ThorCon CHD System Model

QVIST
CONSULTING LIMITED
LONDON / STOCKHOLM

Team introduction

Dr Staffan Qvist, PhD Nuc. Eng. UC Berkeley (13')

- Chair of IAEA reactor shutdown systems study
- Inventor of ARC passive safety systems and lead core designer/developer for SEALER LFR,
- *Project manager for Nuc. Dev Projects*

Dr Carl Hellesen, PhD Nuc. Eng. Uppsala University (10')

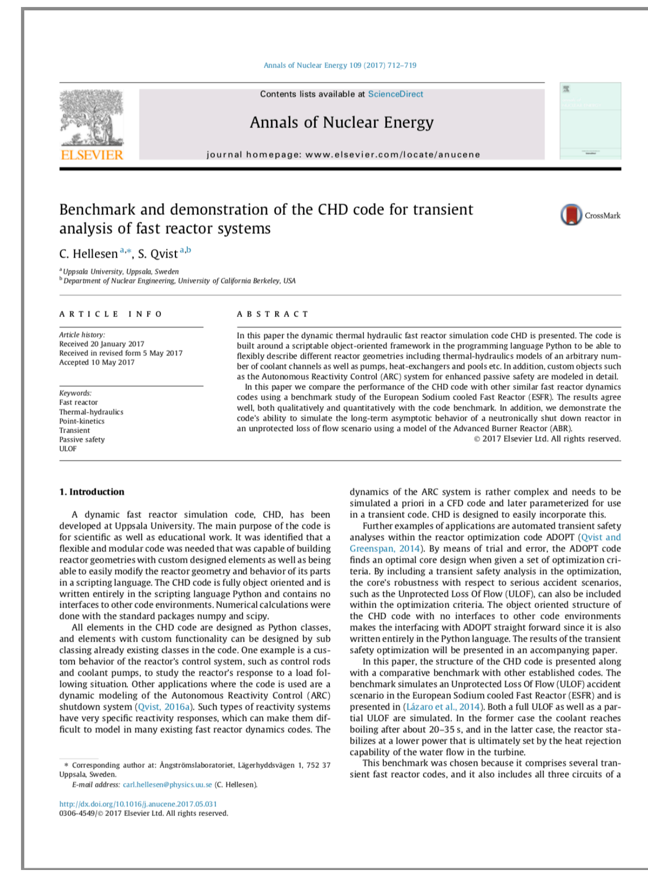
- Lead developer of CHD code
- Physicist and lecturer at Uppsala University
- *Systems code development expert*

Dr Ryan Bergmann, PhD Nuc. Eng. UC Berkeley (14')

- Lead developer of WARP GPU Monte-Carlo code
- Physicist at Paul Scherrer Institute (PSI), Switzerland
- *Neutronics & Monte-Carlo Code Expert*

CHD Code Introduction

- CHD is a multi-channel point-kinetics based dynamic reactor simulation code
- Conceptually similar to codes such as SAS4A/SASSYS-1, THACOS, SSC-L and MAT5-DYN
- Fully object oriented and is written entirely in Python, with numerical calculations done with the standard packages numpy and scipy.
- Extremely flexible and customisable, allowing for rapid addition of complex components
- Originally written for fast reactor analysis, now a fully capable MSR-simulation code further developed *specifically* to model the ThorCon plant
- Validated/benchmarked against the available MSRE and EBR-II experimental results, as well as code-to-code benchmarking including the large ESFR benchmark.



Annals of Nuclear Energy 109 (2017) 712–719

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Benchmark and demonstration of the CHD code for transient analysis of fast reactor systems

C. Hellesen [a,*], S. Qvist [a,b]

[a] Uppsala University, Uppsala, Sweden
[b] Department of Nuclear Engineering, University of California Berkeley, USA

ARTICLE INFO

Article history:
Received 20 January 2017
Received in revised form 5 May 2017
Accepted 10 May 2017

Keywords:
Fast reactor
Thermal-hydraulics
Point-kinetics
Transient
Passive safety
ULOF

ABSTRACT

In this paper the dynamic thermal hydraulic fast reactor simulation code CHD is presented. The code is built around a scriptable object-oriented framework in the programming language Python to be able to flexibly describe different reactor geometries including thermal-hydraulics models of an arbitrary number of coolant channels as well as pumps, heat-exchangers and pools etc. In addition, custom objects such as the Autonomous Reactivity Control (ARC) system for enhanced passive safety are modeled in detail.

In this paper we compare the performance of the CHD code with other similar fast reactor dynamics codes using a benchmark study of the European Sodium cooled Fast Reactor (ESFR). The results agree well, both qualitatively and quantitatively with the code benchmark. In addition, we demonstrate the code's ability to simulate the long-term asymptotic behavior of a neutronically shut down reactor in an unprotected loss of flow scenario using a model of the Advanced Burner Reactor (ABR).

1. Introduction

A dynamic fast reactor simulation code, CHD, has been developed at Uppsala University. The main purpose of the code is for scientific as well as educational work. It was identified that a flexible and modular code was needed that was capable of building reactor geometries with custom designed elements as well as being able to easily modify the reactor geometry and behavior of its parts in a scripting language. The CHD code is fully object oriented and is written entirely in the scripting language Python and contains no interfaces to other code environments. Numerical calculations were done with the standard packages numpy and scipy.

All elements in the CHD code are designed as Python classes, and elements with custom functionality can be designed by sub classing already existing classes in the code. One example is a custom behavior of the reactor's control system, such as control rods and coolant pumps, to study the reactor's response to a load following situation. Other applications where the code is used are a dynamic modeling of the Autonomous Reactivity Control (ARC) shutdown system (Qvist, 2016a). Such types of reactivity systems have very specific reactivity responses, which can make them difficult to model in many existing fast reactor dynamics codes. The dynamics of the ARC system is rather complex and needs to be simulated a priori in a CFD code and later parameterized for use in a transient code. CHD is designed to easily incorporate this.

