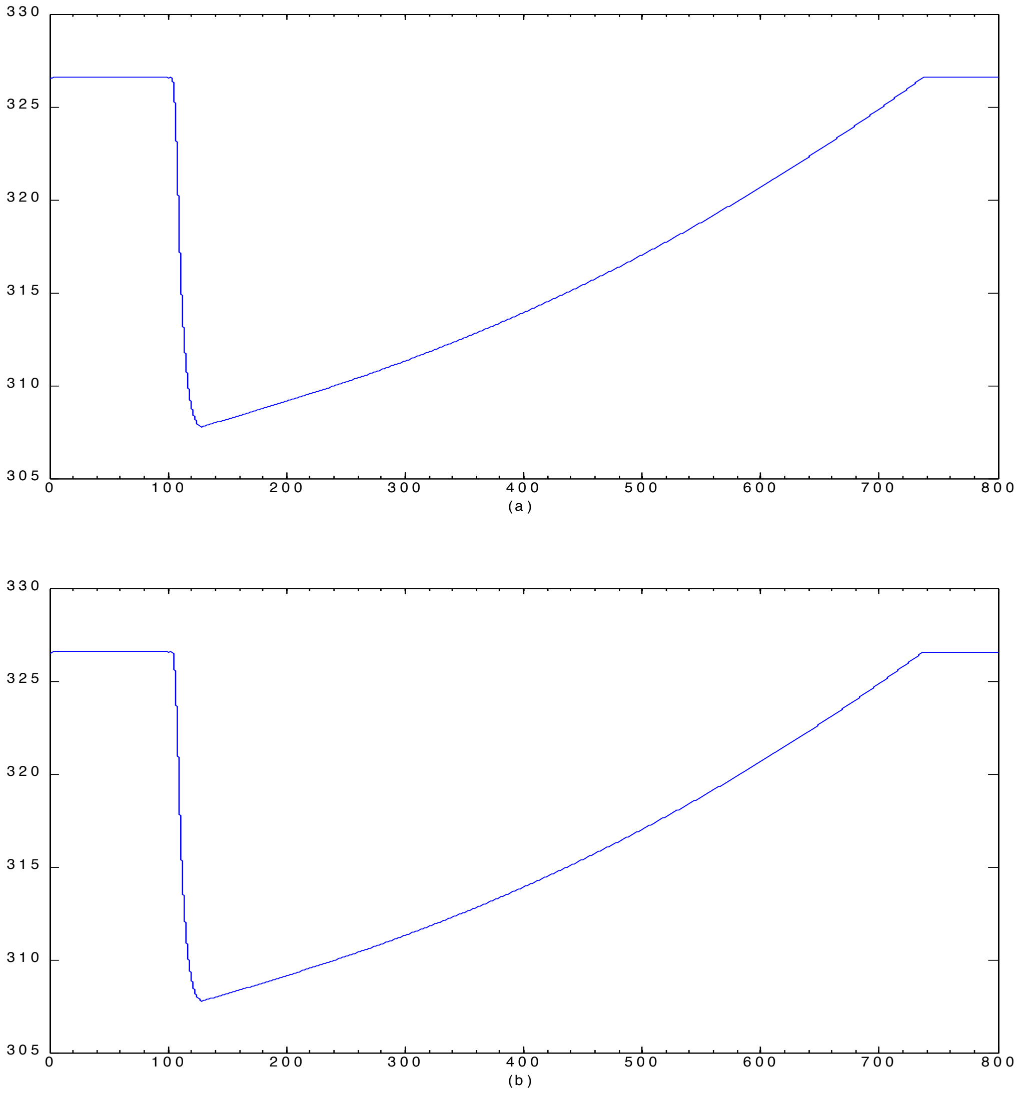


Figure 6: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Pressurizer temperature [°C]; (b) Pressurizer wall temperature [°C]

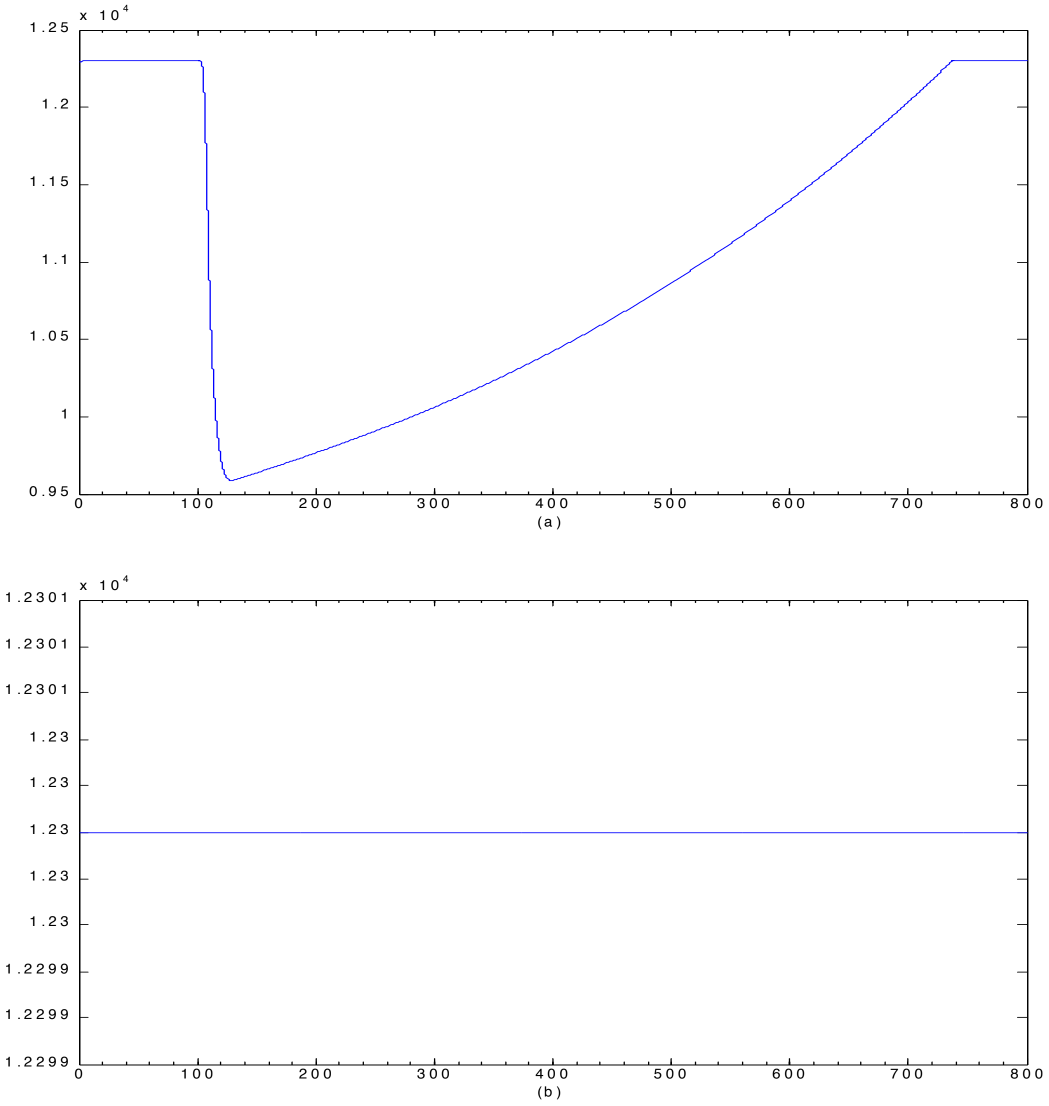


Figure 7: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Pressurizer pressure [Pa]; (b) Pressurizer pressure reference [Pa]

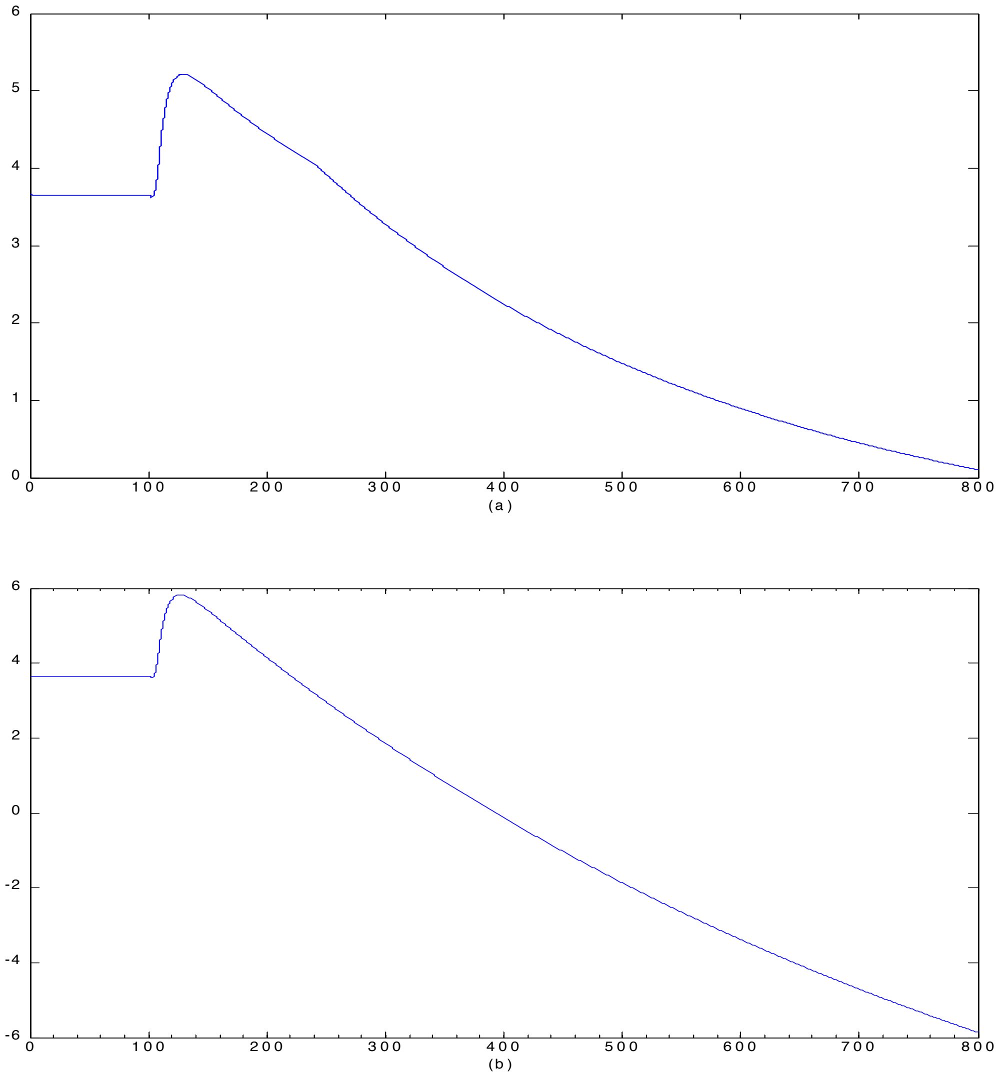


Figure 8: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Pressurizer water level [m]; (b) Pressurizer water level reference [m]

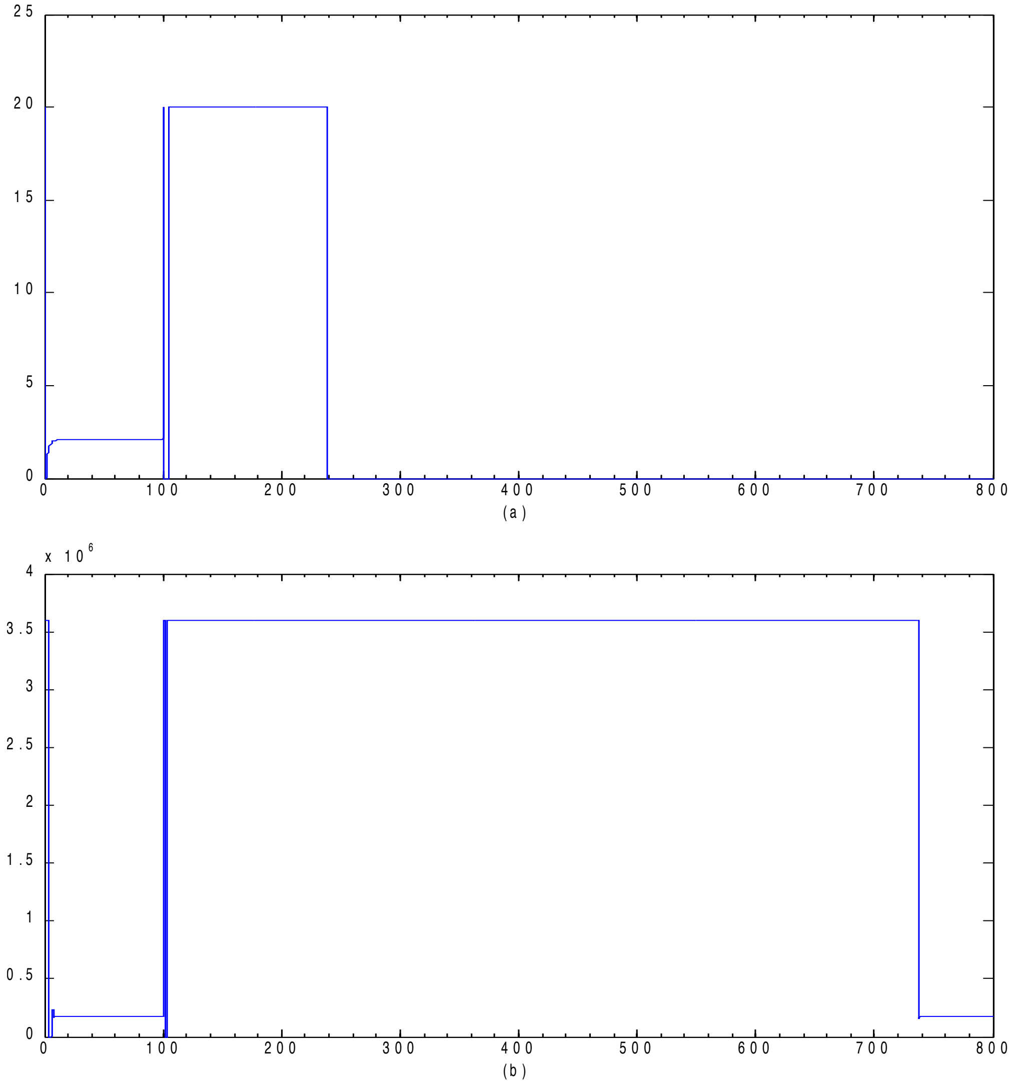


Figure 9: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Inlet mass flow rate [kg/s]; (b) Pressurizer heating power [W]

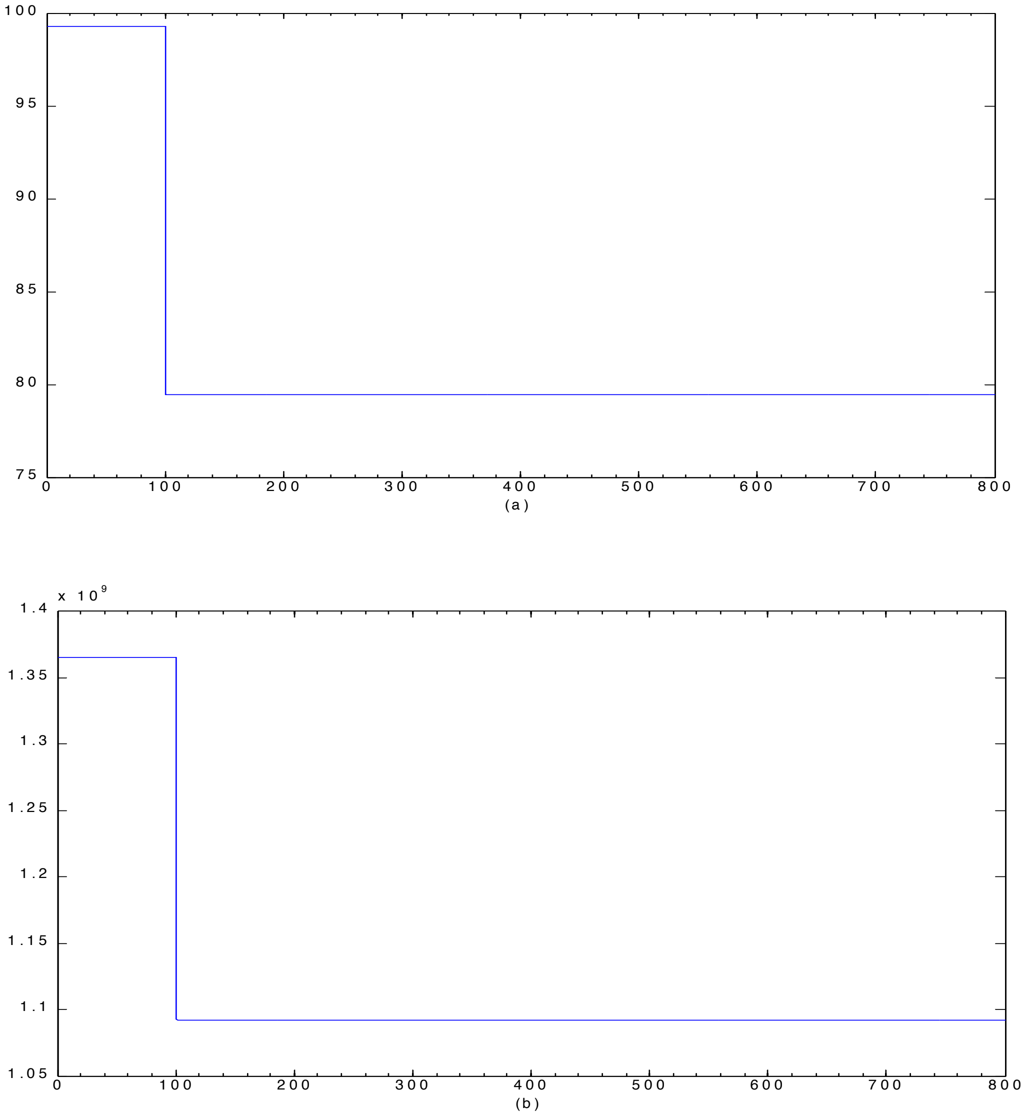


Figure 10: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Neutron flux [%]; (b) Reactor power [W]

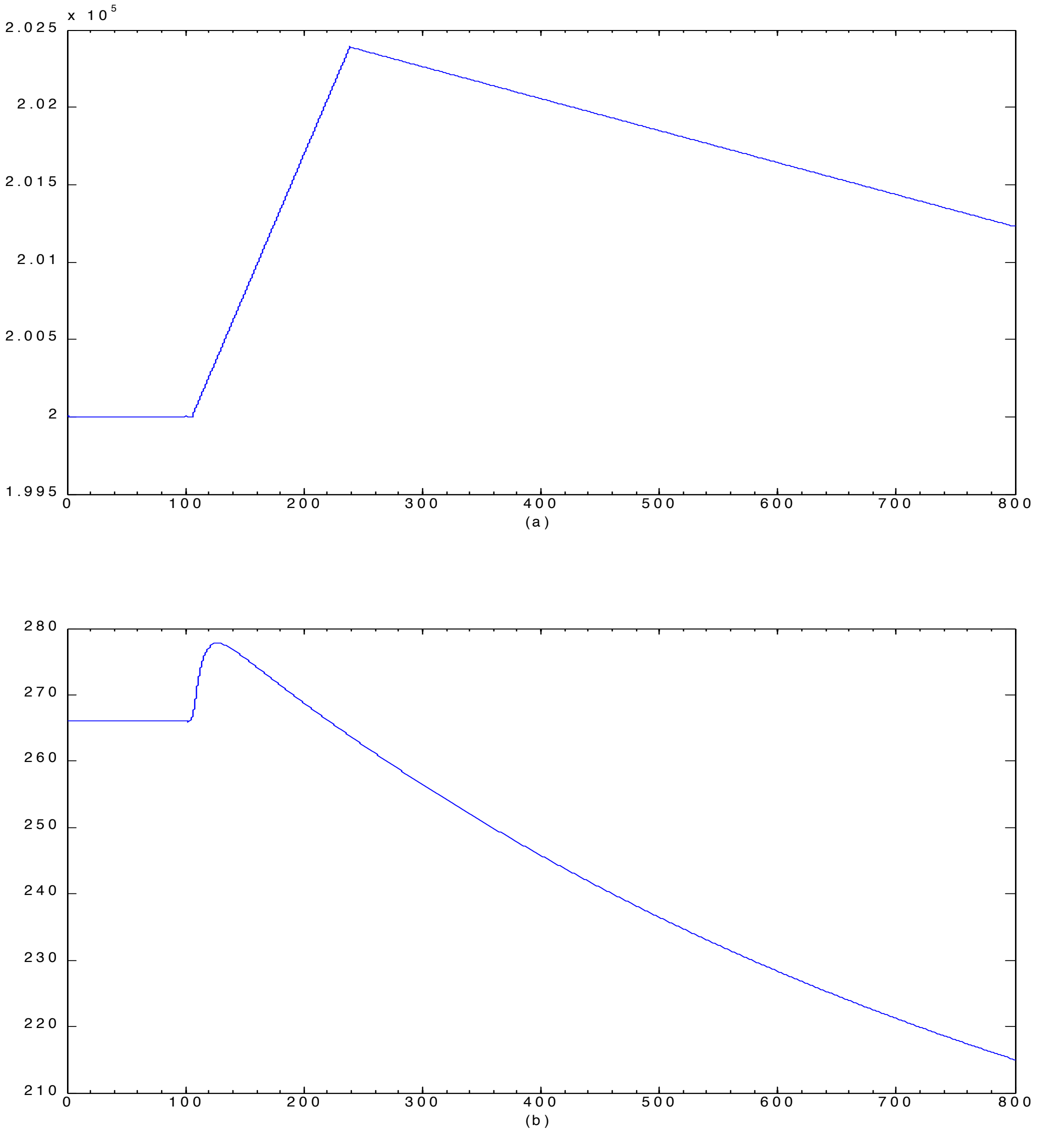


Figure 11: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Primary circuit water mass [kg]; (b) Primary circuit avarege temperature [°C]

