The Design and Performance of the Real-time Software Architecture for the ITER Radial Neutron Camera

N. Cruz[1], B. Santos[1], A. Fernandes[1], P.F. Carvalho[1], J. Sousa[1], B. Gonçalves[1], M. Riva[2], C. Centioli[2], D. Marocco[2], B. Esposito[2], C.M.B. Correia[3], and R.C. Pereira[1]

nunocruz@ipfn.tecnico.ulisboa.pt

[1] *Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal.*

[2] *ENEA C. R. Frascati, Dipartimento FSN, via E. Fermi 45, 00044 Frascati (Roma), Italy.*

[3] *LIBPhys-UC, Department of Physics, University of Coimbra, P-3004 516 Coimbra, Portugal*

ACKNOWLEDGMENT

The work leading to this publication has been funded partially by Fusion for Energy under the Contract F4E-FPA-327. IST activities also received financial support from "Fundação para a Ciência e Tecnologia" through project UID/FIS/50010/2013. This presentation reflects the views only of the authors, and Fusion for Energy cannot be held responsible for any use which may be made of the information contained therein.

Radial Neutron Camera Diagnostic

The Radial Neutron Camera (RNC) diagnostic is a neutron detection system with multiple collimators aiming at characterizing the neutron emission that will be produced by the ITER tokamak:

- a primary role for basic and advanced plasma control measurements
- backup for system machine protection measurements.

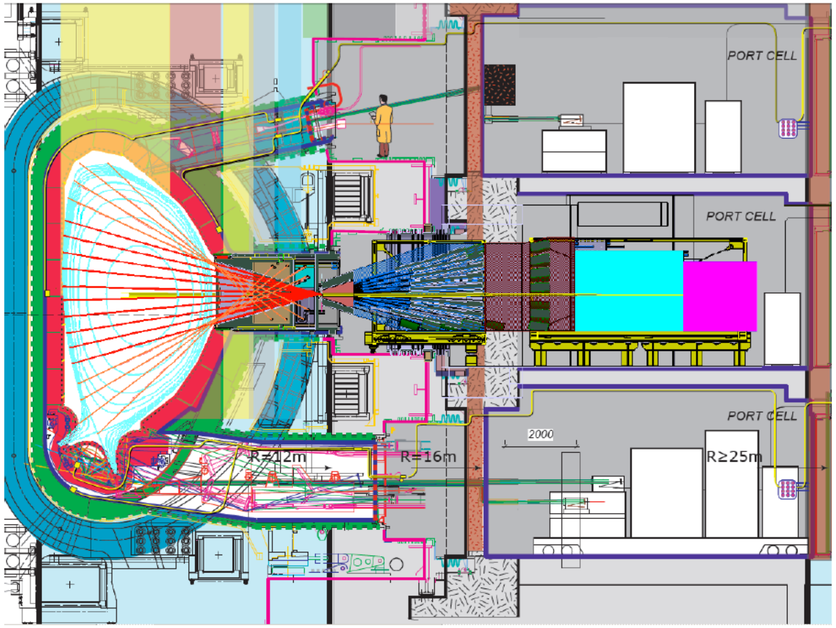
RNC diagnostic DatAcq needs

- acquire, process and store huge amounts of data per ITER discharge at high peak rates
- calculate real time measurements (neutron emissivity profile) on millisecond time scale

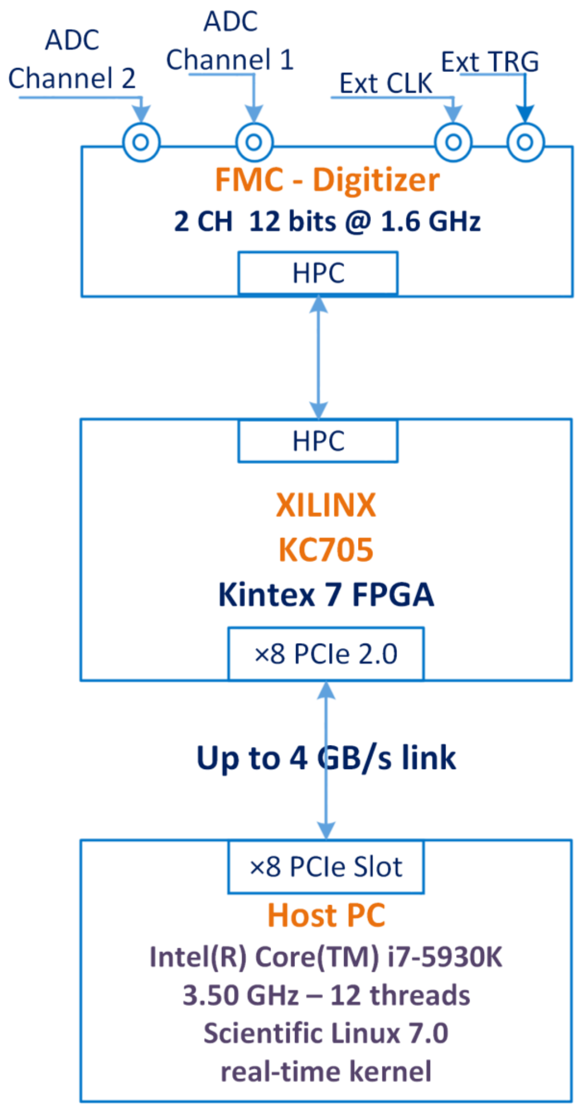
Technical challenges to the hardware, real-time firmware and software architecture design.