Further examples of applications are automated transient safety analyses within the reactor optimization code ADOPT (Qvist and Greenspan, 2014). By means of trial and error, the ADOPT code finds an optimal core design when given a set of optimization criteria. By including a transient safety analysis in the optimization, the core's robustness with respect to serious accident scenarios, such as the Unprotected Loss Of Flow (ULOF), can also be included within the optimization criteria. The object oriented structure of the CHD code with no interfaces to other code environments makes the interfacing with ADOPT straight forward since it is also written entirely in the Python language. The results of the transient safety optimization will be presented in an accompanying paper.

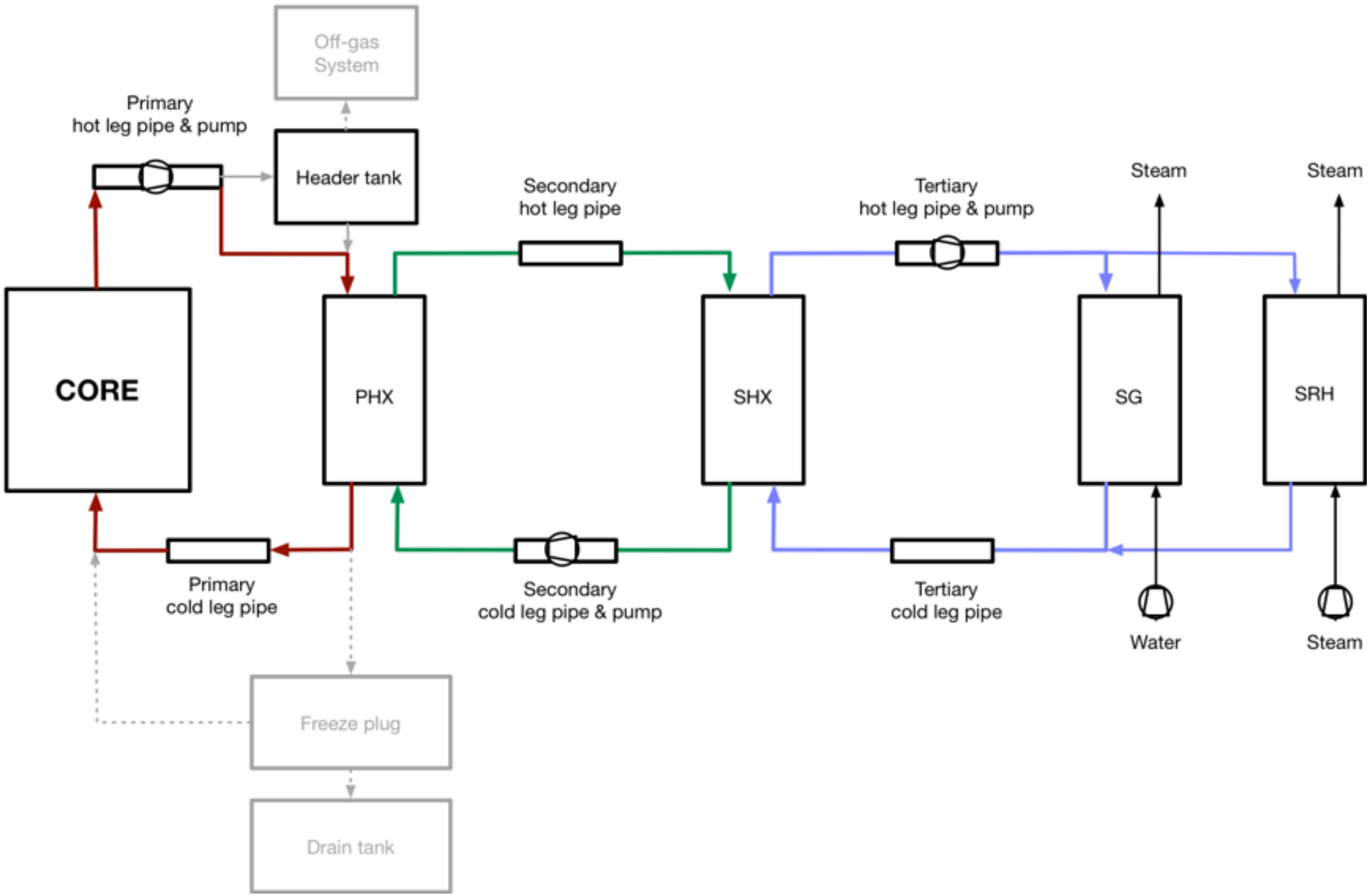
In this paper, the structure of the CHD code is presented along with a comparative benchmark with other established codes. The benchmark simulates an Unprotected Loss Of Flow (ULOF) accident scenario in the European Sodium cooled Fast Reactor (ESFR) and is presented in (Lázaro et al., 2014). Both a full ULOF as well as a partial ULOF are simulated. In the former case the coolant reaches boiling after about 20–35 s, and in the latter case, the reactor stabilizes at a lower power that is ultimately set by the heat rejection capability of the water flow in the turbine.

This benchmark was chosen because it comprises several transient fast reactor codes, and it also includes all three circuits of a