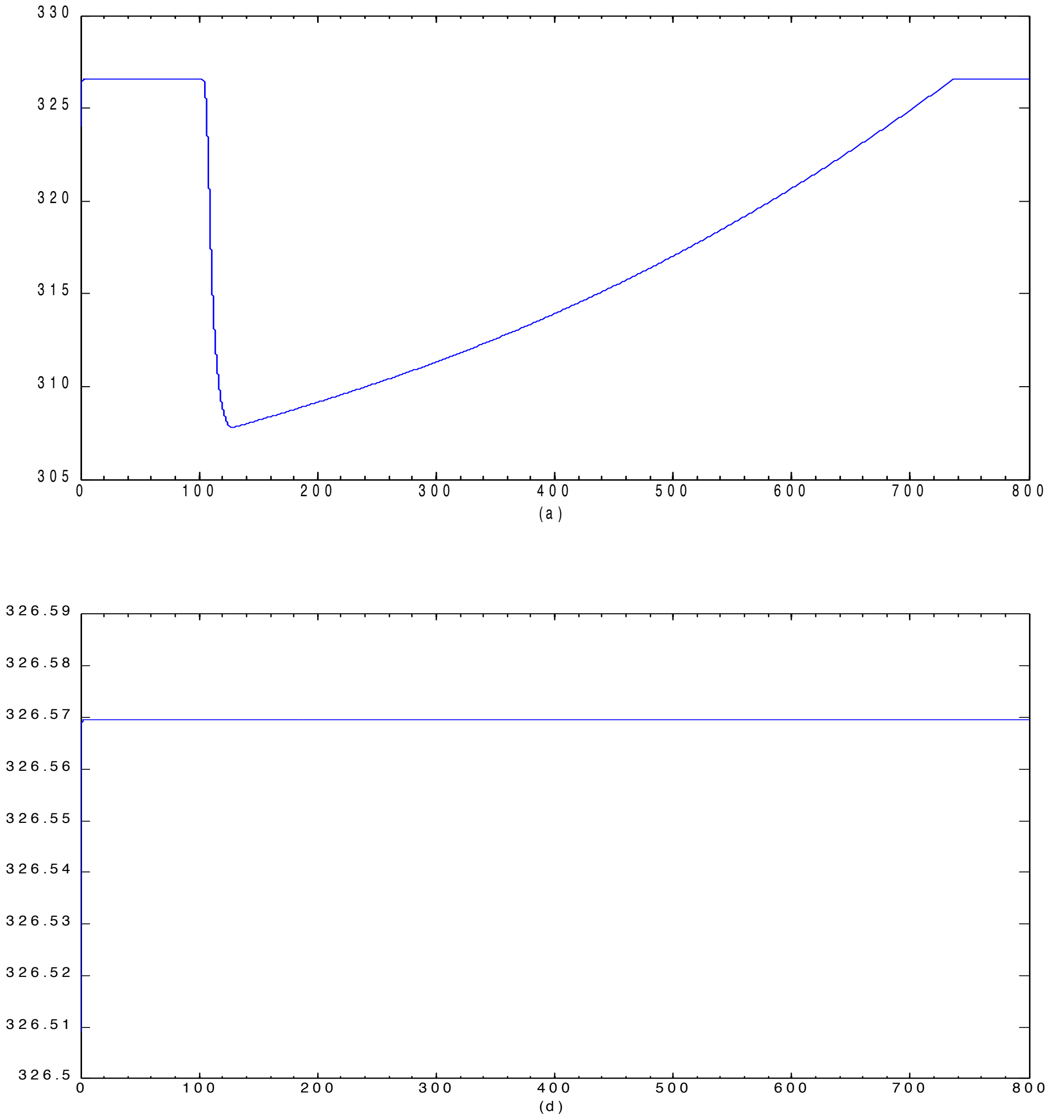


Figure 12: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-3}$ s. (a) Estimated pressurizer temperature [°C]; (b) Pressurizer reference temperature [°C]

1.3.2 Digital Inventory Pressurizer_inventory_controller_1k and Pressure Control Pressurizer_pressure_controller_1k with Sampling Time $\delta = 10^{-2}$ s

The simulation results are summarized in Figures 13–19.

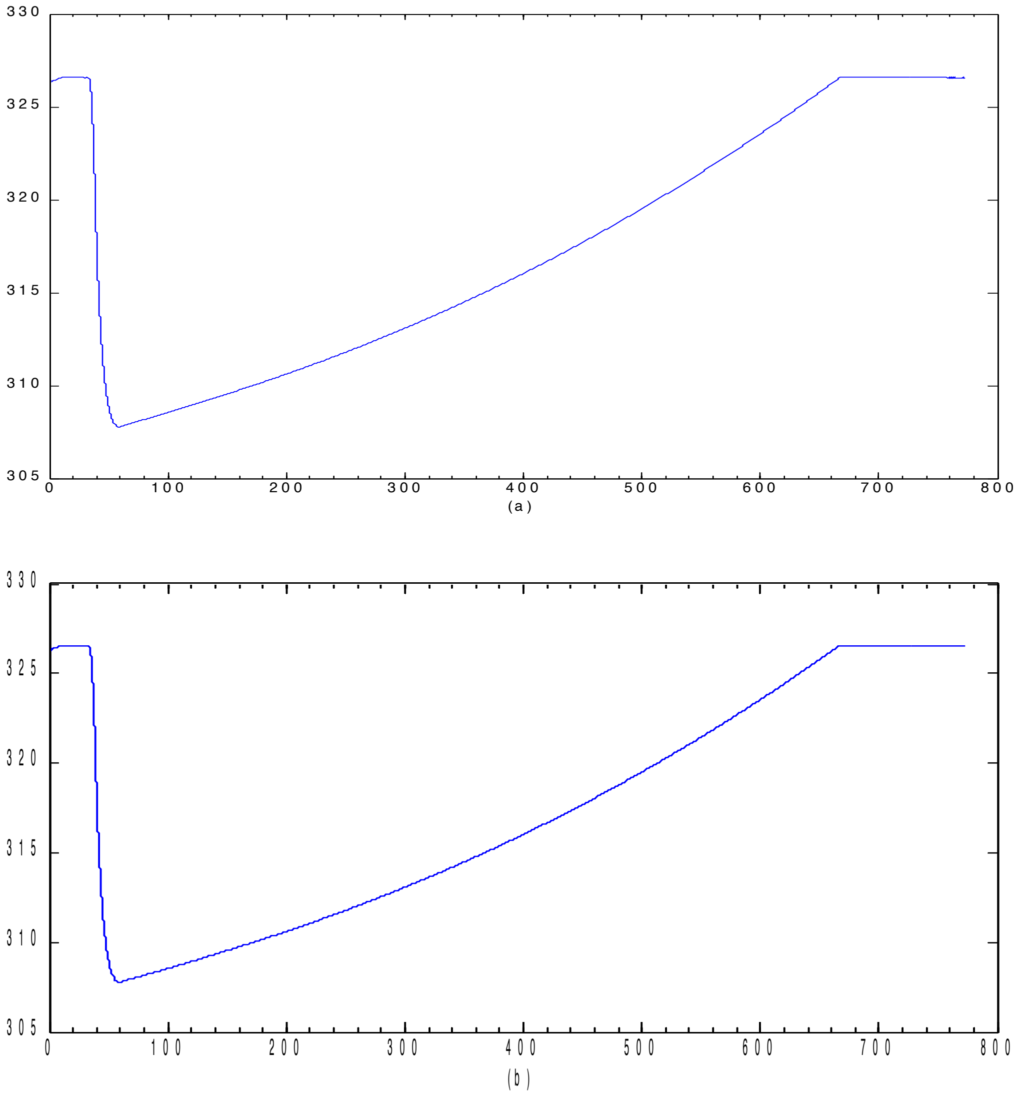


Figure 13: Digital inventory and pressure control Pressurizer_inventory_controller_1k, Pressurizer_pressure_controller_1k with sampling time $\delta = 10^{-2}$ s. (a) Pressurizer temperature [°C]; (b) Pressurizer wall temperature [°C]

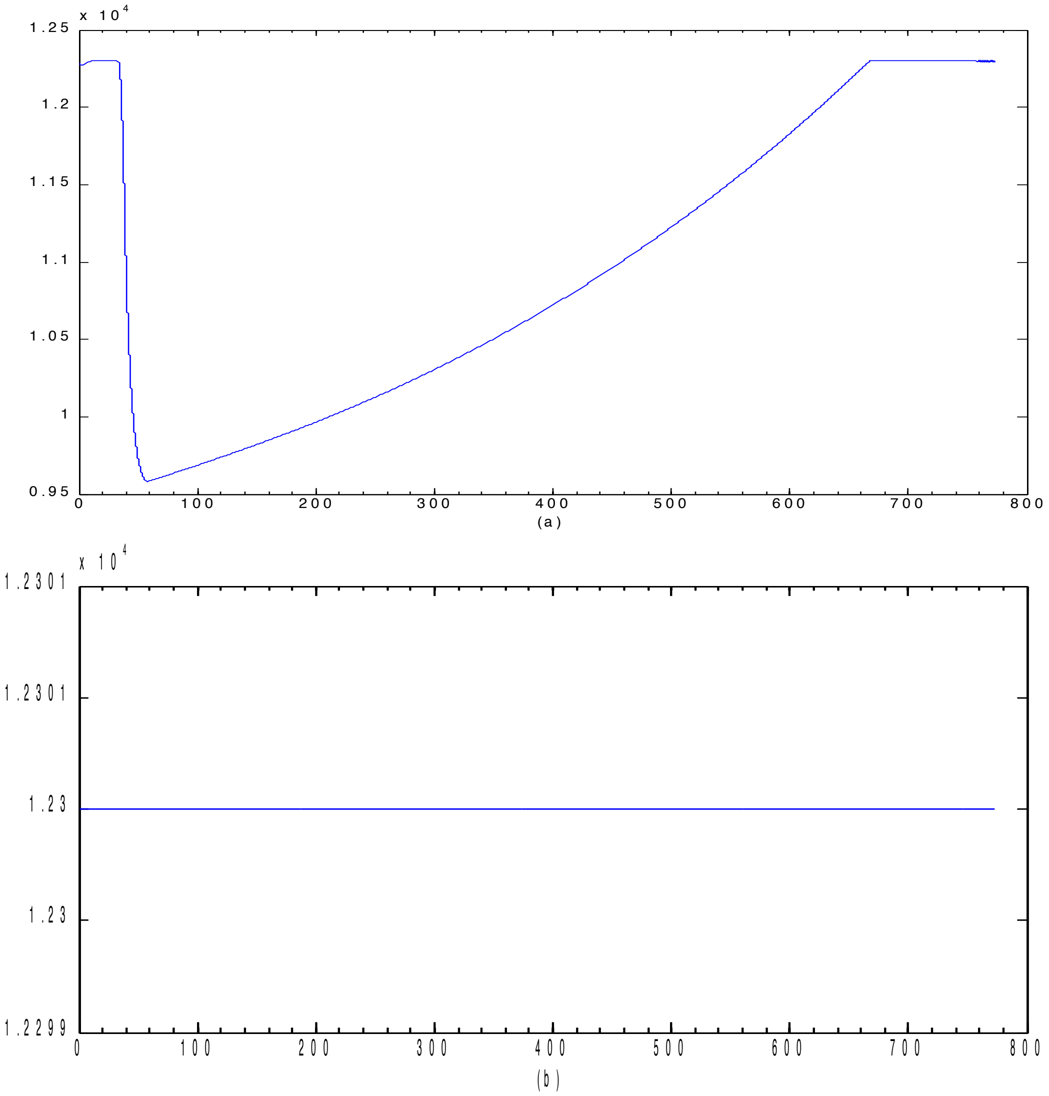


Figure 14: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-2}$ s. (a) Pressurizer pressure [Pa]; (b) Pressurizer pressure reference [Pa]

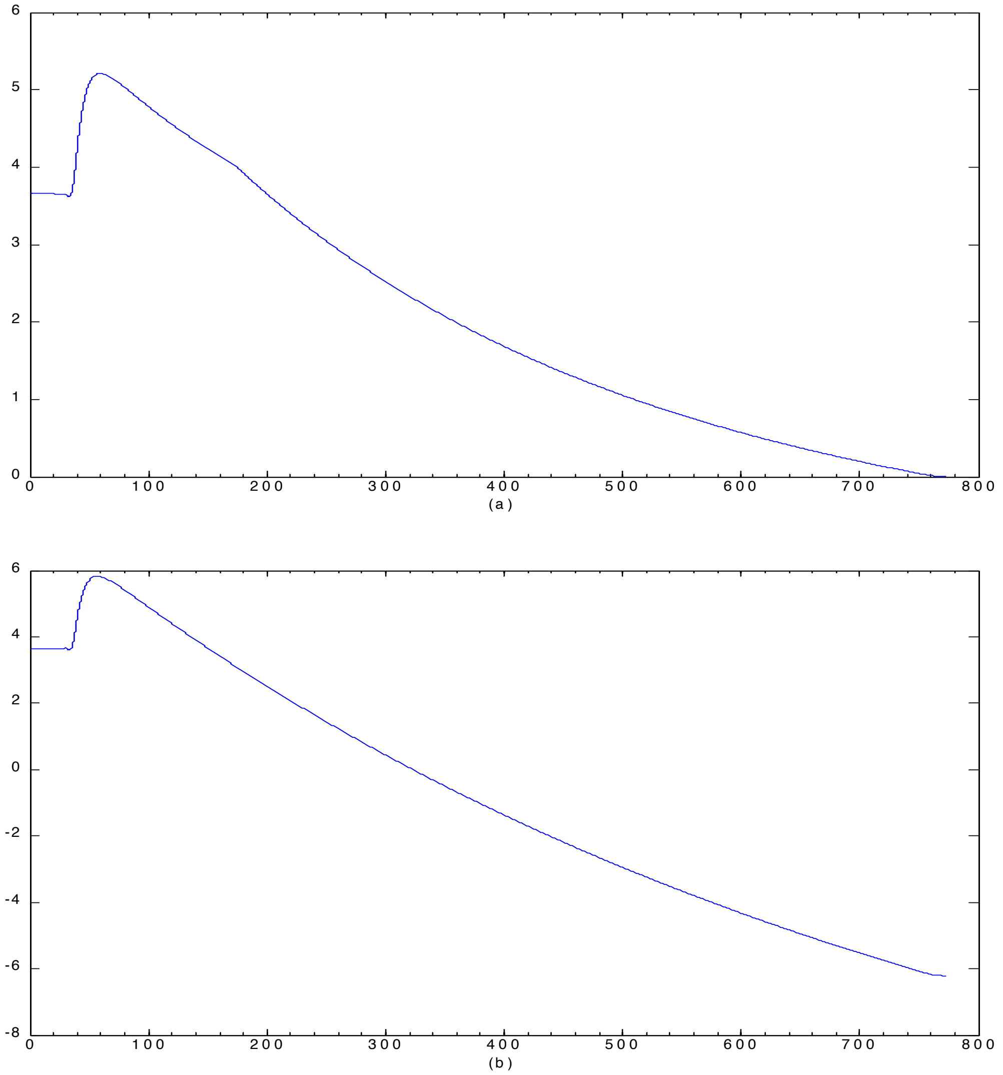


Figure 15: Controllers (6)-(7), $\varDelta_t = 0.01$. (a) Pressurizer water level [m]; (b) Pressurizer water level reference [m]

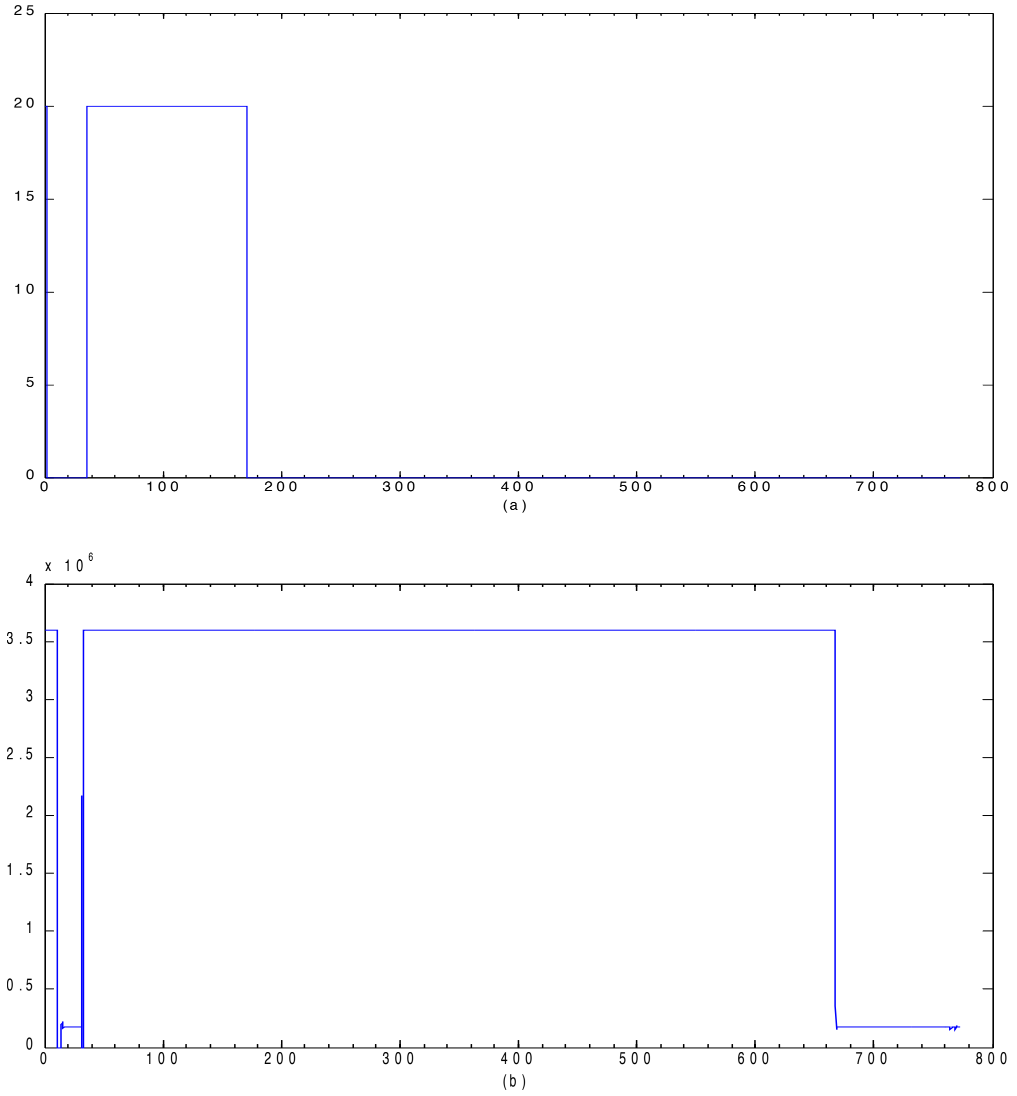


Figure 16: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-2}$ s. (a) Inlet mass flow rate [kg/s]; (b) Pressurizer heating power [W]