For the RNC system level design phase the following real time data processing algorithms were developed and tested:

- real time pulse processing (FPGA and host PC)
- real time calculation of the neutron emissivity radial profile
- real time pulse data compression block;



Prototype Architecture



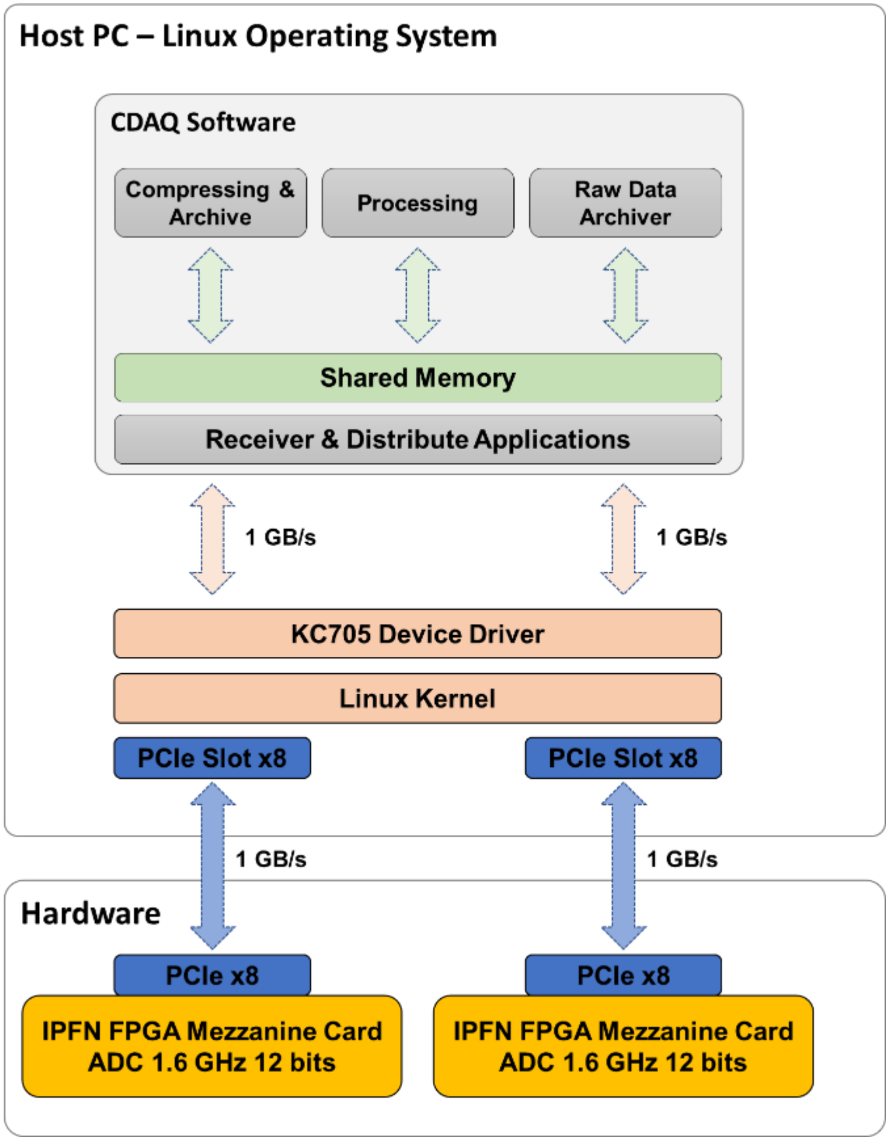
Although ATCA or PXIe architectures are planned to be used for the complete system, the decision is still pending, based on the performance test results, cubicle space optimization and reduction of unnecessary costs.