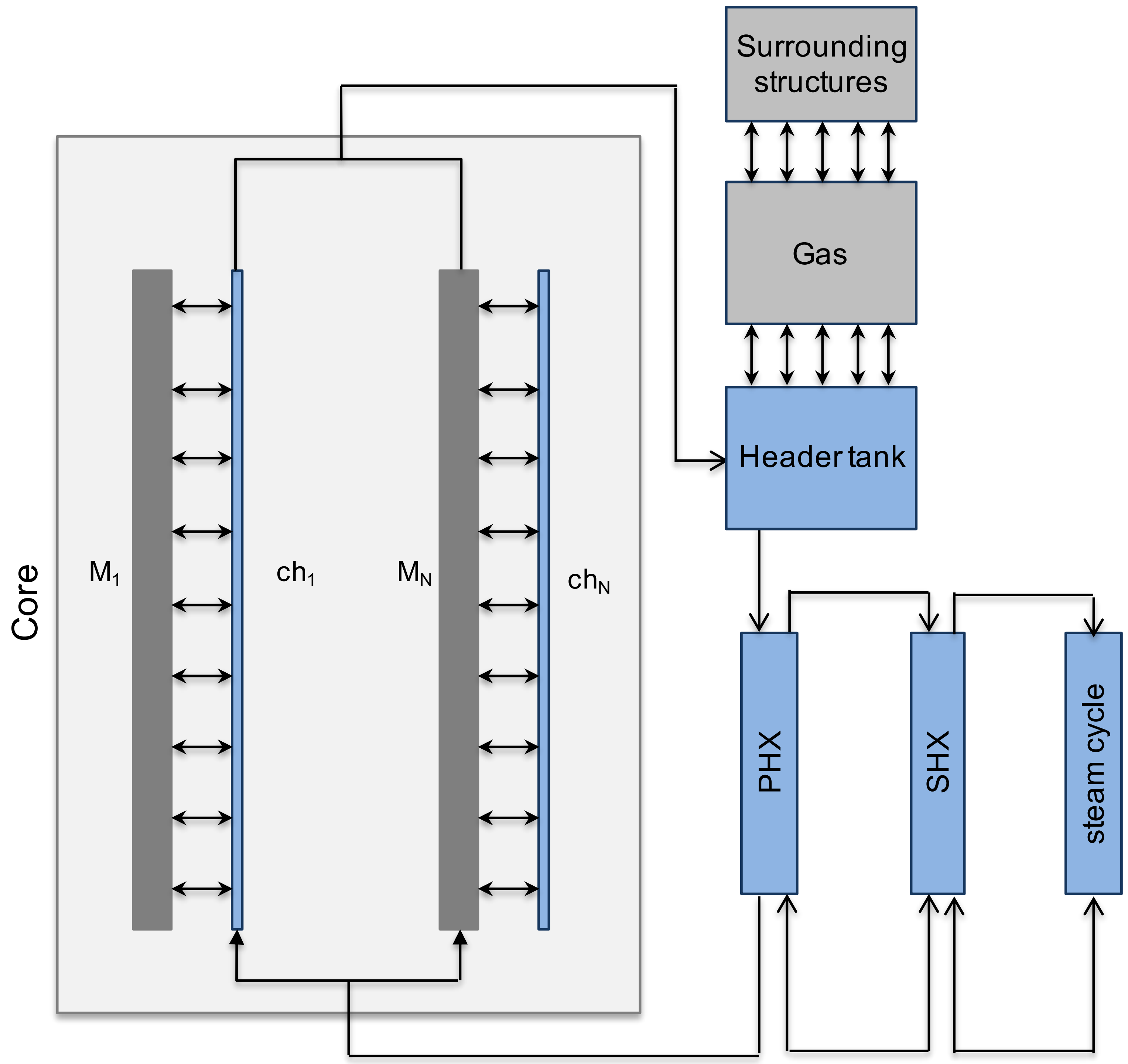
* Corresponding author at: Ångströmslaboratoriet, Lägerhyddsvägen 1, 752 37 Uppsala, Sweden.
E-mail address: carl.hellesen@physics.uu.se (C. Hellesen).

http://dx.doi.org/10.1016/j.anucene.2017.05.031
0306-4549/© 2017 Elsevier Ltd. All rights reserved.