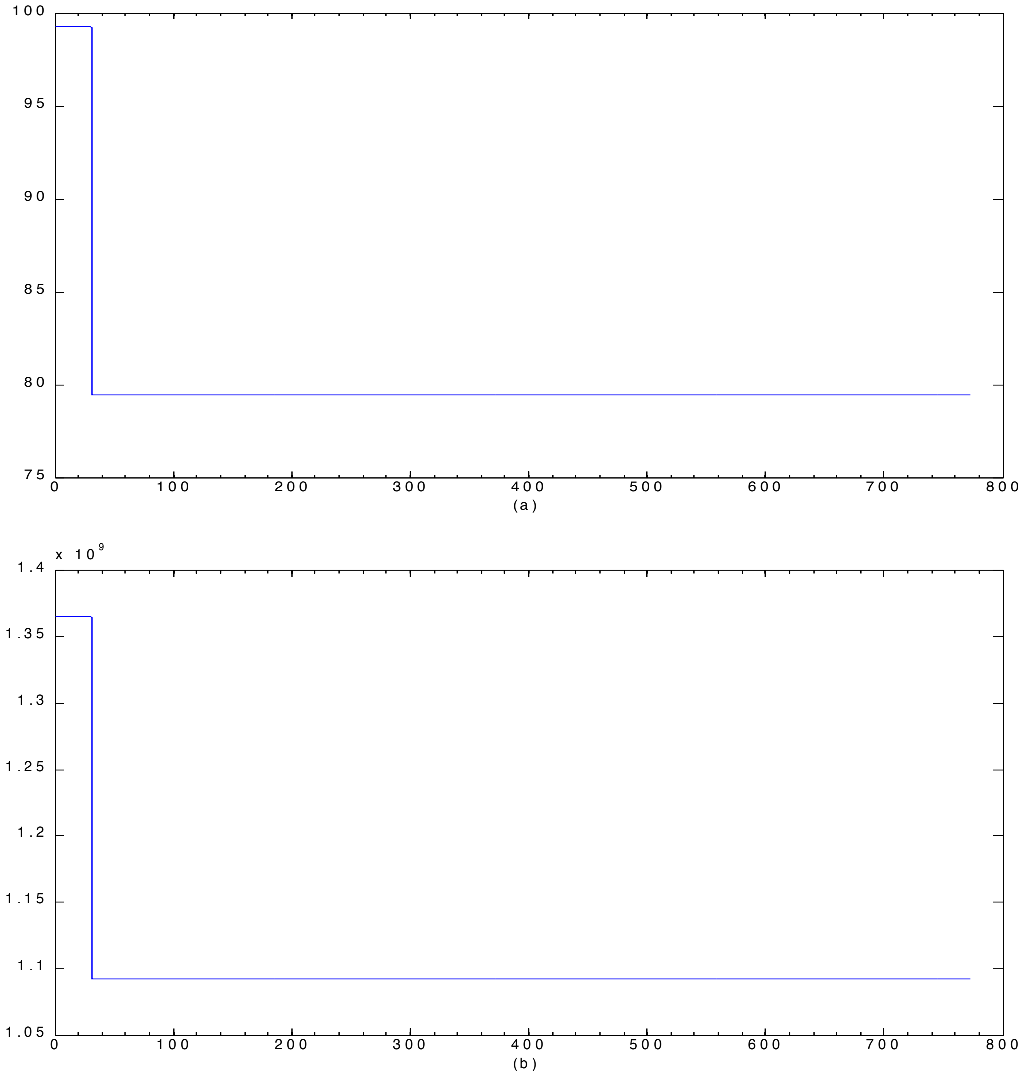


Figure 17: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-2}$ s. (a) Neutron flux [%]; (b) Reactor power [W]

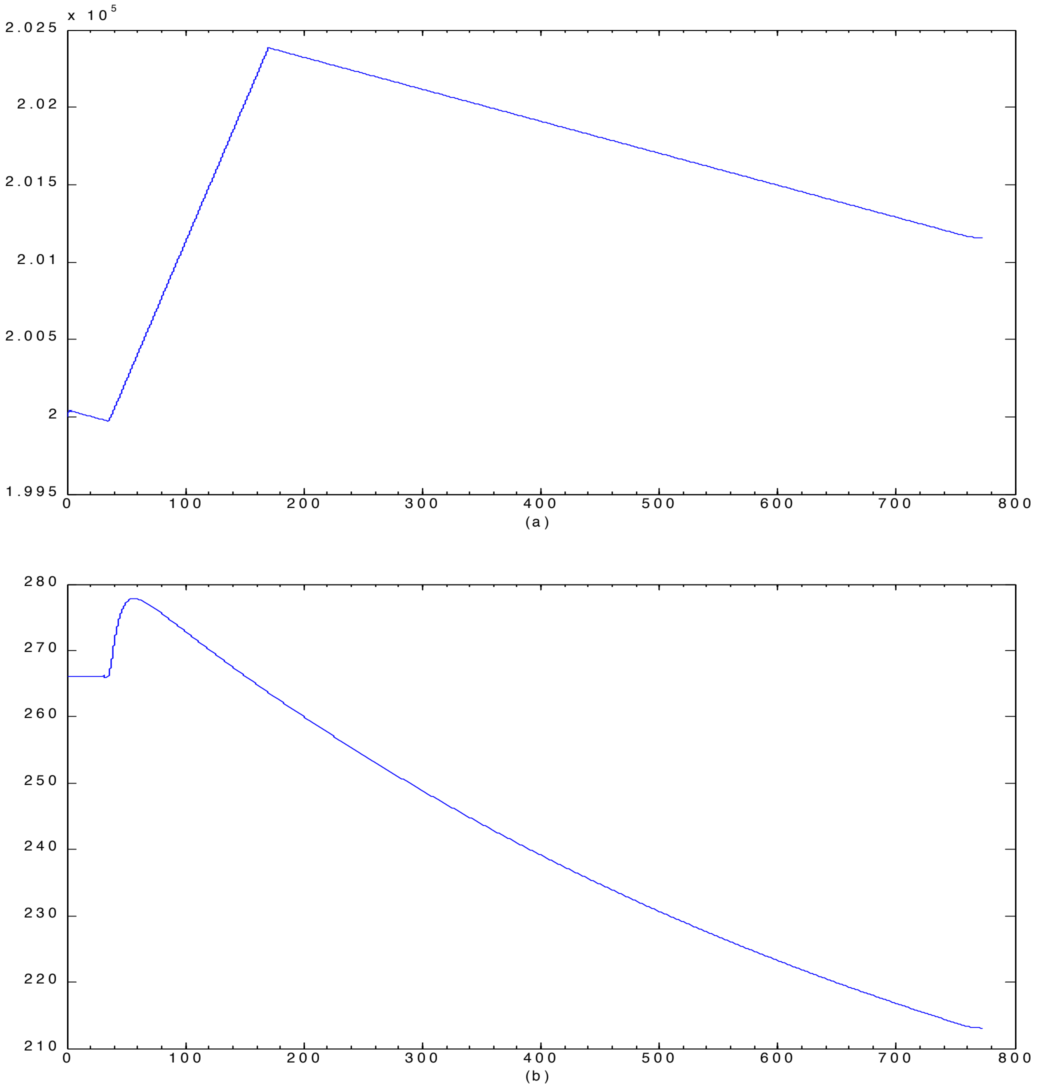


Figure 18: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-2}$ s. (a) Primary circuit water mass [kg]; (b) Primary circuit avarege temperature [°C]

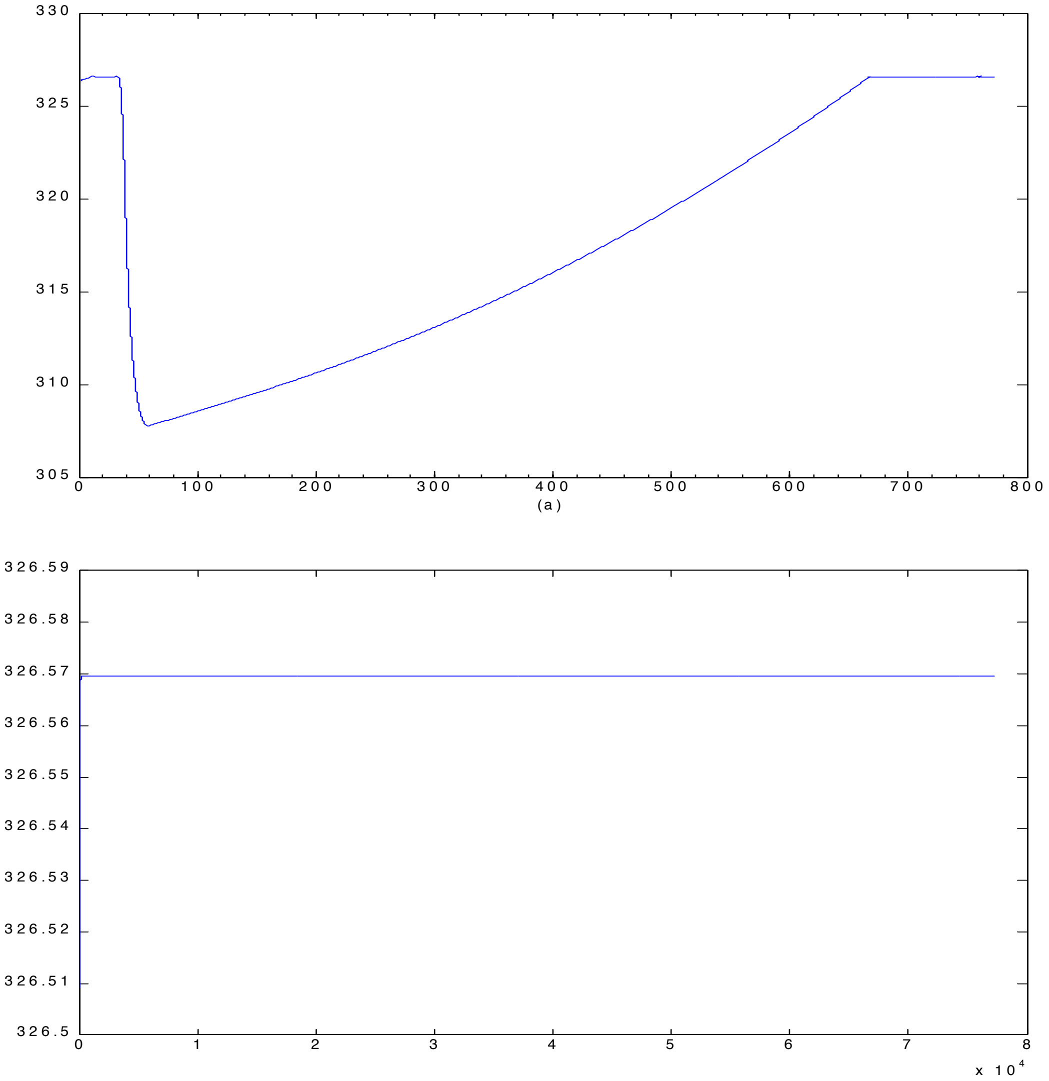


Figure 19: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-2}$ s. (a) Estimated pressurizer temperature [°C]; (b) Pressurizer reference temperature [°C]

1.3.3 Digital Inventory `Pressurizer_inventory_controller_1k` *and Pressure Control* `Pressurizer_pressure_controller_1k` *with Sampling Time* $\delta = 10^{-1}$ *s*

The simulation results are summarized in Figures 20–26.

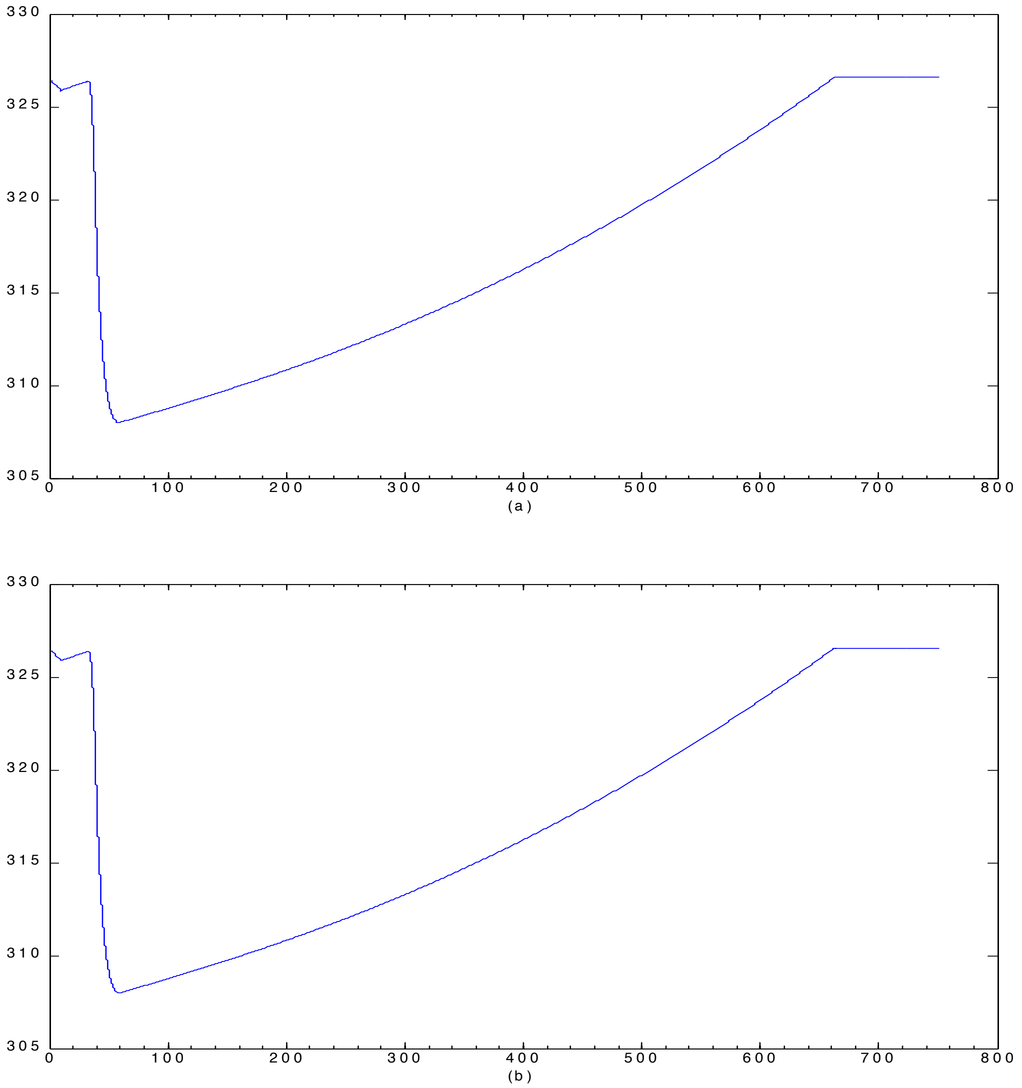


Figure 20: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Pressurizer temperature [°C]; (b) Pressurizer wall temperature [°C]

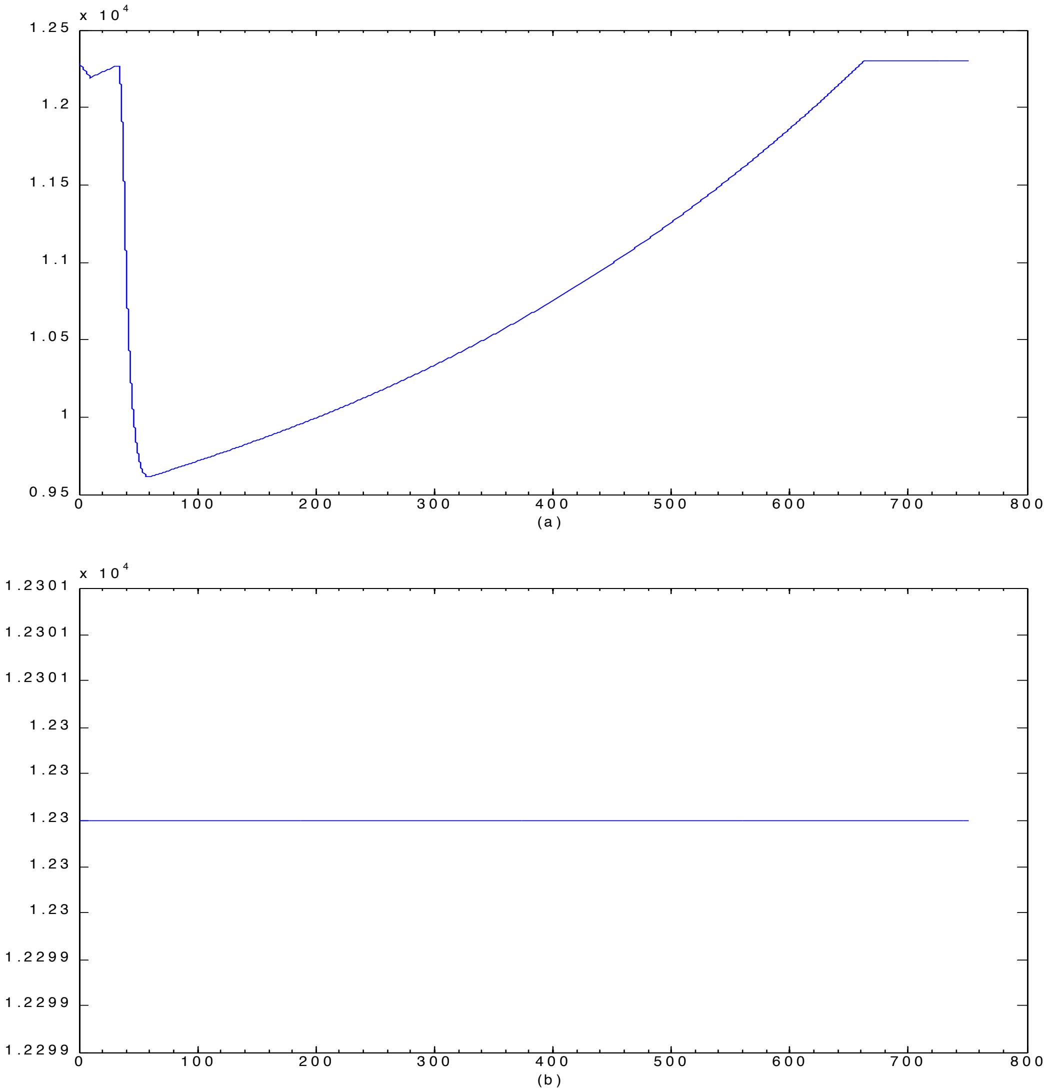


Figure 21: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Pressurizer pressure [Pa]; (b) Pressurizer pressure reference [Pa]

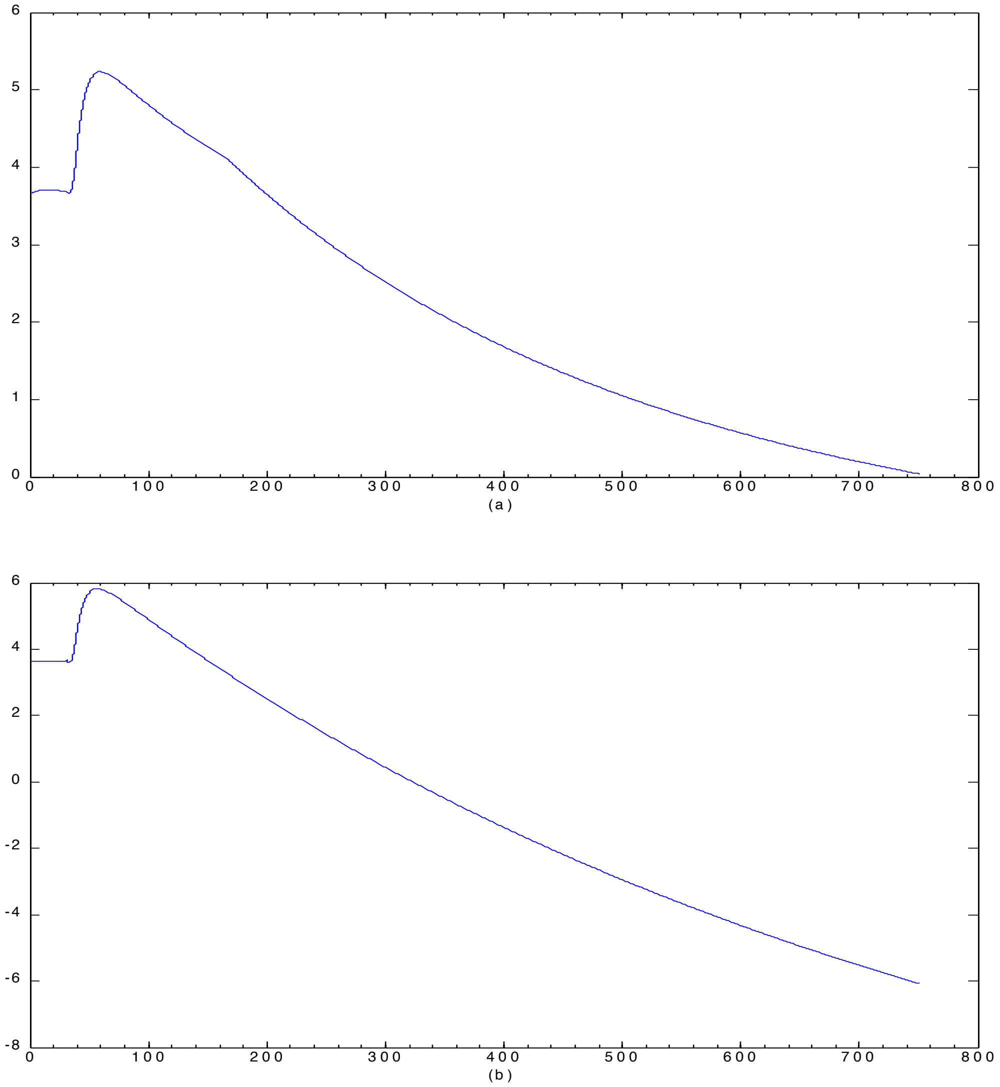


Figure 22: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Pressurizer water level [m]; (b) Pressurizer water level reference [m]