CHD Code ThorCon Model

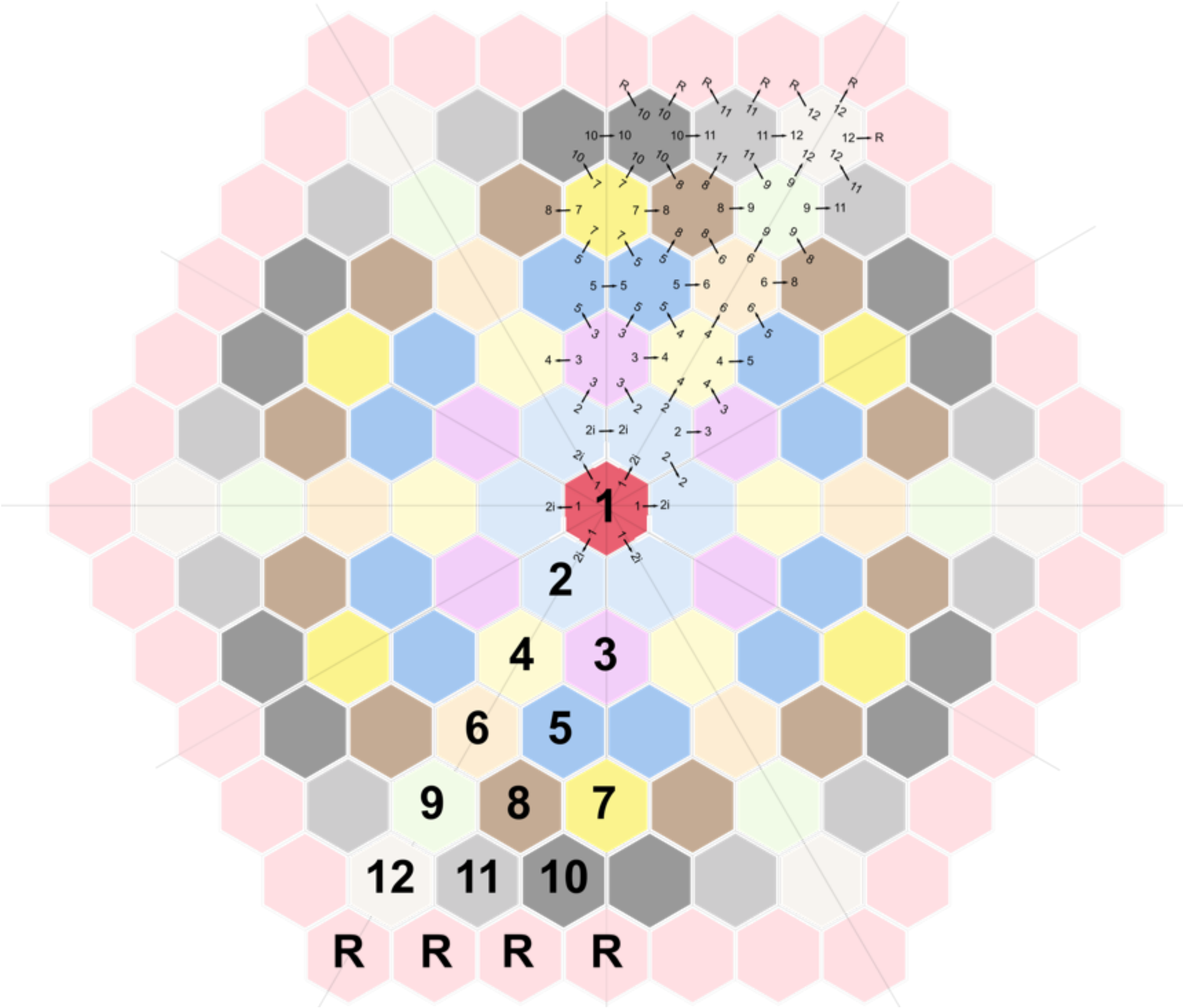


CHD Code ThorCon Model

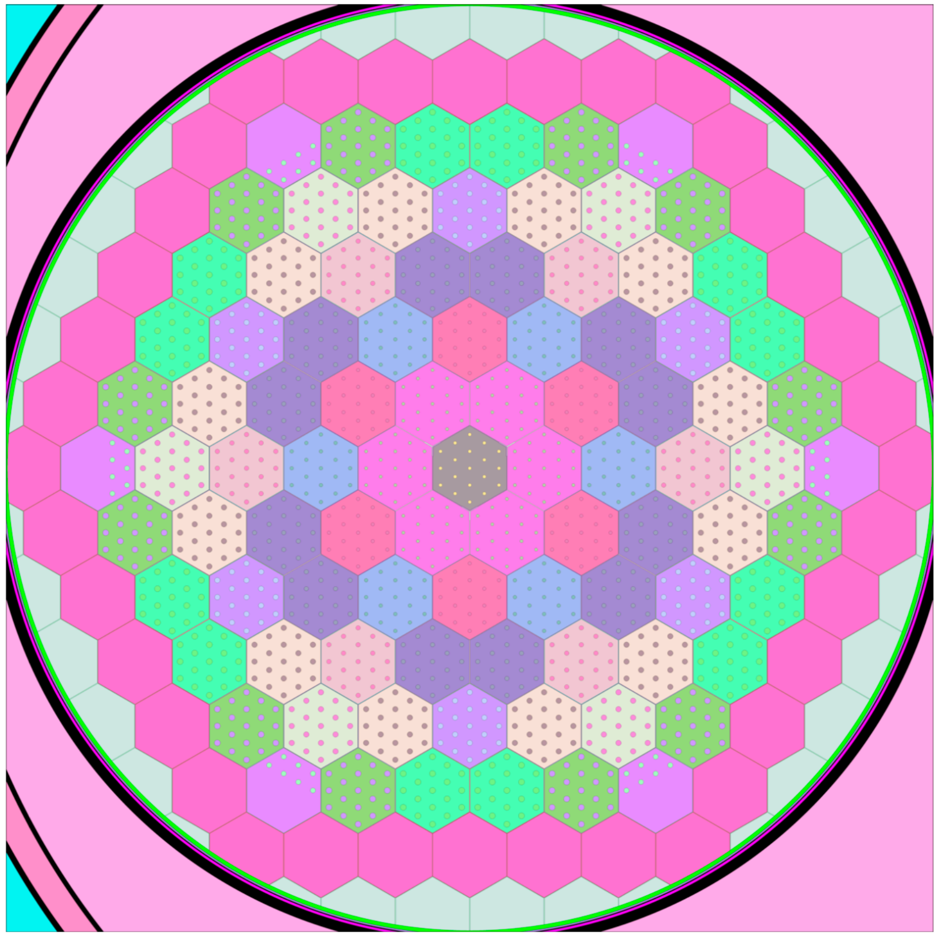


ThorCon Core Representation

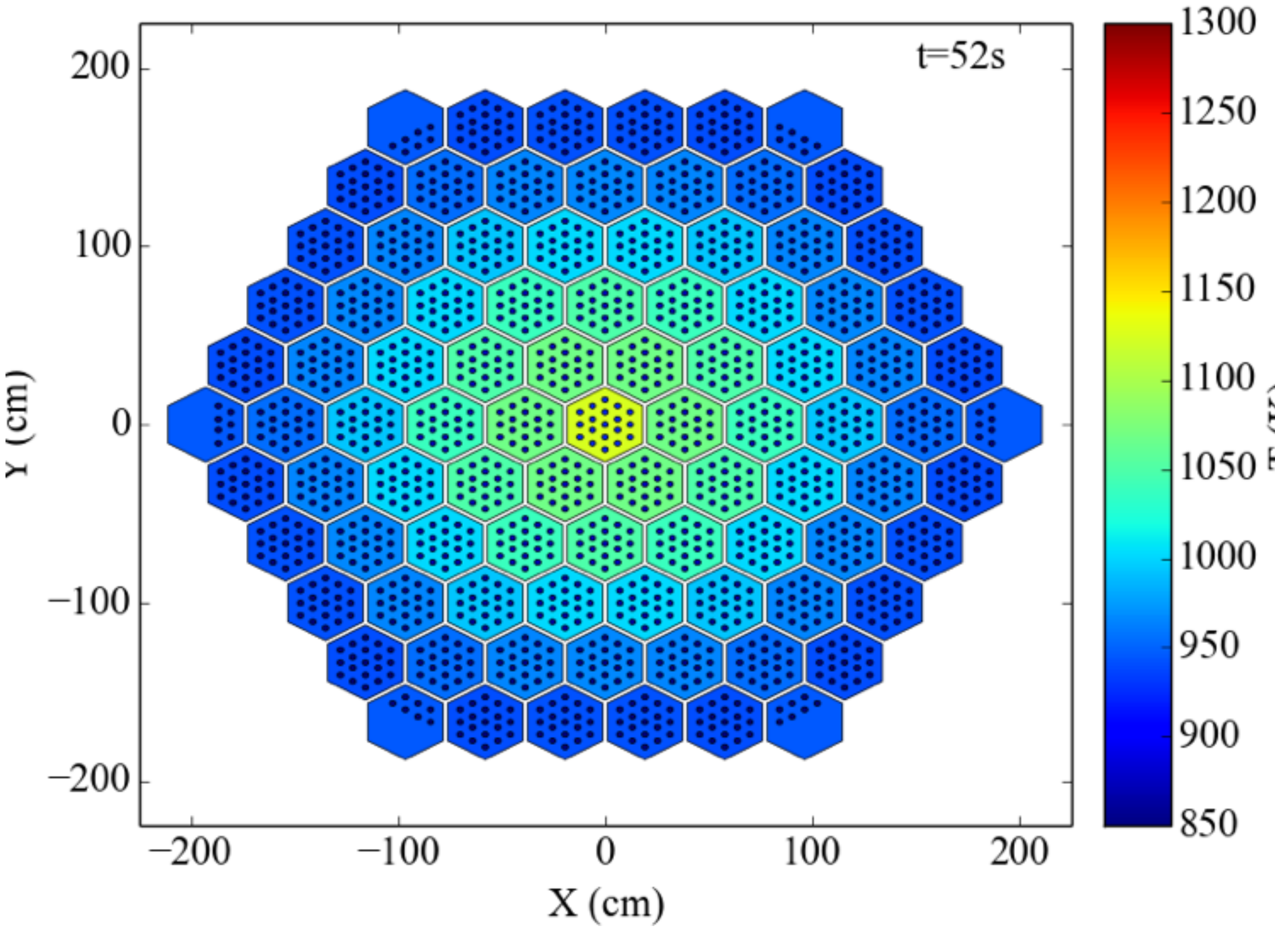
- The core is modelled using 1/12th symmetry, with each gap section between each log modelled as a separate channel and all the holes in each distinct log are also treated as separate channels.
- There are 41 separate parallel channels in the core.
- Delayed neutron precursors are tracked throughout the primary loop. All core channels transport and produce precursors separately.
- Decay heat is modelled using a 23-group structure.



ThorCon Core Representation



Serpent model used to calculate reactivity feedback coefficients



Example of CHD output (temperature distribution during transient)