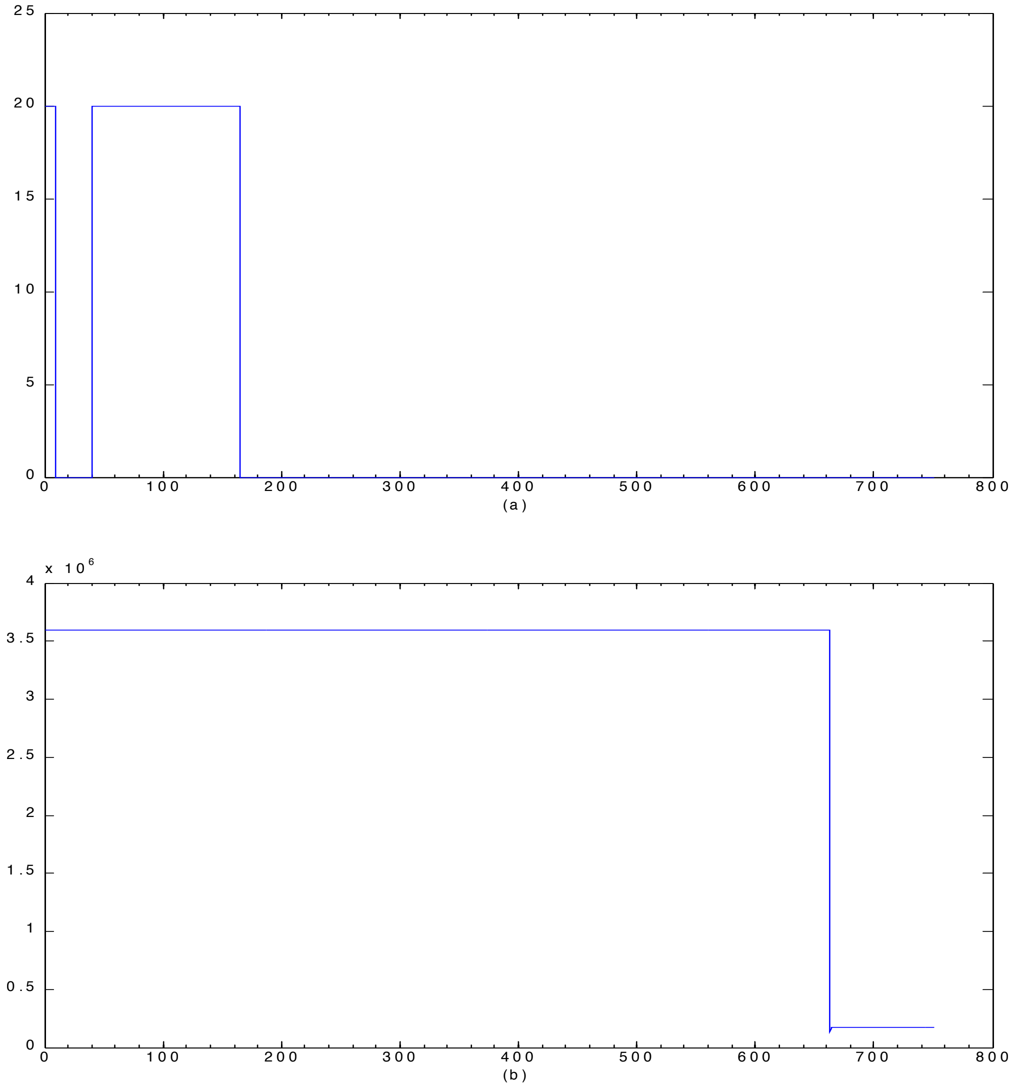


Figure 23: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Inlet mass flow rate [kg/s]; (b) Pressurizer heating power [W]

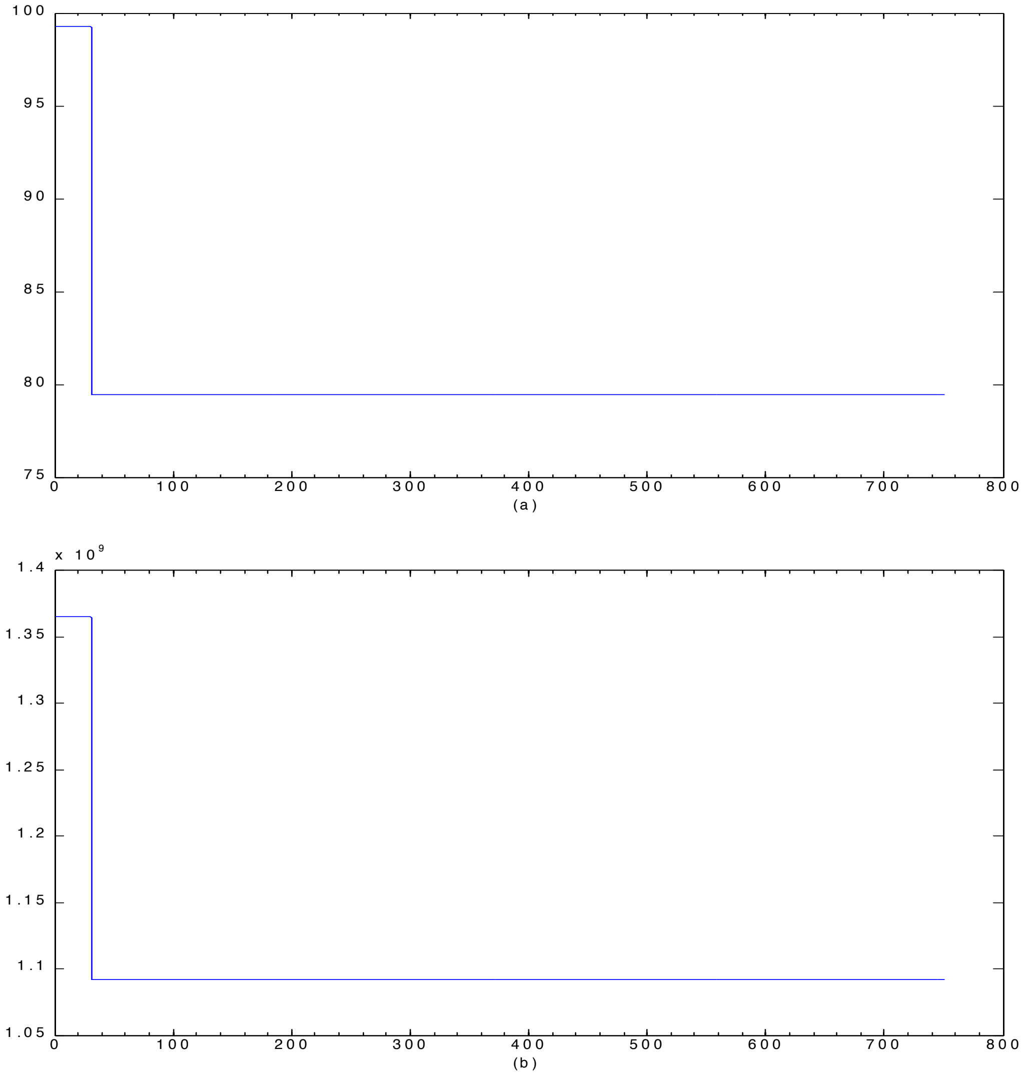


Figure 24: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Neutron flux [%]; (b) Reactor power [W]

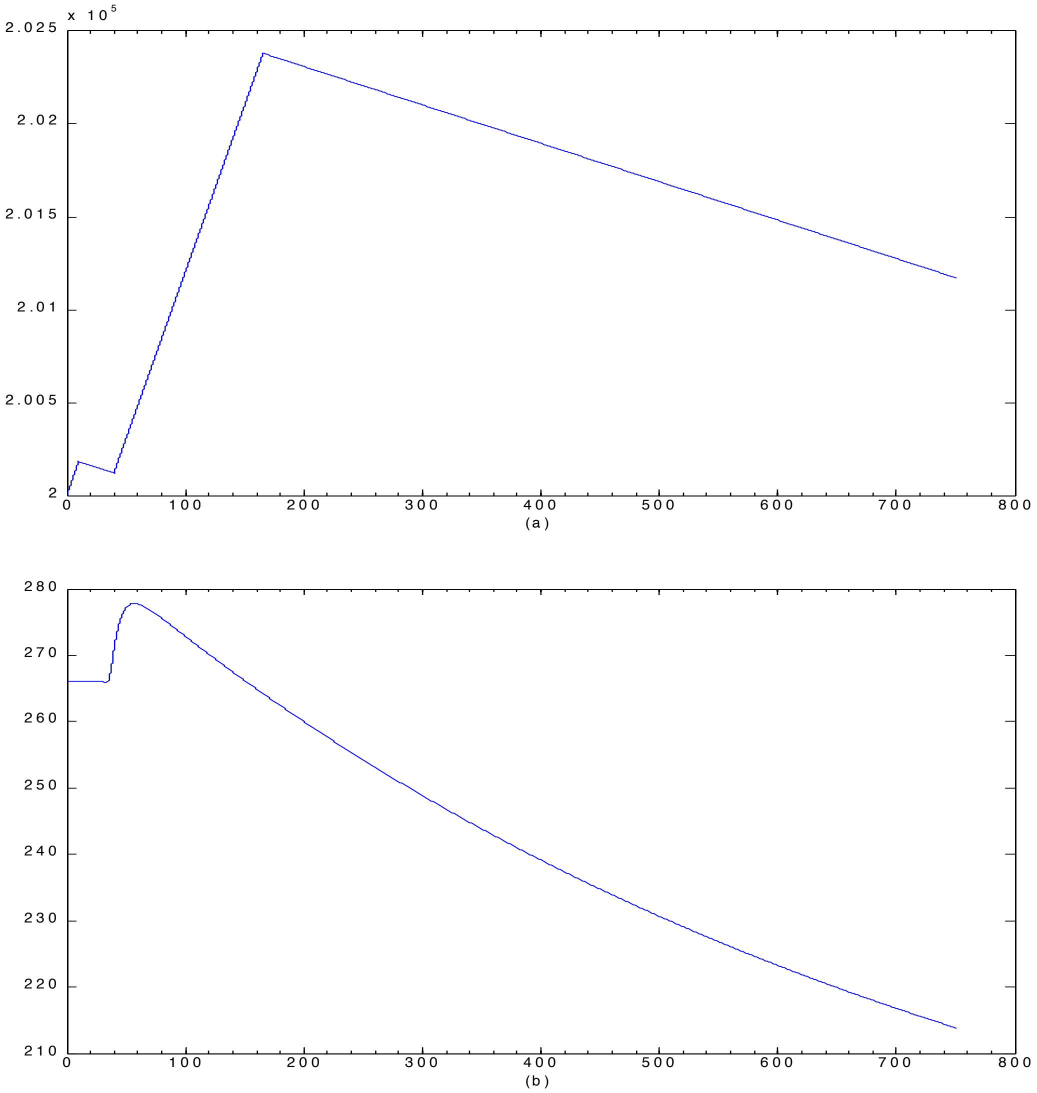


Figure 25: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Primary circuit water mass [kg]; (b) Primary circuit avarege temperature [°C]

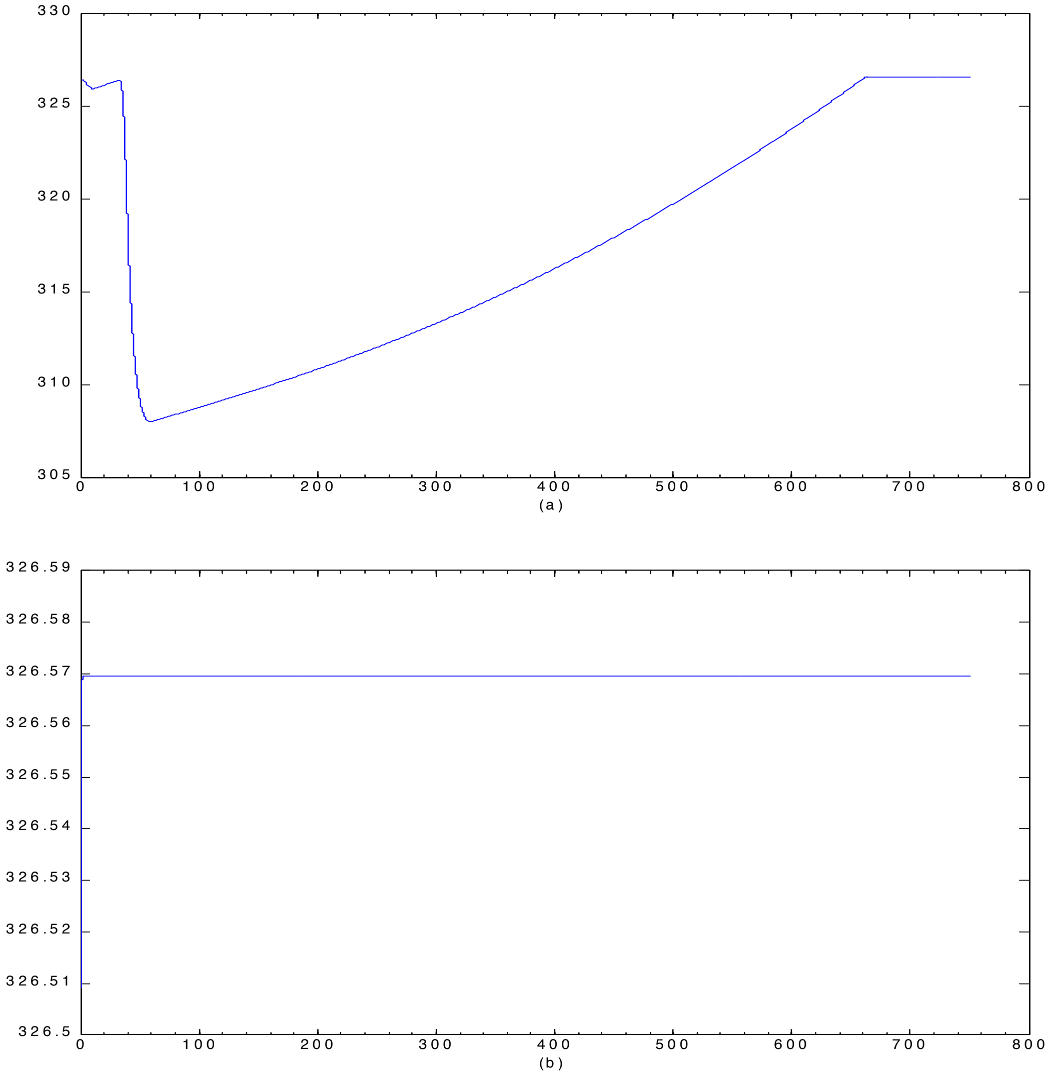


Figure 26: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 10^{-1}$ s. (a) Estimated pressurizer temperature [°C]; (b) Pressurizer reference temperature [°C]

1.3.4 Digital Inventory `Pressurizer_inventory_controller_1k` *and Pressure Control* `Pressurizer_pressure_controller_1k` *with Sampling Time $\delta = 1$ s*

The simulation results are summarized in Figures 27–33.

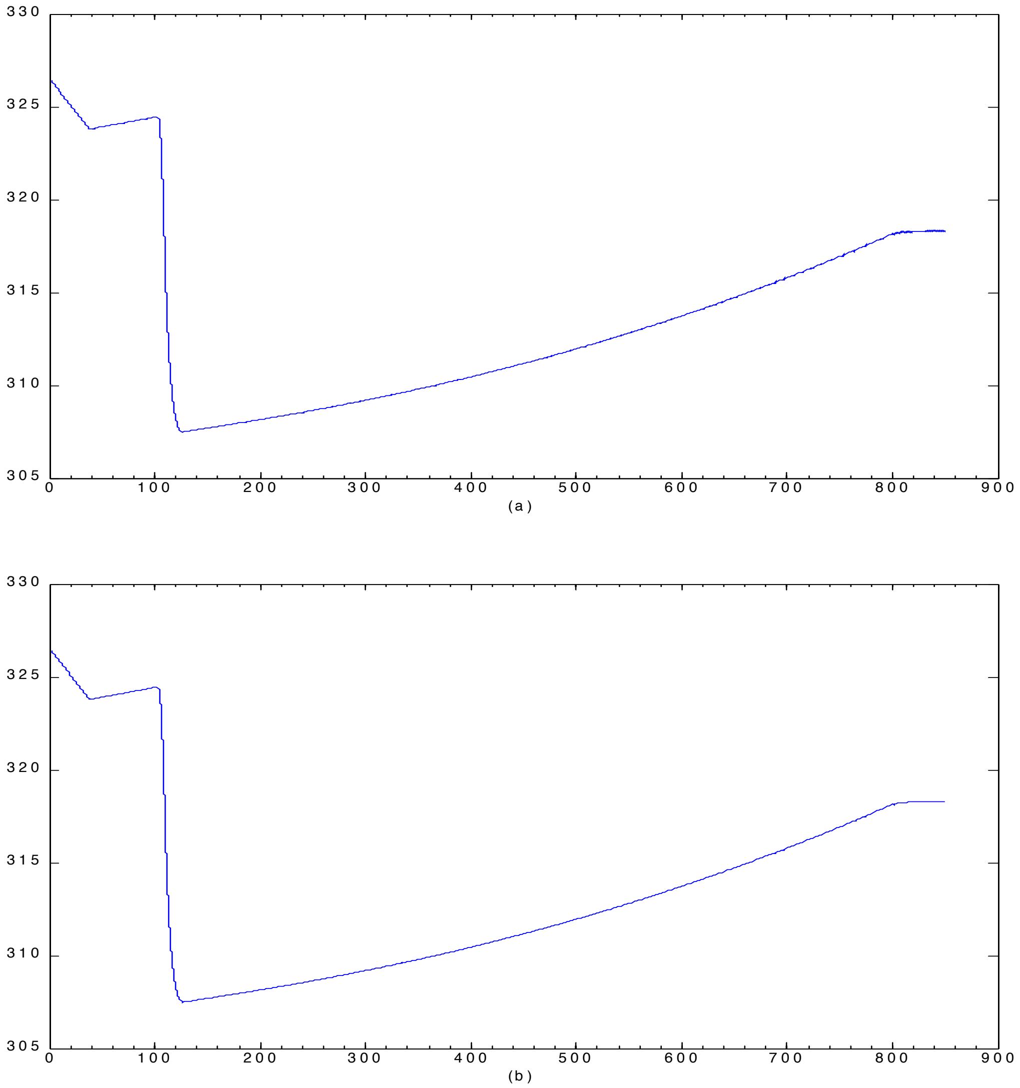


Figure 27: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Pressurizer temperature [°C]; (b) Pressurizer wall temperature [°C]

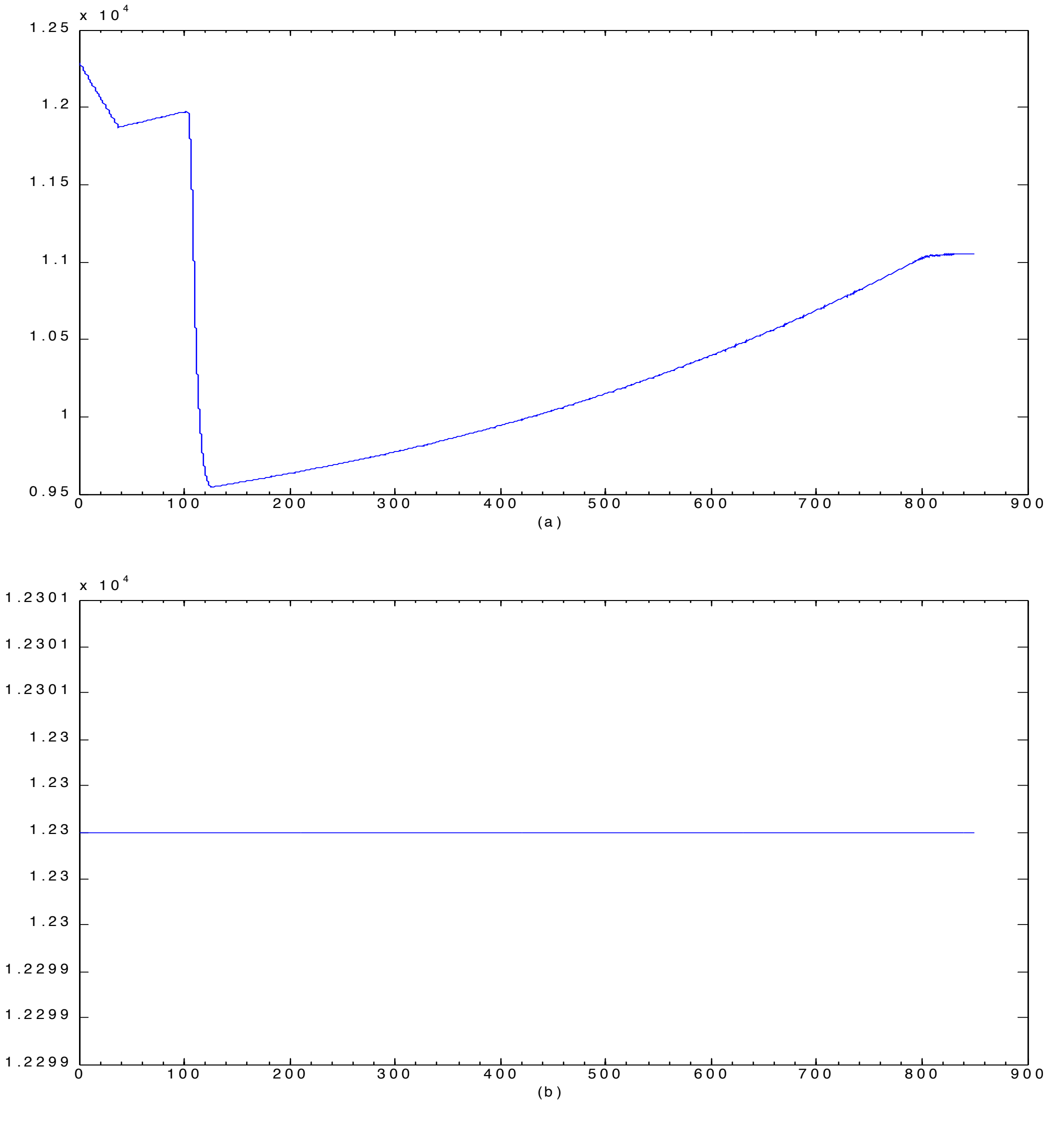


Figure 28: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Pressurizer pressure [Pa]; (b) Pressurizer pressure reference [Pa]

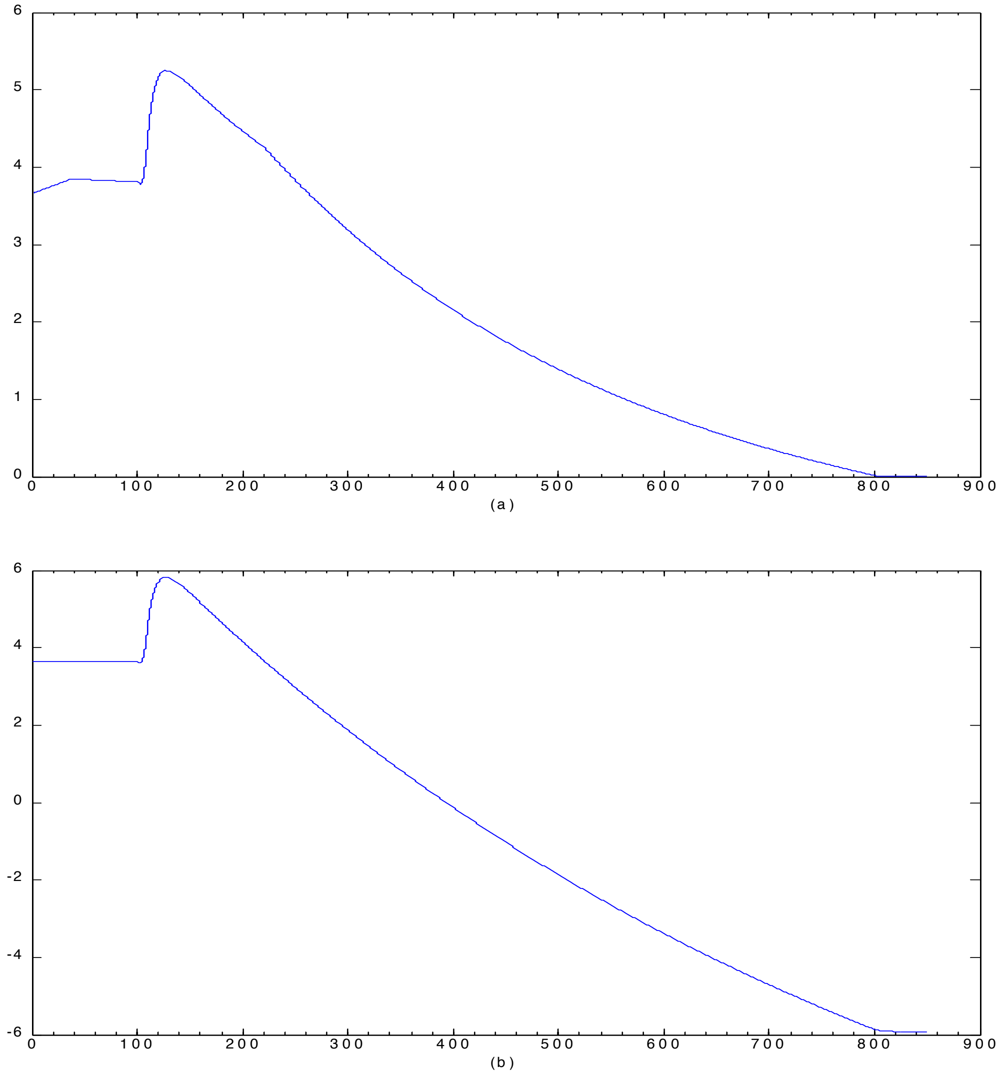


Figure 29: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Pressurizer water level [m]; (b) Pressurizer water level reference [m]

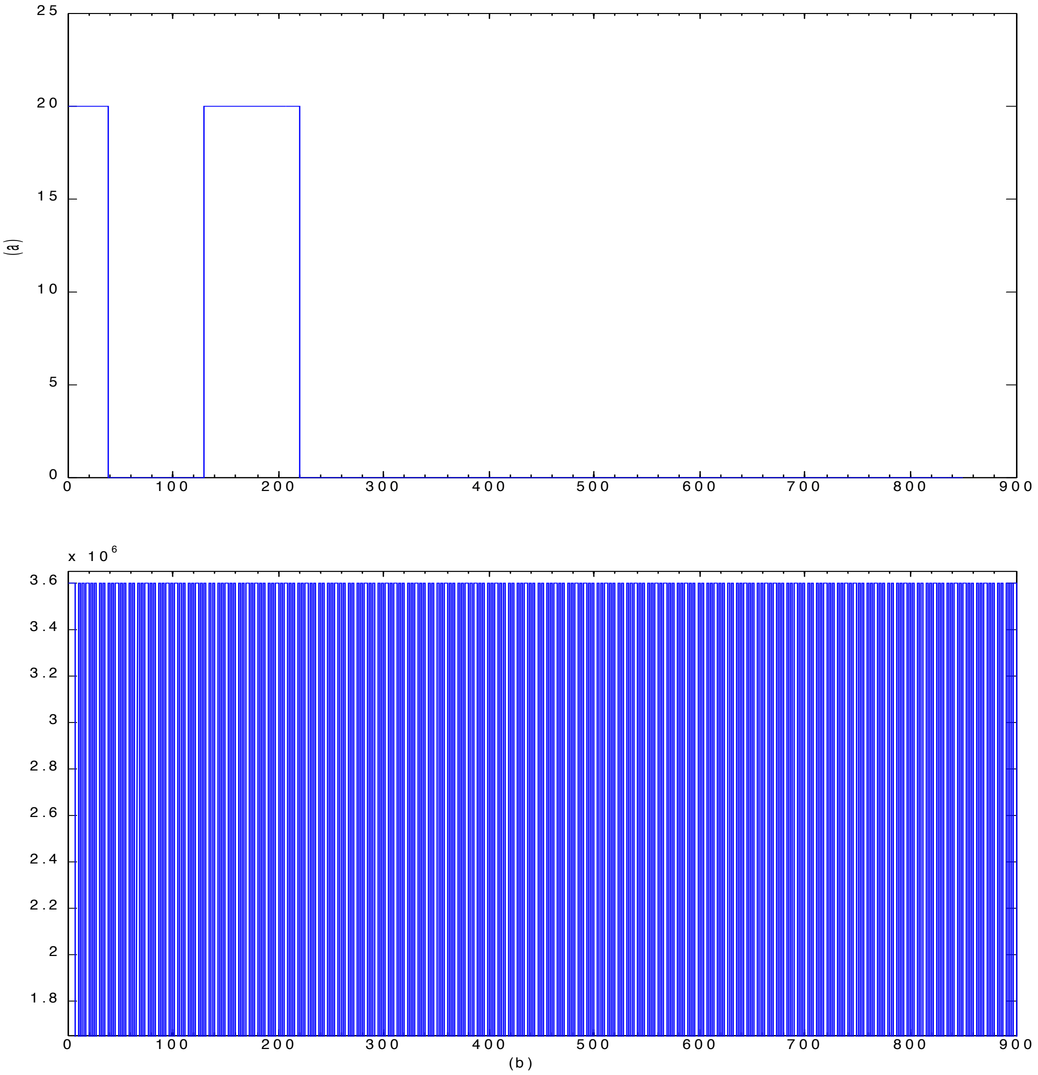


Figure 30: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Inlet mass flow rate [kg/s]; (b) Pressurizer heating power [W]

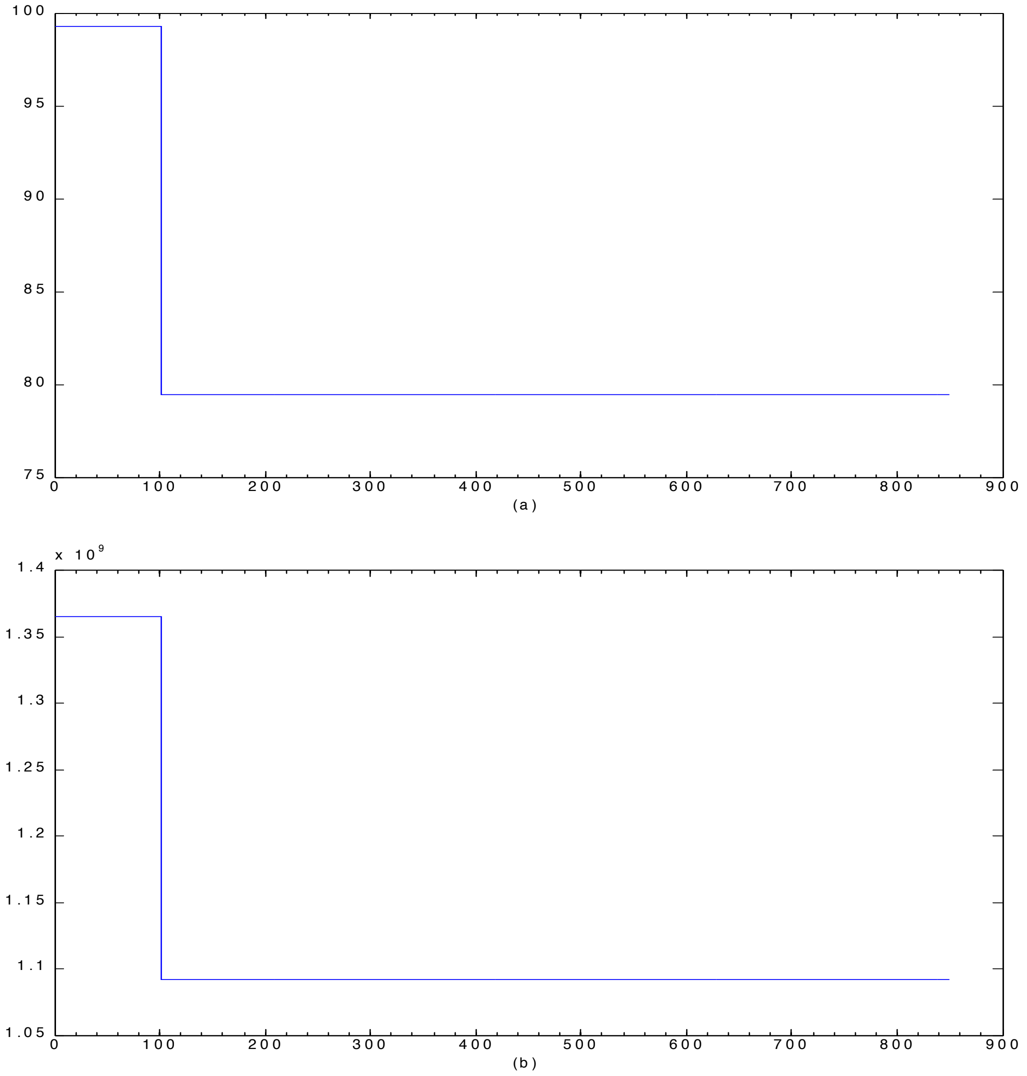


Figure 31: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Neutron flux [%]; (b) Reactor power [W]

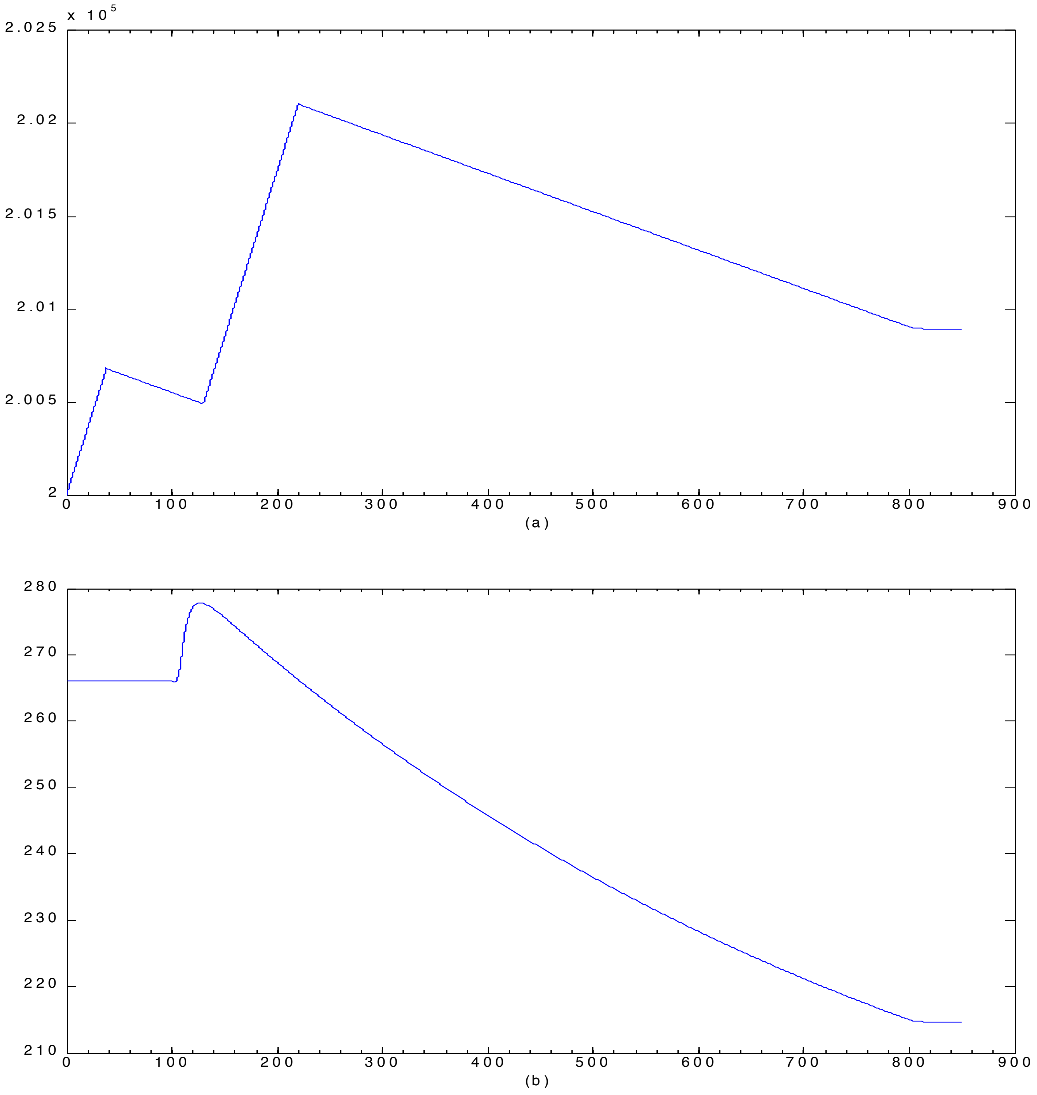


Figure 32: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Primary circuit water mass [kg]; (b) Primary circuit avarege temperature [°C]

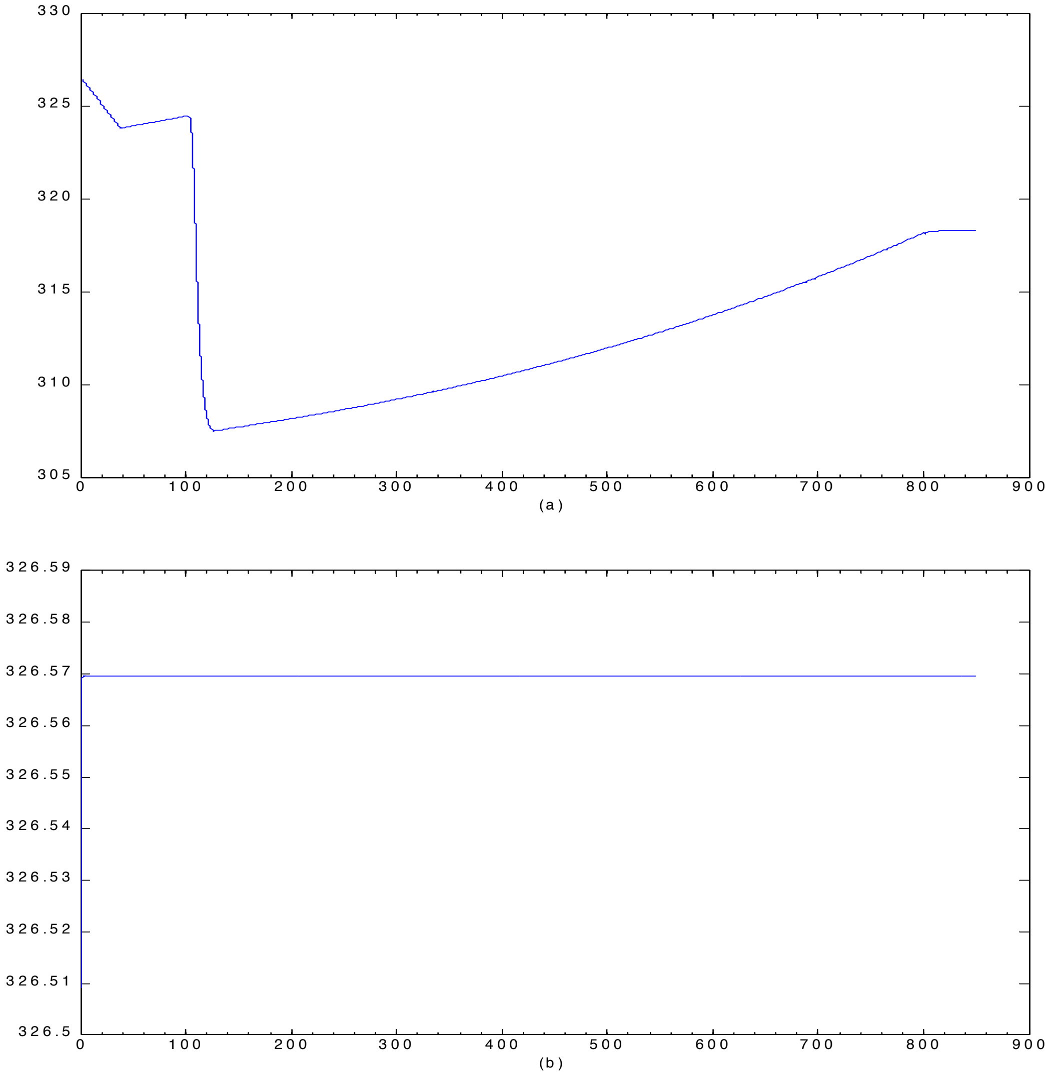


Figure 33: Digital inventory and pressure control `Pressurizer_inventory_controller_1k`, `Pressurizer_pressure_controller_1k` with sampling time $\delta = 1$ s. (a) Estimated pressurizer temperature [°C]; (b) Pressurizer reference temperature [°C]

2 National Instruments LabVIEW: Software and Hardware

The controllers designed and tested on the basis of the simulation environment have to be implemented on a digital device. Before this final step, it is common to insert some hardware in the loop to better check the real behavior of the closed loop system. In particular, it is possible to apply the controller to the real process under study and to calculate the control law considering measurements from the real process. For, it is necessary to use an appropriate real–time environment where is possible to acquire and store data, compute the control algorithm, apply the control action to actuators. Therefore, once the controller has been designed in Simulink, it is necessary to interface Simulink with this real–time environment for further checks of the control algorithm.

One possible choice is LabVIEW$^{©}$, described in the following. Even though LabVIEW is not certified for nuclear applications, this choice is motivated by the fact that it is a very popular software for real–time interfaces, simple enough for the purpose of the present project. LabVIEW, hence, allows checking the methodological steps for the real–time prototyping of the controllers. For future industrial applications this software can still be used, while in the case of applications in nuclear environment, more costly nuclear–certified softwares, ensuring the same real–time performances of LabVIEW, can be considered.

2.1 Programming Language: LabVIEW RT

LabVIEW is a graphic programming language that makes use of icon instead of code lines to create a specific application. This is a software language based on data flow, that determines a program execution. In LabVIEW environment is possible to implement a user interface using objects and tools. This mentioned interface is known as front panel. Code is added adopting a graphic representation of different functions to manage front panel objects: the block diagram checks this elements. LabVIEW is totally integrated to communicate through GPIB, VXI, PXI, RS–232, RS-485 interface and plug–in DAQ. This National Instruments property language is useful to create test and measurement application, to acquire data, instruments control, data storage and analysis.