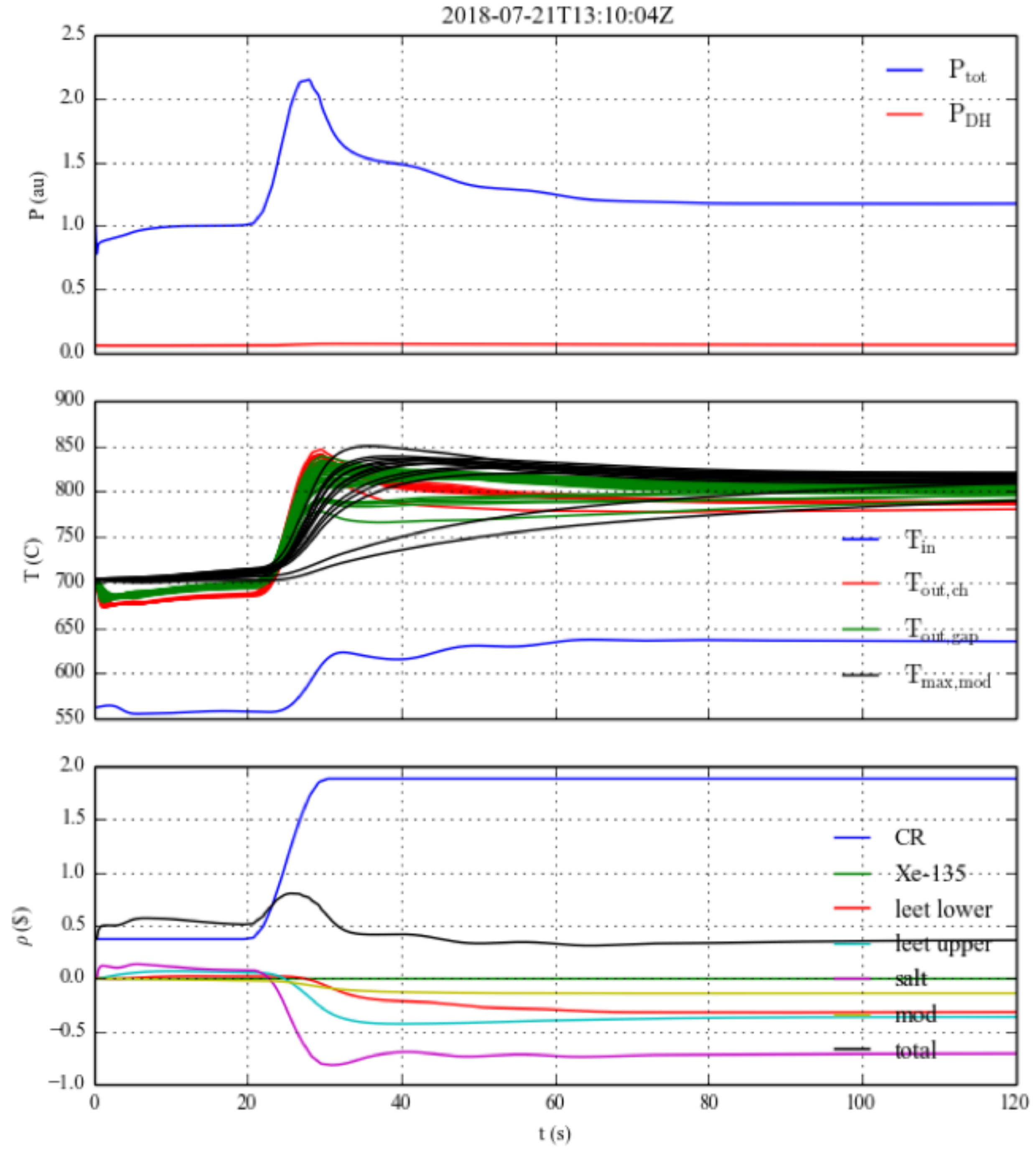
Transient Simulation Results

QVIST CONSULTING LIMITED
LONDON / STOCKHOLM

Transient #1 - Reactivity Insertion

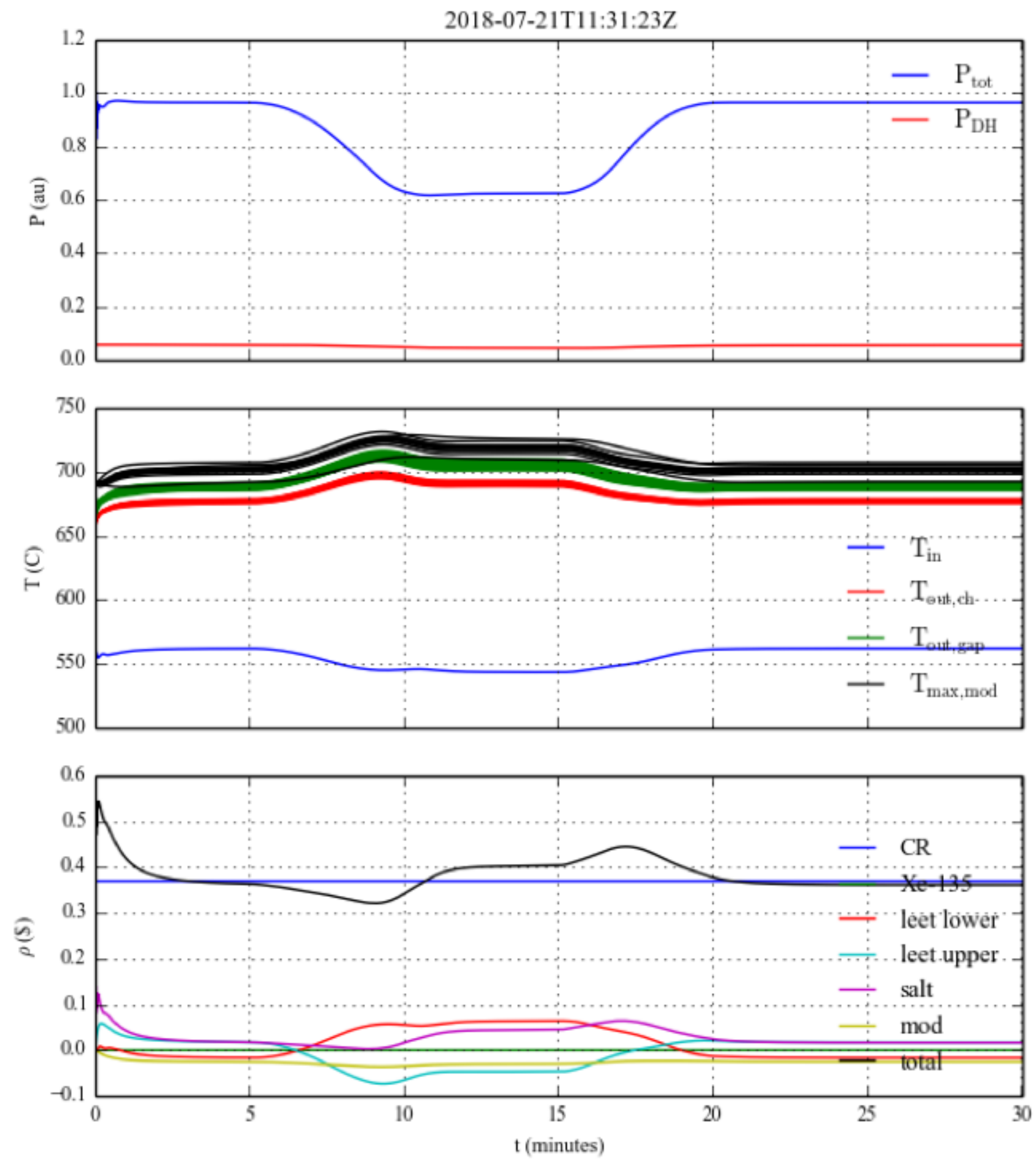
- 400 pcm reactivity inserted during 10 s
  - Power spikes at 210%
  - Settles at 115% after 40 s