LabVIEW programs are called Virtual Instruments and they contain three fundamental elements:

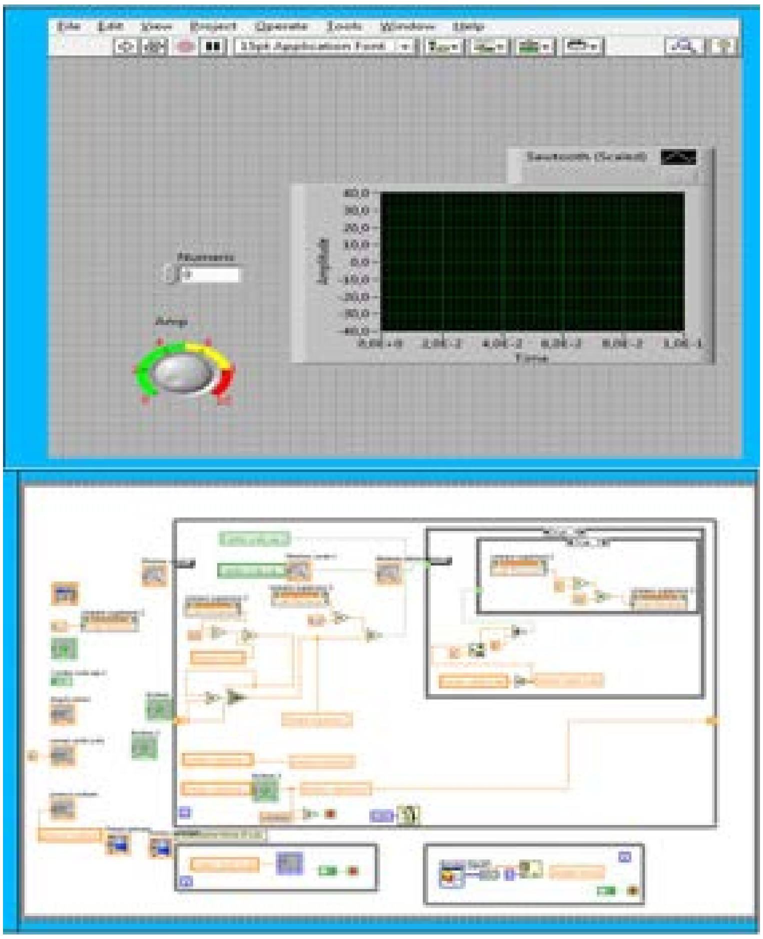


Figure 34: LabView Programming Language Example of Back–End Layer and Applicative User Interface

a front panel, a block diagram and one icon with the connector box. The front panel has indicators, controls, buttons and other elements. Indicators are graphs, LED, etc. Controls simulate instrument input and provide data to the block diagram. Indicators simulate device output and display data that the block diagram generates and acquires. Primary potentiality of LabVIEW resides in its hierarchical nature of programming: it is possible to create subroutines, that can be recalled and reused in other projects. In addition to the LabVIEW real time module, an FPGA one is available to create a lower level application to program the FPGA. This kind of module requests some important rules and tricks to optimize the FPGA resources usage and exploit its potentials.

2.2 Introduction to the CompactRIO Architecture

The hardware architecture employed in this project includes a National Instruments product known as CompactRIO, a reconfigurable, embedded system implementing data acquisition and manipulation [27]. This kind of architecture presents I/O modules, a FPGA chassis and an embedded controller, programmable through LabVIEW programming language. The main reason leading to choose this system typology is the ability to make advanced analysis, but its potentialities also cover elevated signals elaboration, control algorithms from immediate PID systems to dynamic ones like model predictive controls (MPCs). All of these features are well–defined to achieve an adequate level of determinism. Another benefit is represented by the modularity and flexibility guaranteed by this platform: as a matter of fact a large range of controller, reconfigurable chassis and I/O modules can be included to easily change over from prototype to production.

The real–time controller is a microprocessor that executes LabVIEW real–time code in a reliable and deterministic way offering control, datalogging and peripheral communication. Because each acquisition module is connected directly to the FPGA rather than through a bus, there is almost no control latency for system response compared to other controller architectures.

By default, this FPGA automatically communicates with I/O modules and provides deterministic I/O to the real–time processor through a PCI bus communication. Out of the box, the FPGA enables programs on the real–time controller to access I/O with less than 500 ns of jitter between loops. You can also directly program this FPGA to further customize the system. Because of the FPGA speed, this chassis is frequently used to create controller systems that incorporate high-speed buffered I/O, very fast control loops, or custom signal filtering. For instance, using the FPGA, a single chassis can execute more than 20 analog proportional integral derivative (PID) control loops simultaneously at a rate of 100 kHz. Additionally, because the FPGA runs all code in hardware, it provides the highest reliability and determinism that is ideal for hardware-based interlocks, custom timing and triggering, or eliminating the custom circuitry normally required with nonstandard sensors and buses.

Lastly I/O modules comprise isolation stages, conversion and signal conditioning for a direct connection to sensors and motor units. Modules easily available on the market includes several signal typologies inputs and actuators: thermocouple inputs, current or voltage, strain gauges, or digital inputs/outputs. Additionally, you can build your own modules or purchase modules from third-party vendors.

A similar platform has been chosen because with LabVIEW FPGA is possible to

1. Collect analog waveform at rates of hundreds of kilohertz;

2. Create custom digital pulse trains at up to 40 MHz;

3. Implement custom digital communication protocols;

4. Run control loops at rates in the hundreds of kilohertz;

5. Use modules not supported by the scan mode including CAN and PROFIBUS communication;

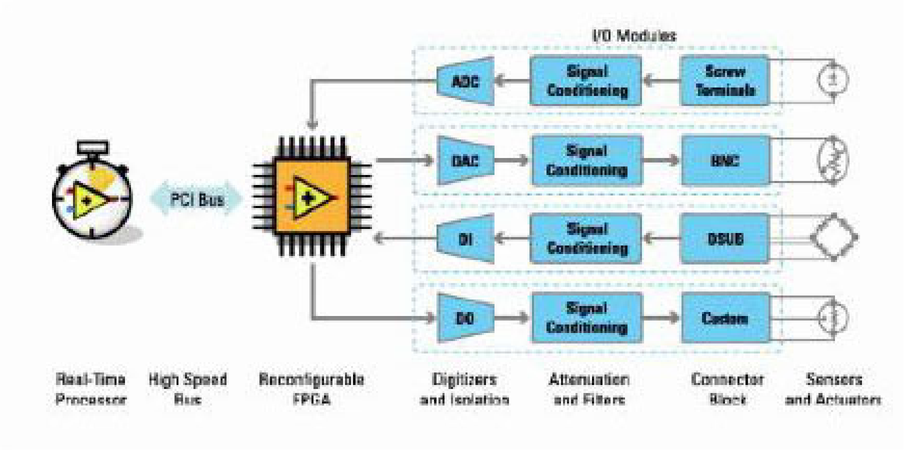


Figure 35: Scheme of CompactRIO Embedded System, Exploiting Processor and FPGA Technologies

6. Implement custom timing, triggering, and filtering.

With the combination of a real–time processor and programmable FPGA on CompactRIO, we take advantage of the strengths of each computing platform. The real–time processor excels at floating–point math and analysis and peripheral communication such as network-published shared variables and Web services. The FPGA excels at smaller tasks that require very high-speed logic and precise timing. A scenario for which the FPGA is programmed directly can include

1. High–Speed Waveform Acquisition /Generation (greater than 1 kHz).

   If an aquisition or generation at speeds higher than 1 kHz is needed to take full advantage of these module features, LabVIEW FPGA can acquire at a user-defined rate tailored to a specific application.

2. Custom Triggering/Timing/Synchronization.

   With the reconfigurable FPGA, a programmer can create simple, advanced, or otherwise custom implementations of triggers, timing schemes, and I/O or chassis synchronization. These can be as elaborate as triggering a custom CAN message based on the rise of an analog acquisition exceeding a threshold or as simple as acquiring input values on the rising edge of an external clock source.

3. Hardware–Based Analysis/Generation and Coprocessing.

   Many sensors output more data than can be reasonably processed on the real–time processor alone. The FPGA can be used as a valuable coprocessor to analyze or generate complex signals while freeing the processor for other critical threads.

This type of FPGA based coprocessing is commonly used in applications such as

Encoding/decoding sensors

(a) Tachometers

(b) Standard and/or custom digital protocols

Signal processing and analysis

(a) Spectral analysis (fast Fourier transforms and windowing)

(b) Filtering, averaging, and so on

(c) Data reduction

(d) Third-party IP integration

Sensor simulation

(a) Linear-voltage differential transformers (LVDTs)

(b) Cam and crank

Hardware-in-the-loop simulation.

4. Highest–Performance Control

   Not only the FPGA can realize high-speed acquisition and generation, but also can implement many control algorithms on the FPGA. You can use single–point I/O with multichannel, tunable PID or other control algorithms to implement deterministic control with loop rates up to hundreds of kilohertz.

5. Unsupported Modules

   Several C Series modules do not feature scan mode support. For these modules, you need to use LabVIEW FPGA to build an interface between the I/O and your real–time application. For a list of modules that feature scan mode support, see C Series Modules Supported by CompactRIO Scan Mode. Unsupported Targets CompactRIO targets with 1M gate FPGAs cannot fully support the scan mode. You can implement some scan mode features on unsupported targets, but you must use LabVIEW FPGA. The knowledge base article Using CompactRIO Scan Mode with Unsupported Backplanesi describes how to use LabVIEW FPGA to build a custom scan mode interface for an unsupported FPGA target.

Because LabVIEW FPGA code runs directly on hardware, the main advantages of FPGA–based design are:

1. *High Reliability*

   LabVIEW FPGA code running on a FPGA is highly reliable because the logic is compiled into a physical hardware design. Once the FPGA is programmed, it becomes a hardware chip with all of the associated reliability.

2. *High Determinism*

   Processor-based systems often involve several abstraction layers to help schedule tasks and share resources among multiple processes. The driver layer controls hardware resources and the operating system manages memory and processor bandwidth. For any given processor core, only one instruction can execute at a time. Real-time operating systems reduce jitter to a finite maximum when programmed with good priority hierarchy. FPGAs do not use any operating systems. This minimizes reliability concerns with true parallel execution and deterministic hardware dedicated to every task. For NI FPGA–based hardware, you can achieve 25 ns timing accuracy of critical components within your design.

3. *True Parallelism*

   Multithreaded applications break down into multiple parallel sections of code which are executed in a round-robin fashion, giving the appearance of parallel execution. Multicore processors expand on that idea by allowing multithreaded applications to truly execute multiple parallel code at one time. The number of parallel pieces of code executing concurrently is limited to the number of cores available in the specific processor. Because an FPGA implements parallel code as parallel circuits in hardware, you are not limited by processor cores; therefore, every piece of parallel code in an entire FPGA application can execute concurrently. Even traditionally serial operations can improve throughput on FPGAs by implementing pipelining.

4. *Reconfigurability*

   Being reconfigurable, FPGA chips are able to keep up with any future modifications you might need. As a product or system matures, functional enhancements are always feasable without spending time redesigning hardware or modifying a board layout. This is especially applicable to industrial communication protocols. As communication protocols evolve and improve over time, you can modify the implementation of that protocol within an FPGA to support the latest technology features and changes.

5. *Instant Boot Up*

   Because LabVIEW FPGA code runs directly on the FPGA without an overarching operating system, the code downloaded to the FPGA flash memory begins running within milliseconds of powering on your CompactRIO chassis. This code can begin executing a control loop or setting startup output values.

2.3 Mathworks–LabVIEW Interface

Designed to offer a desktop interface to mathematically manage a model, the languages known as .m files simplify the process of development of intellectual algorithms and IP but they often complicate the embedded hardware conversion. This kind of programs, used in software as MathWorks and Scilab, manipulate data as a numerical matrix, in this way the concept of ”data type” is not realized and there is a dynamic allocation of memory. This can be a strong limit in embedded architectures and OS cannot operate in similar conditions, due to timing constraints and their deterministic nature. Another aspect to focus that .m files are not compiled but interpreted: without a code compilation the fundamental benefit of errors identification before the execution is lost. Furthermore, this language does not contain timing and resources management causing the code to be written once again in a more suitable one (such as C) to program the embedded system. Taking a script developed using Matlab and replicating it in a multicore realtime hardware can request the depicted steps.

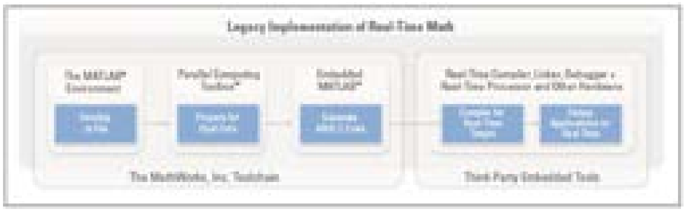
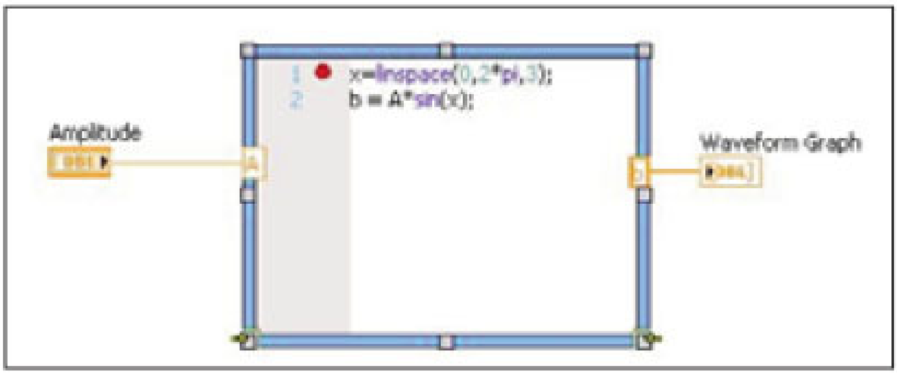

Legacy Implementation of Real-Time Math
The MATLAB® Environment
Parallel Computing Toolbox™
Embedded MATLAB®
Real-Time Compiler, Linker, Debugger + Real-Time Processor and Other Hardware
The MathWorks, Inc. Toolchain
Third-Party Embedded Tools

Figure 36: Main steps passing through Simulink environment to a real time one

With the matlab syntax, firstly it is necessary to test a script in the desktop development environment using the Parallel Computing toolbox to prepare code for a dual-core environment. Then the Embedded Matlab is adopted to generate C code and in the end the code has to be compiled and debugged in a seperate embedded toolchain. This path can be very dangerous for timing and precision of the mathematical implementations.

In this scenario, NI provides a Mathscript module that provides a textual programming math-oriented through a native compiler for .m files. In this way every .m file can be included in realtime hardware. The following figure shows the Mathscript Node which permit to execute scripts .m from a LabVIEW VI.



```
x=linspace(0,2*pi,3);
b = A*sin(x);
```

Figure 37: Matlab rootines read in a Labview application

A first advantage of the LabVIEW compiler is the capacity to express parallelism: there is no need for special markup or in the code to force a parallelism on the compiler, feature needed for a textual programming language.

Another important feature about the Labview–Matworks linking is the possibility to reproduce Simulink projects in the LabVIEW development, just in a few steps. The NI LabVIEW Simulation Interface Toolkit gives control system design and test engineers a link between the NI LabVIEW graphical development environment and The MathWorks, Inc. Simulink software. With the LabVIEW Simulation Interface Toolkit, it is easily possible to build custom LabVIEW user interfaces to view and control your simulation model during run time. This toolkit also provides a plug–in for use with The MathWorks, so designers are allowed to connect, using LabVIEW, a model developed in the Simulink environment, to the real world through a variety of real–time I/O platforms. With these capabilities, you can easily take your models from software verification to real–world prototyping and hardware–in–the–loop simulation. This particular characteristic was exploited and described to implement the controller shown in the previous sections.

3 The Use of FPGA in Industrial Applications

In this section we will present an experimental set–up that has been used to show the benefit of the FPGA features, to be integrated with the simulation environment previously presented, in order to better validate the control laws described in [3]. This set–up implements a fist prototype of a supervisory tool for the TRIGA reactor RC–1 [26], in the Rome Casaccia Centre. In the first part, it will be shown how Simulink and LabVIEW are used to obtain a tool capable of simulating mathematical model described by differential equations, with the possibility of 'downloading' the resulting numerical model on a mixed CPU–FPGA architecture. LabVIEW allows users to create an intuitive and user friendly panel, very useful to have a real–time synoptic representing the model state. In the second part, we will focus on the possibility to push control algorithms to very high performances, describing VHDL code generation and digital circuits synthesis through FPGA family development tools.

3.1 A Digital Supervisory Tool using LabVIEW Development Environment

LabVIEW allows users to create an intuitive front panel easy to customize. An example of such front panel is given in Figure 38. In this application the pressurizer model previously described has been reproduced with the three heaters at its bottom. A similar synoptic loads the Simulink model and parameters and it visualize them in a more user–friendly way, thus resulting in a better perception of the process real estate. Beside the pressurizer, some analog indicators have been implemented such as pressure, power and temperature monitoring. During the simulation all the implemented elements are dynamically animated, showing the same behavior recorded with simple graphs of the variables time evolution. Just using the simulation interface toolkit simulations have become more realistic and in this direction an hypothetical step to download a similar architecture on a realtime system would be immediate.

In the following section, the second test application will be presented to underline the advantages from the use of CompactRIO. This test shows the importance of such a digital architecture in supervision and monitoring systems: in the so–called old generation nuclear plant a digitalization process will be desirable in order to improve the HMI and to have a user–friendly visualization beside the always indispensable analog instrumentation. In this sense some important variables have been acquired and

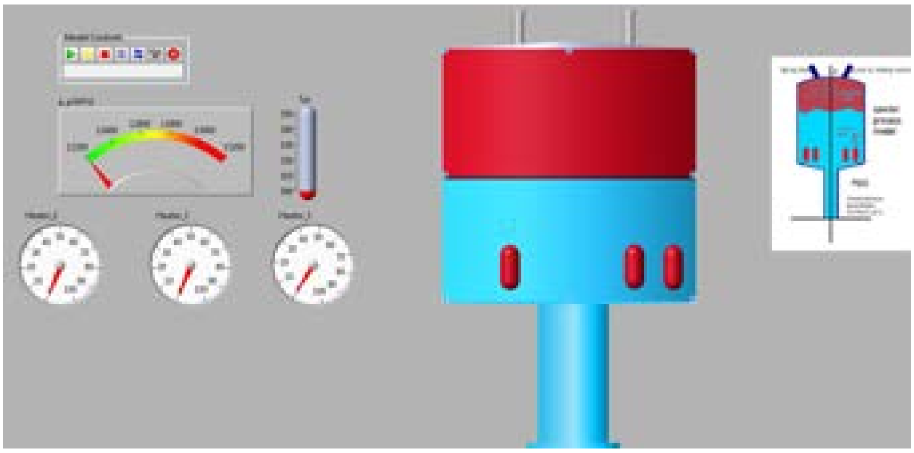


Figure 38: Synoptical view of the LabVIEW application loading the Simulink model

presented to the plant operator in a intuitive way, also with the possibility to record data in a dedicated memory and reproduce some particular plant conditions.

3.1.1 Test Bench: TRIGA RC–1 Reactor

TRIGA RC–1 is a pool thermal reactor having a core contained in an aluminum vessel and placed inside a cylindrical graphite reflector, bounded with lead shielding. The biological shield is provided by concrete having mean thickness of 2.2 m. Demineralized water, filling the vessel, ensures the functions of neutron moderator, cooling mean and first biological shield. Reactor control is ensured by four rods: two shims, one safety fuel–follower rods and one regulation rod. Produced thermal power is removed by natural water circulation through a suitable thermo–hydraulic loop including heat exchangers and cooling towers. Some irradiation facilities are listed below.

The core and the reflector assemblies are located at the bottom of an aluminum tank (190.5 cm diameter). The overall height of the tank is about 7 m, therefore the core is shielded by about 6m of water. The core, surrounded by the graphite reflector, consists of a lattice of fuel elements, graphite dummy elements, control and regulation rods. There are 127 channels divided in seven concentric rings (from 1 to 36 channels per ring). The channels are loaded with fuel rods, graphite dummies and regulation and control rods depending on the power level required. One channel houses the start–up Am–Be source, while two fixed channels (the central one and a peripheral) are available for irradiation or experiments. A pneumatic transfer system allows fast transfer from the peripheral irradiation channel and the radiochemistry end station. The diameter of the core is about 56.5 cm while the height is 72 cm. Neutron reflection is provided by graphite contained in an aluminum container, surrounded by 5 cm of lead acting as a thermal shield. The fuel elements consist of a stainless steel clad (AISI–304, 0.05 cm thick, 7.5

g/cm^3 density) characterized by an external diameter of 3.73 cm and a total height of 72 cm, end cap included. The fuel is a cylinder (38.1 cm high, 3.63 cm in diameter, 5.8 g/cm^3 of density) of a ternary alloy uranium–zirconium–hydrogen (H–to–Zr atom ratio is 1.7 to 1; the uranium, enriched to 20% in ^{235}U, makes up 8.5% of the mixture by weight: the total uranium content of a rod is 190.4 g, of which 37.7 g is fissile) with a metallic zirconium rod inside (38.1 cm high, 0.5 cm in diameter, 6.49 g/cm^3 of density). There are two graphite cylinders (8.7 cm high, 3.63 cm in diameter, 2.25 g/cm^3 of density) at the top and bottom of the fuel rod. Externally two end–fittings are present in order to allow the remote movements and the correct locking to the grid. The regulation rod has the same morphological aspect as the fuel rod: the only difference is that instead of the mixture of the ternary alloy Uranium–Zirconium–Hydrogen there is the absorber (graphite with powdered boron carbide). The control and safety rods are 'fuel followed': the geometry is similar to that of the regulation rod but with fuel element at its bottom. The graphite dummies are similar to a fuel element but the cladding is filled with graphite. Figure 39 shows an horizontal section.

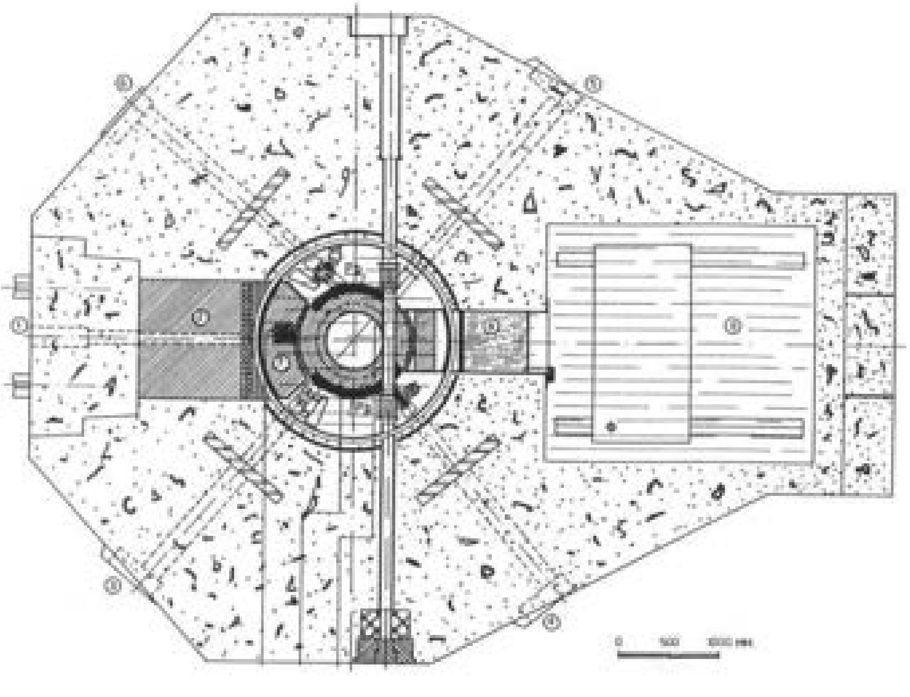


Figure 39: Horizontal section of TRIGA RC–1 nuclear research reactor at ENEA–Casaccia research Centre

The parameters used in order to perform the reactor monitoring can be classified into three large groups: power monitoring, process monitoring and radiological monitoring. The reactor power is monitored by means of one starting channel (0.0 to 1W), two wide range linear channels (0.5 -5.0o106 W) and one safety channel (10 kW to 1.1 MW). The process monitoring includes 6 temperatures (fuel elements, primary and secondary loops, cooling towers), flow rates (primary and secondary loops, water cleaning system, reactor hall air), levels (reactor pool, shielding tank), conductivities (primary loop, shielding tank loop). The radiological control is carried out by monitoring water activity (primary and secondary loops), air activity (reactor hall and experimental channels) and environmental radiation levels (reactor hall, control room and experimental channels). Only main plant parameters are mentioned herein, but a lot of secondary parameters can be easily monitored (such as control rods positions, switches status, alarms and so on).

3.1.2 Analyzed Parameters

In this prototypical phase, in way of obtaining a complete knowledge about the plant and representing the operation of power increase, three kind of signals have been isolated

1. Variables representing generated power by the nuclear reactor.
2. Temperatures concerned with positions in the core and in the linked thermo–hydraulic loops.
3. Signals referred to rods position.

A logarithmic amplifier provides power value from the shutdown to the full power status, this is possible because the amplifier is linked to a ionization chamber covering a current range from 100 pA to 100μA. Variables coming from thermocouples are stored for a temperature monitoring. Conditioning of thermocouple signals is available for the input of our modules through transducers. The most important temperatures collected are process temperatures (well surface, purificator, towers entrance, exchangers entrance, etc.), and fuel temperatures recorded in two different positions of the core. An indication of rod positions (two shim rods, safety and regulation rod) is obtained through potentiometer repeater. On the front panel, numeric indicators presented on the control console are replicated, together with the luminous indicators representing: rod at full lower stroke and full upper stroke; rod uncoupled from the electromagnet. The system synoptic faithfully depicts a similar representation of the control panel on the console. Extraction and insertion (fixed–speed) of a selected rod is determined raising or lowering a lever, with a central equilibrium position. In the following picture the synoptic just described is shown.

At this stage, 45 different signals have been identified and sampled by the supervisory tool. The number of channels and the frequency of acquisition would not justify the choice of using a FPGA architecture: in fact, the real–time processor alone is presently able to guarantee a satisfactory result in terms of syncronism and parallelism of different process, but it could become obsolete increasing the number of signals and the rate of acquisition. In this sense, using an FPGA from scratch can satisfy

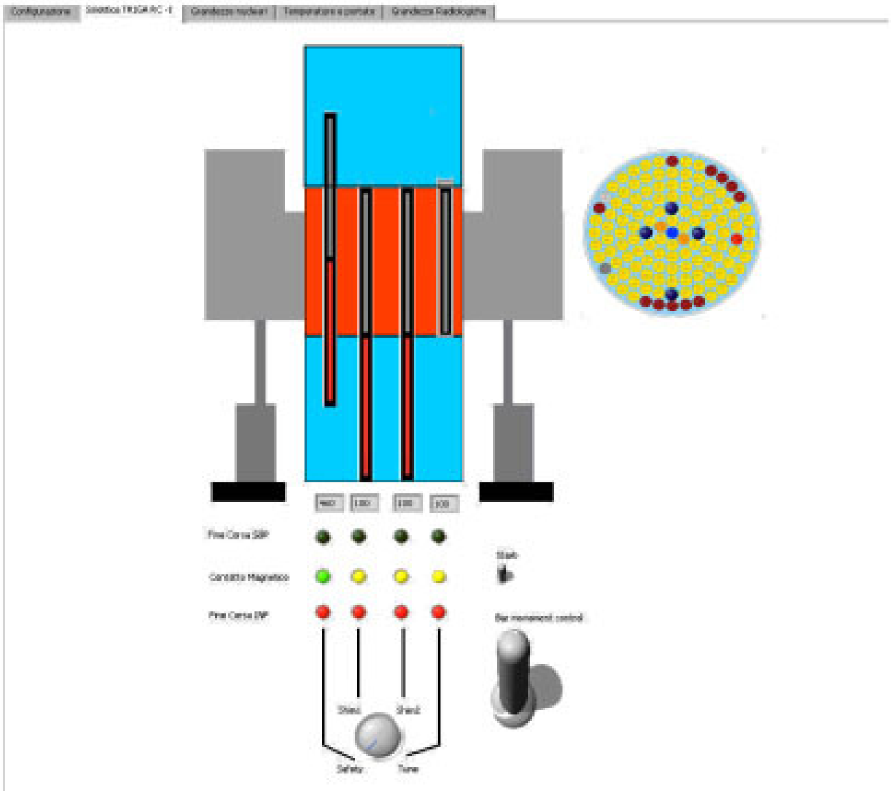


Figure 40: A Software Synoptic of the Real TRIGA RC–1 Control Console

the requirements of real–time acquisition and effective independence between processes also with more severe specs and facilitating a possible process of the system licensing.

3.1.3 Process on Investigation: Attainment of Maximum Power (1MW)

The plant process on investigation represents the transient state reaching the maximum power of operation. 1 MW top power is achieved by plant operators respecting some operating rules. Our digital system is able to monitor and save all of the user maneuvers, therefore in the following graphs main parameters development is shown. This process took 30 minutes to run out and in this lapse of time the prototype worked in parallel providing a user–friendly visualization, permitting to memorize in .txt files and offering its front panel on web server through the dedicated LAN.

The Figure 41 depicts the control rods rise, during the phenomenon formerly described. It is possible to recognized the correct sequence of rod rising as in a safe operating maneuver, in sequence: safety, regulation, shim1 and shim2 rod.

The power transient is shown from start–up to 1 MW state during the process. Two fuel elements, far–between in the core, recorded this kind of trend during this transient state, falling completely into fuel range of granted temperatures.

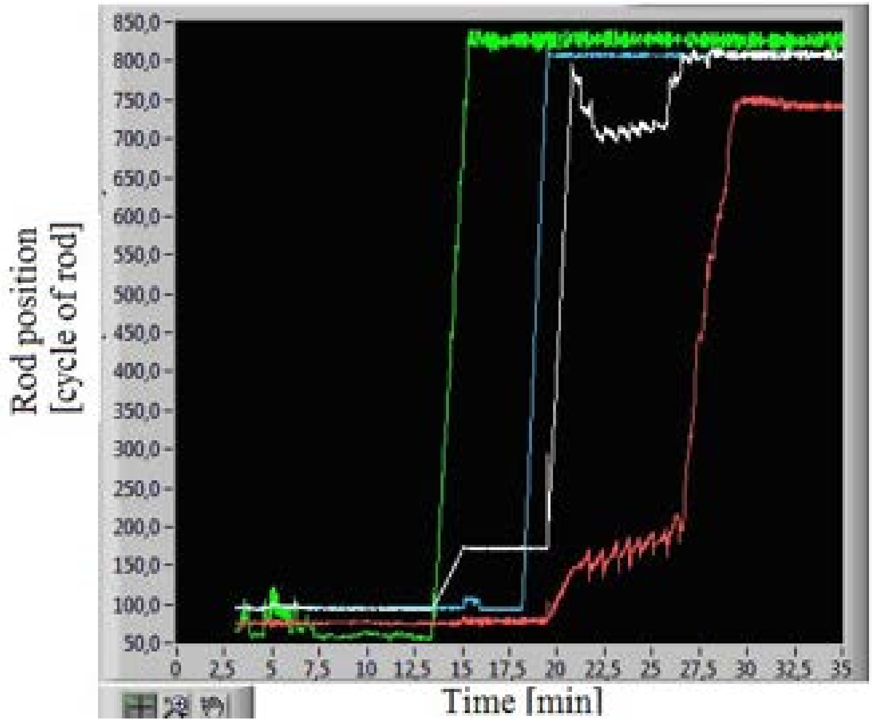


Figure 41: Transient State of Rod Position during Power Increase

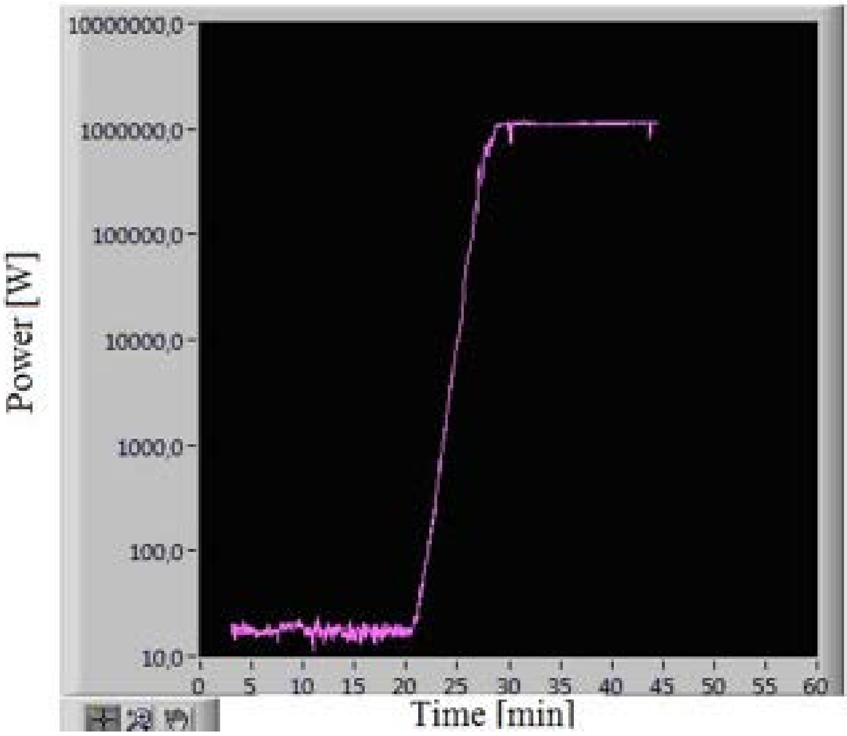


Figure 42: Power Evolution Reaching 1 MW vs time [min] – Power Values are Expressed in W

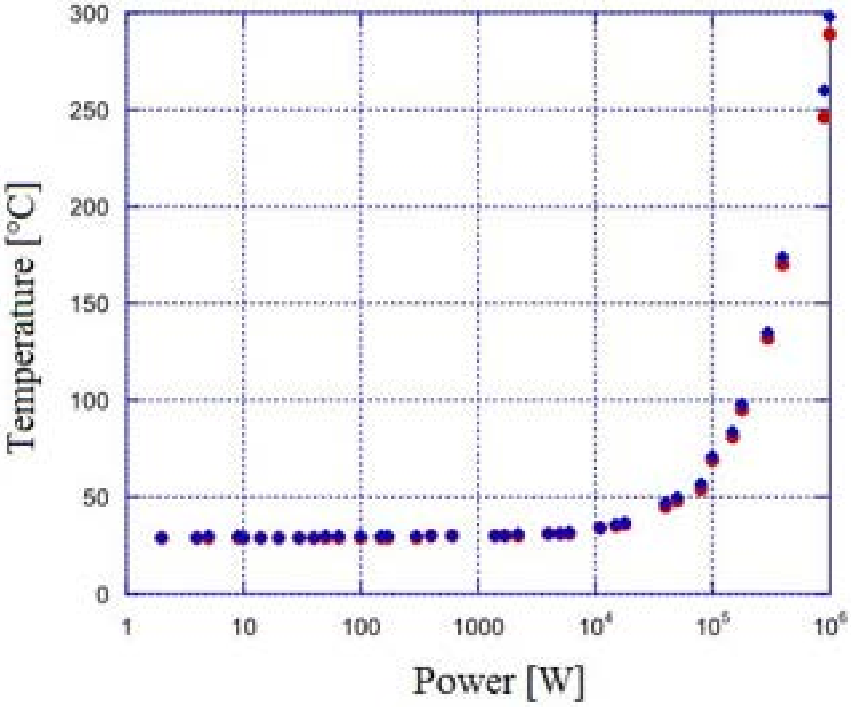


Figure 43: Fuel Temperature during Power Increase

A further useful tool, presented in the proposed supervisory system, is a dynamic visualization of the reactor synoptic: indeed an operator is able to check real–time rod positions in a very intuitive way compared to a digital indication of rod cycles. This user–friendly methods of displaying is conceived to facilitate the analogue to digital changeover, as required for plant console modernizations.

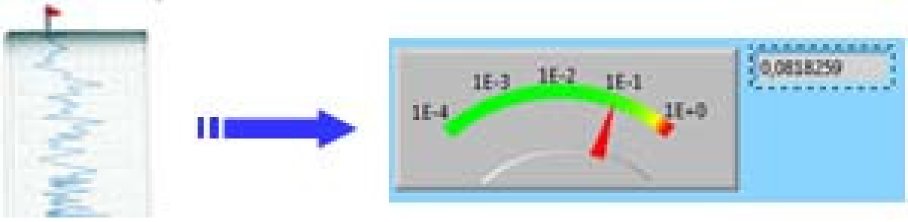


Figure 44: Analog to Digital Changeover using LabVIEW Tools

3.2 Generic Controller Implementation on FPGA–Based Platforms

In the previous sections we described how control algorithms can be simulated and then implemented on CPUs systems or hybrid CPU–FPGA architectures, achieving good performances in terms of time, programming complexity and graphical interface improvements. In this section we focus on the possibility to push control algorithms to very high performances, describing two different approaches: VHDL code generation and digital circuits synthesis through FPGA family development tools, as maintaining the compactRIO architecture with a lower level approach making use of the FPGA module. As a test application we defined a PID control loop algorithm in the different described design flows. It is good to underline the fact that we focused on the "PID core" implementation, presuming that problems such as signals conversion, conditioning and compatibility have already designed and solved.

3.2.1 Flow Design of Digital Architectures Using VHDL Code Generation

VHDL is a language for describing digital electronic systems. It arose out of the United States Government's Very High Speed Integrated Circuits (VHSIC) program, initiated in 1980. It became clear that there was a need for a standard language for describing the structure and function of integrated circuits (ICs). Hence the VHSIC Hardware Description Language (VHDL) was developed, and subsequently adopted as a standard by the Institute of Electrical and Electronic Engineers (IEEE) in the US. VHDL is designed to fill a number of needs in the design process. Firstly, it allows description of the structure of a design, that is how it is decomposed into sub–designs, and how those sub–designs are interconnected. Secondly, it allows the specification of the function of designs using familiar programming language forms. Thirdly, as a result, it allows a design to be simulated before being manufactured, so that designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

1. Describing structure

   A digital electronic system can be described as a module with inputs and/or outputs. The electrical values on the outputs arefunction of the values on the inputs. Figure 45 shows an example of this view of a digital system. The module F has two inputs, A and B, and an output Y. Using VHDL terminology, we call the module F a design entity, and the inputs and outputs are called ports. One way of describing the function of a module is to describe how it is composed of sub–modules. Each of the sub–modules is an instance of some entity, and the ports of the instances are connected using signals. Figure 45 also shows how the entity F might be composed of instances of entities G, H and I. This kind of description is called a structural description. Note that each of the entities

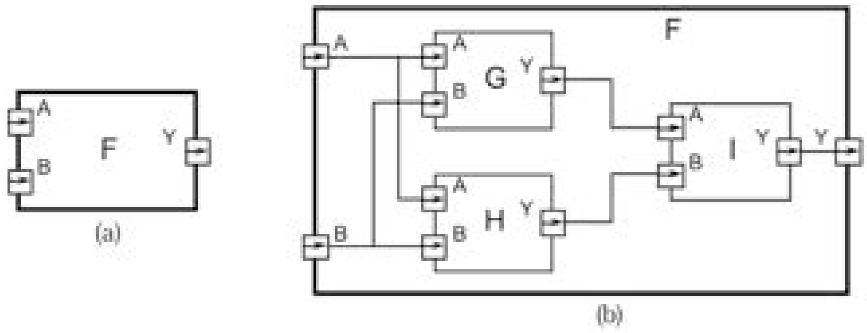


Figure 45: Example of a structural description

G, H and I might also have a structural description.

2. Describing Behaviour

   In many cases, it is not appropriate to describe a module structurally. One such case is a module which is at the bottom of the hierarchy of some other structural description. For example, if you are designing a system using IC packages bought from an IC shop, you do not need to describe the internal structure of an IC. In such cases, a description of the function performed by the module is required, without reference to its actual internal structure. Such a description is called a functional or behavioural description. To illustrate this, suppose that the function of the entity F in Figure 45 is the exclusive–or function. Then a behavioural description of F could be the Boolean function: $Y = \overline{A}.B + A.\overline{B}$

   More complex behaviours cannot be described purely as a function of inputs. In systems with feedback, the outputs are also a function of time. VHDL solves this problem by allowing description of behaviour in the form of an executable program.

Once the structure and behavior of a module have been specified, it is possible to simulate the module by executing its behavioral description. This is done by simulating the passage of time in discrete steps. At some simulation time, a module input may be stimulated by changing the value on an input port. The module reacts by running the code of its behavioral description and scheduling new values to be placed on the signals connected to its output ports at some later simulated time. This is called scheduling a transaction on that signal. If the new value is different from the previous value on the signal, an event occurs, and other modules with input ports connected to the signal may be activated. The simulation starts with an initialization phase, and then proceeds by repeating a two–stage simulation cycle. In the initialization phase, all signals are given initial values, the simulation time is set to zero, and each module's behavior program is executed. This usually results in transactions being scheduled on output signals for some later time. In the first stage of a simulation cycle, the simulated time is advanced to the earliest time at which a transaction has been scheduled. All transactions scheduled for that time are executed, and this may cause events to occur on some signals.

In the second stage, all modules which react to events occurring in the first stage have their behavior program executed. These programs will usually schedule further transactions on their output signals. When all of the behavior programs have finished executing, the simulation cycle repeats. If there are no more scheduled transactions, the whole simulation is completed. The purpose of the simulation is to gather information about the changes in system state over time. This can be done by running the simulation under the control of a simulation monitor. The monitor allows signals and other state information to be viewed or stored in a trace file for later analysis. It may also allow interactive stepping of the simulation process, much like an interactive program debugger.