- No separate channel or graphite log above 850 C



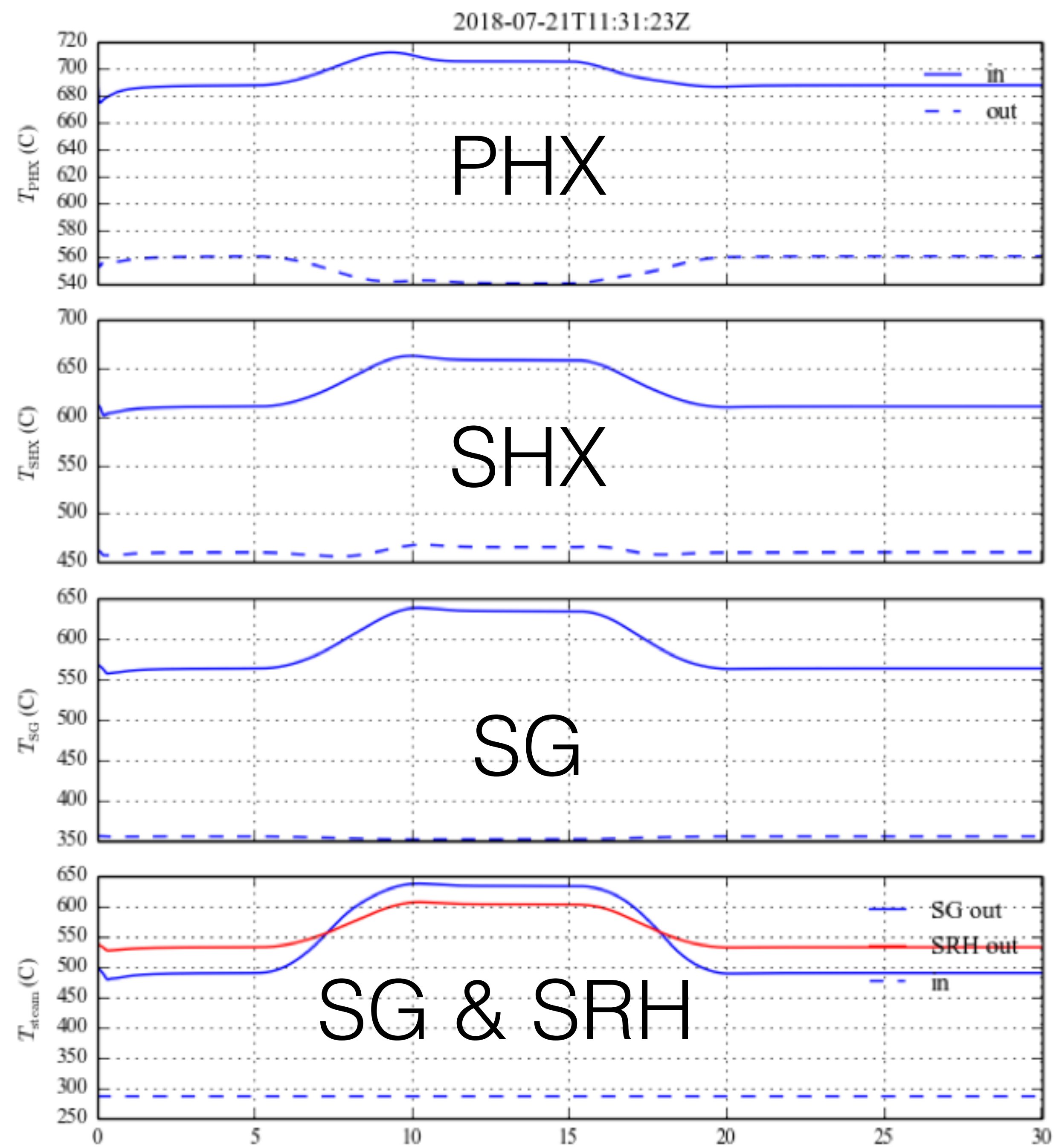
Transient #2 - Flow&Power Ramp (1/2)

- All salt loops and feed water flow reduced by 50% and ramped up again by 50%
  - 300 s ramp time(10%/min)
- Power can be controlled using only flow rates in loops
  - No control rods required for load following
- In this example, all loops ramped uniformly
  - Control algorithms to adjust flows individually for constant steam temperatures will be develepd using ThorCon model



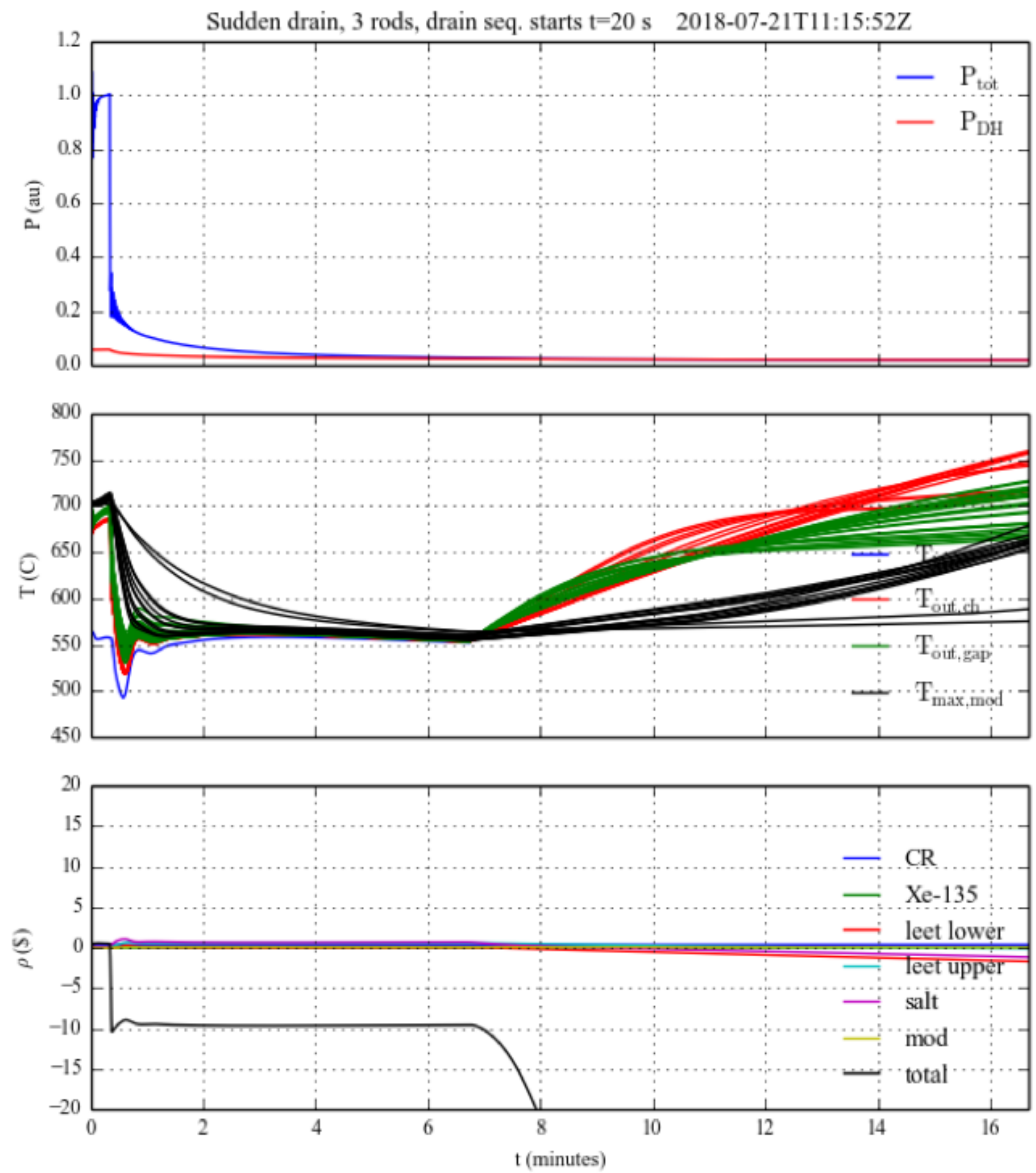
Transient #2 - Flow&Power Ramp (2/2)