3.2.2 Example of a PID Digital Implementation through VHDL Code

In this section the PID algorithm described in deliverable1 is reproposed to show how it is immediately possible to create a digital core from a low level description as VHDL code. Firstly a new project in VHDL has been created using the same nomenclature adopted in deliverable1. The code is reported below:

```
library IEEE;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;
use IEEE.STD_LOGIC_1164.ALL;

entity PID_FSM is
  Port ( ADC_DATA : in  STD_LOGIC_VECTOR (15 downto 0);  --16 bit unsigned PID input
         DAC_DATA : out  STD_LOGIC_VECTOR (15 downto 0); --16 bit unsigned PID output
         mr,clk : std_logic);
end PID_FSM;

architecture Behavioral of PID_FSM is
    type state_type is (      Start,
                              CalculateNewError,
                              CalculatePID,
                              SOverload,
                              ConvDac
                              );

     signal state : state_type := Start;

     CONSTANT SetVal : integer := 33259;
     CONSTANT Kp : integer := 10;
     CONSTANT Ki : integer := 20;
     CONSTANT Kd : integer := 1;
     CONSTANT Kg : integer := 256;
begin
     states: process
     variable p,i,d : integer := 0;
     variable Output_Old : integer := 0;
     variable Error_Old : integer := 0;
     variable err: integer := 1000;
     variable out_value : integer := 1;
     variable sAdc : integer := 0 ;


     begin
     if (mr='0') then
          state <= Start;
     end if;
     wait until clk='1';
     case state is
          when Start =>
               sAdc := conv_integer(ADC_DATA);  --Get the input for PID
               DAC_DATA<= conv_std_logic_vector(out_value ,16);
               state <= CalculateNewError;
               Error_Old := err;  --Capture old error
               Output_Old := Out_value;    --Capture old PID output
          when CalculateNewError =>  --
               state <= CalculatePID;
               err := (SetVal-sAdc); --Calculate Error
          when CalculatePID =>
               state <= SOverload;
               p := Kp*(err);              --Calculate PID
               i := Ki*(err+Error_Old);
               d := Kd *(err-Error_Old);
               out_value :=  output_Old+(p+i+d)/2048; --Calculate new output
          when SOverload =>
               state <=ConvDac;
               if out_value > 65535 then
                    out_value := 65535 ;
               end if;
                if out_value < 1 then
                    out_value := 1;
               end if;
          when ConvDac =>           --Send the output to port
               DAC_DATA<= conv_std_logic_vector(out_value ,16);
               state <= Start;
               when others =>
               state <= Start;
               end case;
     END PROCESS states;

end Behavioral;
```

Once VHDL code is developed, the ALTERA tool “Quartus II” is used to synthesize the digital circuit: like other kind of FPGA design tools expanding a new project consists of some fundamentals phases. The first step is the “Analysis and Synthesis” step, where the VHDL code is check and compiled (on the basis of the chosen device); if this step is completed without any error, it is possible to display how the development environment has placed components to realize the circuit which is able to satisfy the main requirements. Using different tools can lead to obtain not equals results, because of the use of different optimization algorithms. In this application the obtained circuit is depicted in the Figure 46, at register (RTL) level.

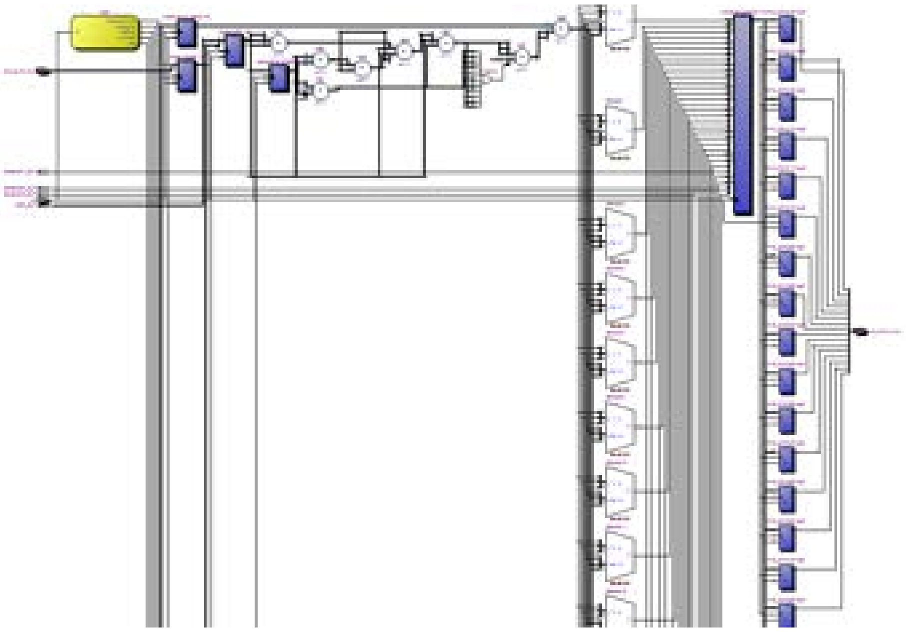
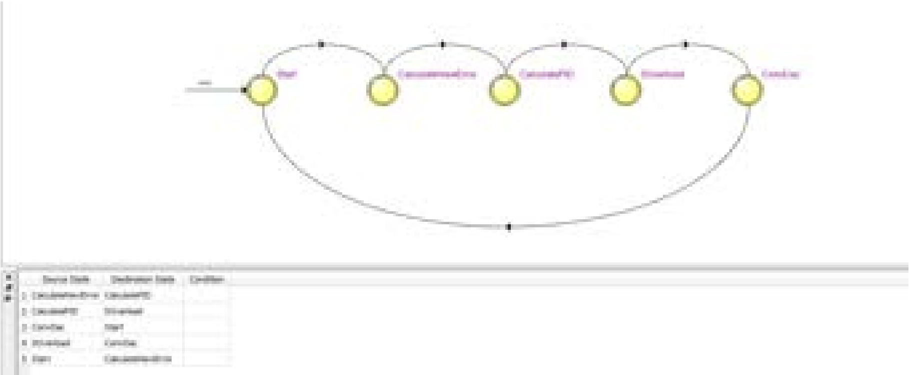


Figure 46: Register Transfer Level of main PID digital core implementation in FPGA



	Source State	Destination State	Condition
1.	CalculateNewError	CalculatePID	
2.	CalculatePID	[illegible]	
3.	ConvDac	Start	
4.	[illegible]	ConvDac	
5.	Start	CalculateNewError	

Figure 47: FSM visualization of PID digital core implementation in FPGA

At this stage, a compilation report is also available to check the FPGA total logic port usage and the maximum delay between input and output values in terms of clock edges. It is good rule to remember that in this phase this delays are evaluated just on the basis of the PID entity behavior and in the last phase the timing constraints of ports and real delays are considered.

3.2.3 Flow Design of Digital Architectures Using LabVIEW FPGA–Module

Alternatively to VHDL code description, a PID algorithm can be implemented maintaining the same National Instruments architecture adopted LabVIEW descriprion language [30]. As previously mentioned, not only can you use compactRIO FPGA for high–speed acquisition and generation, but also to implement several control algorithms on the FPGA. Single–point I/O with multichannel can be exploited, tunable PID or other control approaches realize deterministic control with loop rates up to hundreds of kilohertz.

DMA channels can be used to stream high–speed data between the FPGA and real–time hardware. To create a DMA buffer for streaming data, just select on FPGA target a new FIFO. Giving the FIFO structure a descriptive name and choosing target–to–host as the type, the real–time processor is able to read from the FIFO those values stored by the FPGA. This means that data should flow through this DMA FIFO from the FPGA target to the real–time host. Data type can also be set and FPGA FIFO depths.

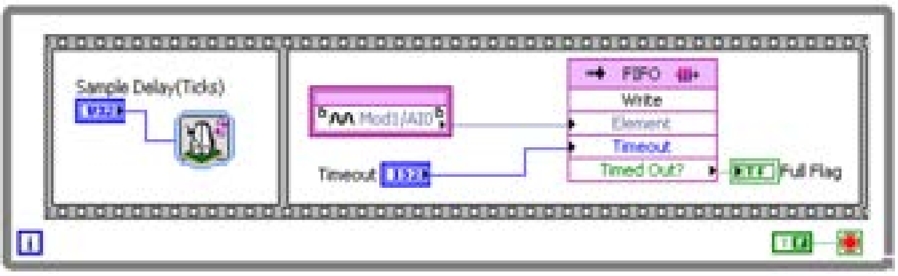


Figure 48: Simple MA transfer on One channel

It is fairly simple to put a DMA FIFO on a diagram. However, complexities arise when the default settings on the DMA transfer are not sufficient. If missing data points create a bug in your system, you must monitor the full flag on the FPGA and latch it when a fault occurs. Simply sampling this register from the host is not sufficient to catch quick transitions on that variable. Figure 49 shows various latching techniques on the timeout (full flag).

If you are receiving full flags, you need to either increase buffer size on the host, read larger chunks on the host, or read faster on the host. Keep in mind that many control and monitoring applications need only the most up–to–date data. Therefore losing data may not be an issue for a system as long as it returns the most recent data when called.

Another consideration for DMA transfer is using one DMA FIFO for multiple channels. Hybrid mode CompactRIO systems only have one DMA channel available. To pack multiple channels into one DMA FIFO, use an interleaving technique, and unpack using decimation on the host.

To read DMA channels from a realtime program, a reference to the FPGA VI or bitfile and FPGA target is needed to be specify. Then the modality to read a value from or write a value to a control or indicator in the FPGA VI on the FPGA target is set. This can be a trigger condition, sampling rate, or

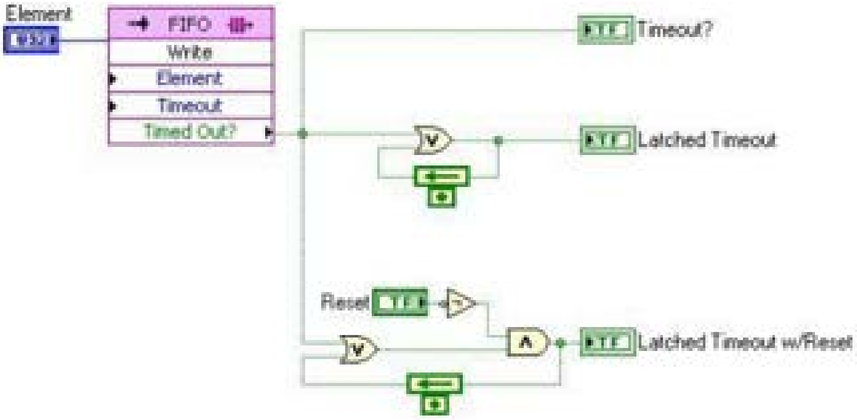


Figure 49: Example of No Latch, Simple Latch, and Latch with Reset

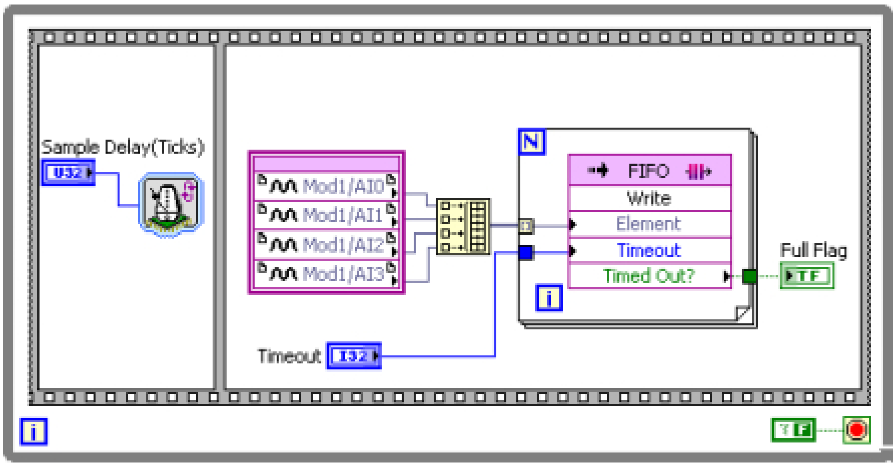


Figure 50: interleaved multichannel data stream

any other data or parameter set by a control or indicator in the FPGA VI. Invoking an FPGA interface method or action from a host VI on an FPGA VI is essential to implement the following operations: download, abort, reset, and run the FPGA VI on the FPGA target; wait for and acknowledge FPGA VI interrupts; read DMA FIFOs; and write to DMA FIFOs. The methods a user can choose from depend on the target hardware and the FPGA VI. It is necessary to wire the FPGA VI Reference input to view the available methods in the shortcut menu. On the realtime applicative we have to open a reference to the FPGA VI, set parameters, use the Invoke Node in a task loop to read waveform data, and close the FPGA reference to implement the simplest DMA read. However, as previously discussed, complexities can arise from buffer sizes, timeouts, and synchronization. Below is a simple real–time application that reads a waveform from the FPGA, performs an average calculation on the waveform and passes the data to a separate control loop that is running a PID loop to control a PWM output based on the waveform average. This type of application might be used to control a signal generator or laser that is tuned using a PWM input signal.

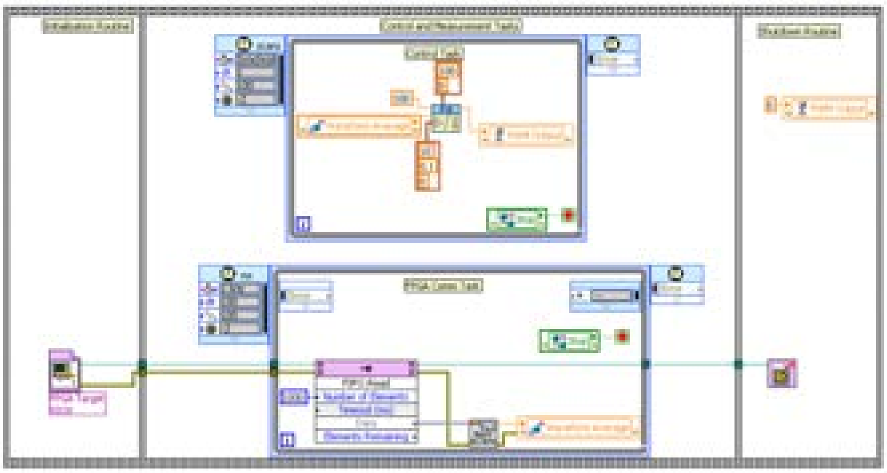


Figure 51: Simple DMA Read Using the FPGA Interface on the Real–Time Host

4 Conclusions

In this deliverable, a Simulink simulation environment has been used to check the performance of the designed digital controllers, for various values of the sampling time. Moreover, LabVIEW has been analyzed as potential solution to interface Simulink with real–time environments. This allows testing the controllers with some hardware devices, such as sensors and actuator, in the control loop. Although not certified for nuclear applications, this allows checking the methodological steps for the real–time prototyping of the controllers for non–nuclear industrial applications. For nuclear applications, more costly nuclear–certified softwares have to be considered. Such Simulink/LabVIEW simulation environment can be integrated in the next future with an experimental set–up, here proposed to show the benefit of the FPGA features.

References

[1] S. Di Gennaro and B. Castillo–Toledo, Comparative Study of Controllers for the Supervision, Control and Protection Systems in Pressurized Water Reactors of Evolutive Generation, Deliverable 2, PAR2010 Project, 2011.

[2] S. Di Gennaro and B. Castillo–Toledo, Performance Study of the Control Systems in the presence of Faults and/or Reference Accidents in Pressurized Water Reactors of Evolutive Generation, Deliverable 2, PAR2010 Project, 2011.

[3] S. Di Gennaro, B. Castillo–Toledo, and F. Memmi, Study, Design and Realization of Supervisory, Control and Protection Systems for Performance and Safety Improvements of Novel Nuclear Plants, Deliverable 1, PAR2011 Project, 2012.

[4] Areva, *U.S. EPR Nuclear Plant – The Path of Greatest Certainty*, Areva 2007.

[5] Areva, *Design Control Document "U.S. EPR Final Safety Analysis Report"*, available at US–NRC, 2008.

[6] B. Castillo–Toledo, M. Cappelli, S. Di Gennaro, and M. Sepielli, Pressurizer Pressure Control in PWRs of New Generation, *Proceedings of the* 8^{th} *Internatonal Topical Meeting on Nuclear Plant Instrumentation, Control, and Human–Machine Interface Technologies*, San Diego, CA, USA, 2012.

[7] A. Crabtree, M. and Siman–Tov, *Thermophysical Properties of Saturated Light and Heavy Water for Advanced Neutron Source Applications*, Oak Ridge National Laboratory, 1993.

[8] Cranfor, III et al, Device and method for measuring temperature of a liquid contained in a pressurizer vessel, United States Patent N. 5426677, June 20, 1995.

[9] F. Champomier, *PCSR – Sub–chapter 5.4 – Components and Systems Sizing*, UKEPR–0002–054 Issue 02, AREVA NP & EDF, June 19, 2009.

[10] J. Freixa, and A. Manera, Verification of a TRACE EPRTM Model on the Basis of a Scaling Calculation of an SBLOCA ROSA Test, *Nuclear Engineering and Design*, Vol. 241, pp. 888–896, 2011.

[11] Hou, D. Lin, M. Xu, Z.–H. Yang, Z. Y.-h. Chen, Expansion of Relap5 Control and Protection Simulation Functions with Simulink, *Nuclear Power Engineering*, Hidongli Gongcheng, Cina, Vol. 28, Part 6, pp. 112–115, 2007.

[12] N. Larsen, *Simulation Model of a PWR Power Plant*, Riso National Laboratory, DK–4000, Roskilde, Denmark, 1987.

[13] J. Lienhard, and J. Lienhard, *A Heat Transfer Textbook*, 3^{rd} Edition, Phlohiston, Press Cambridge University, 2003.

[14] Meng Lin, Dong Hou, Zhihong Xu, etc., System Simulation of Nuclear Power Plant by Coupling Relap5 and Matlab/Simulink., 14th International Conference on Nuclear Engineering, Miami, Florida, USA, pp. –, 2006.

[15] Mitsubishi Heavy Industries Ltd, *Design Control Document for the US–APWR*, Chapter 7 – Instrumentation and Controls, MUAP–DC007, Revision 3, March 2011.

[16] N. Muellner, M. Lanfredini, Bases for Setting up a Thermal–Hydraulic Model of a Generic PWR Pressurizer, including Controls, Agreement N.IN.E S.r.l. – University of L'Aquila, 2011.

[17] G. Petrangli, *Nuclear Safety*, 2005.

[18] Z. Szabó, P. Gárspár, and J. Bokor, Reference Tracking of Wiener Systems Using Dynamic Inversion, *Proceeding International Symposium on Intelligent Control*, Limassol, Cyprus, on CD, 2005.

[19] N. E. Todreas, and M. S. Kazimi, *Nuclear Systems I – Thermal Hydraulic Fundamentals*, Hemisphere Publishing Cooperation, 1990.

[20] N. E. Todreas, and M. S. Kazimi, "Nuclear Systems II – Elements of Thermal Hydraulic Design", Hemisphere Publishing Cooperation, 2001.

[21] L. S. Tong, and J. Weisman, *Thermal Analysis of Pressurized Water Reactors*, 3^{rd} Edition, ANS, 1996.

[22] J. L. Tylee, Bias Identification in PWR Pressurizer Instrumentation using Generalized Likehood–Ratio Techquique, *Proceedings of the 1981 American Control Conference*, Charlottesville, Virginia USA, Tech. Report N. CONF–810605–1, 1981.

[23] UK–EPR, *Fundamental Safety Overview, Vol. 2: Design and Safety, Chapter E: the Reactor Coolant System and Related Systems.*

[24] Research Reactor Group, Research reactor Modernization and Refurbishment, *IAEA* Progress Report, 2009.

[25] E. Bachmach, O.Siora, V.Tokarev, S. Reshetytsky, V. Kharchenko, and V. Bezsalyi, FPGA–Based Technology and Systems for I&C of Existing and Advanced Reactors, *IAEA–CN–164–7S04*, 2010.

[26] F. Memmi, O. Aronica, R. Bove, M. Cappelli, L. Falconi, M. Palomba, E. Santoro, and M. Sepielli, A High Operability Supervisory Digital System for TRIGA–Type Research Reactors, *Proceedings for the European Research Reactor Conference – RRFM*, 2010.

[27] CompactRIO Developers Guide, *Recommended LabVIEW Architectures and Development Practice for Machine Control Applications*, National Instruments, 2011.

[28] F. Memmi, R. Bove, M. Cappelli, L. Falconi, M. Palomba, E. Santoro, and M. Sepielli, A Preliminary User–Friendly, Digital Console for the Control Room Parameters Supervision in Old–Generation Nuclear Plants, *Proceedings for the International Congress on Advances in NPPs – ICAPP*, 2012.

[29] F. Memmi, R. Bove, M. Cappelli, L. Falconi, M. Palomba, and M. Sepielli, The Role of FPGA–based Architectures in the Control Room Modernization Process: Preliminary Results of a Case–Study, *ICONE20*, 2012.

[30] Press LCC, *LabVIEW Advanced Programming Techniques*, 2001.