- All salt loops and feed water flow reduced by 50% and ramped up again by 50%
  - 300 s ramp time(10%/min)
- Power can be controlled using only flow rates in loops
  - No control rods required for load following
- In this example, all loops ramped uniformly
  - Control algorithms to adjust flows individually for constant steam temperatures will be developd using ThorCon model



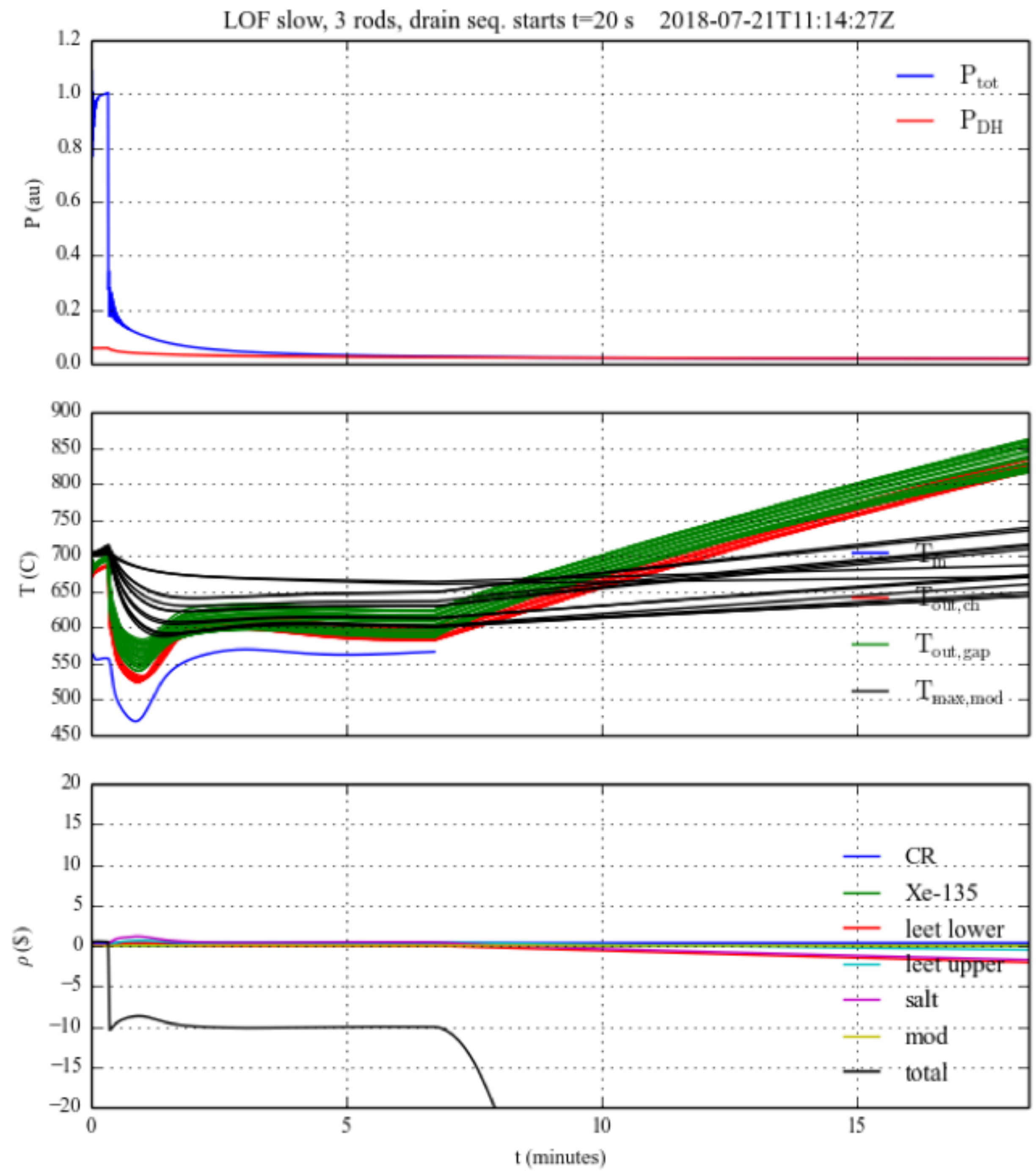
Transient #3 - Fukushima-Eq. Scenario

- At time of earthquake, a controlled drain is initiated
  - SCRAM shuts down the fission power
- Power and cooling available for a limited time after SCRAM
  - AC power, batteries, diesels, …
- At drain time, the reactor is put into a safe state with salt in drain tank
  - Salt temperatures max at 750 C



Transient #4 - Instant Station Blackout

- At time of earthquake, a drain is initiated
  - SCRAM shuts down the fission power
- All power and cooling lost directly after SCRAM
  - Worse than Fukushima
  - Core is initially cooled by natural convection
- At drain time, the reactor is put into a safe state with salt in drain tank
  - Salt temperatures max at 850 C



Transient #5 - Instant Unprotected Station Blackout

- All safety systems fail
  - No shutdown rods
  - No backup power
  - No cooling
  - Much worse than Fukushima

- Fission power shut down from negative feedbacks
  - Passive natural circulation provides initial cooling

- At drain time, the reactor is put into a safe state with salt in drain tank
  - Salt temperatures max at 1000 C
  - 0.25% of steel creep lifetime used up