Thus, the prototype architecture (PCIe bus) was decided based on a cost effective solution capable of deploying the necessary performance tests, software architecture tools and algorithms.

Prototype hardware and software specifications:

- Real-time OS: Scientific Linux 7.0 with RT kernel
- In-house developed device drivers
- Host PC running on Intel(R) Core(TM) i7-5930K, 64 GB memory and 256 GB SSD;
- Xilinx KC705 Evaluation board (Kintex 7 FPGA);
- In-house developed analogue input with 2 ADC 12-bit @ up to 1.6 GHz FPGA Mezzanine Card (FMC-AD2-1600).

Software Architecture



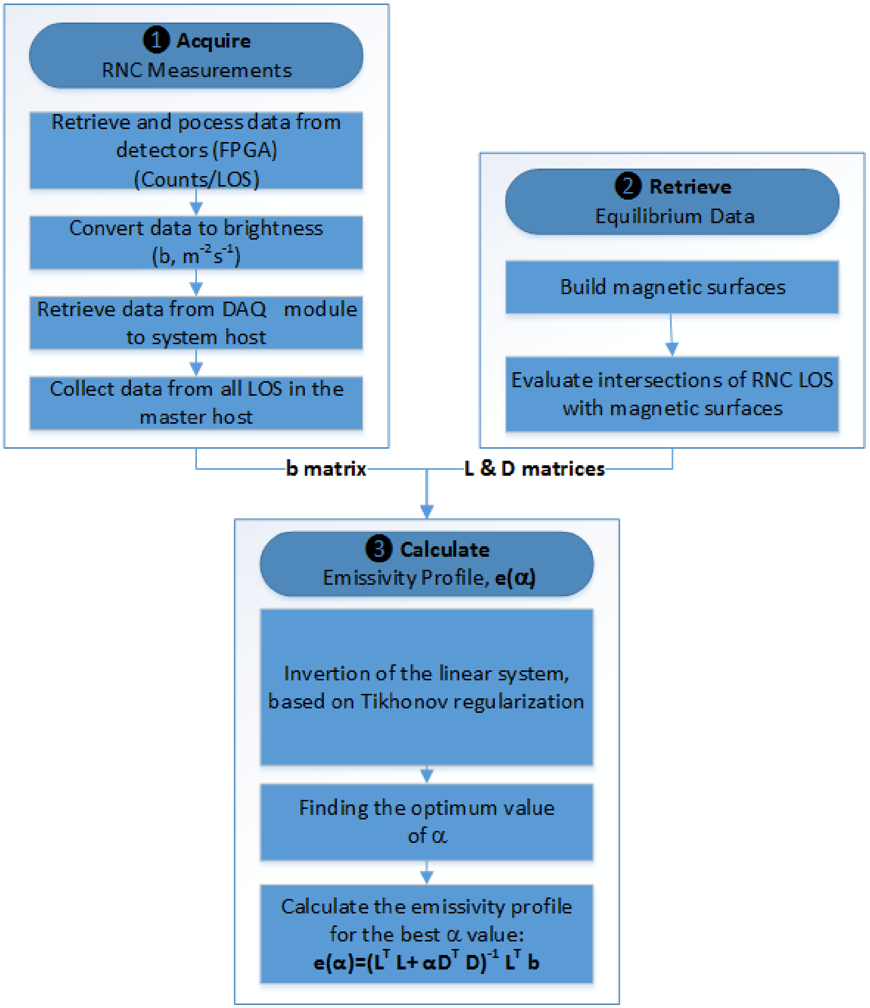
Main software modules:
- Linux device driver;
- Data receiver & distribution;
- Pulse processing for particle energy spectrum construction;
- Data compressing & archiving;
- Raw data archiver (for testing and validation).

The use of a shared memory permits that several clients can access in real time to the most recent pulse data.

Each software module run in dedicated isolated cores for real time & optimal system performance:
- Operating System - 2 logical cores
- Device Driver - 1 logical core per KC705
- Data receiver - 1 logical core per KC705
- Pulse processing - 3 cores per channel
- Data compression - up to 5 cores per KC705*

* B. Santos et al. P1.504, this conference.

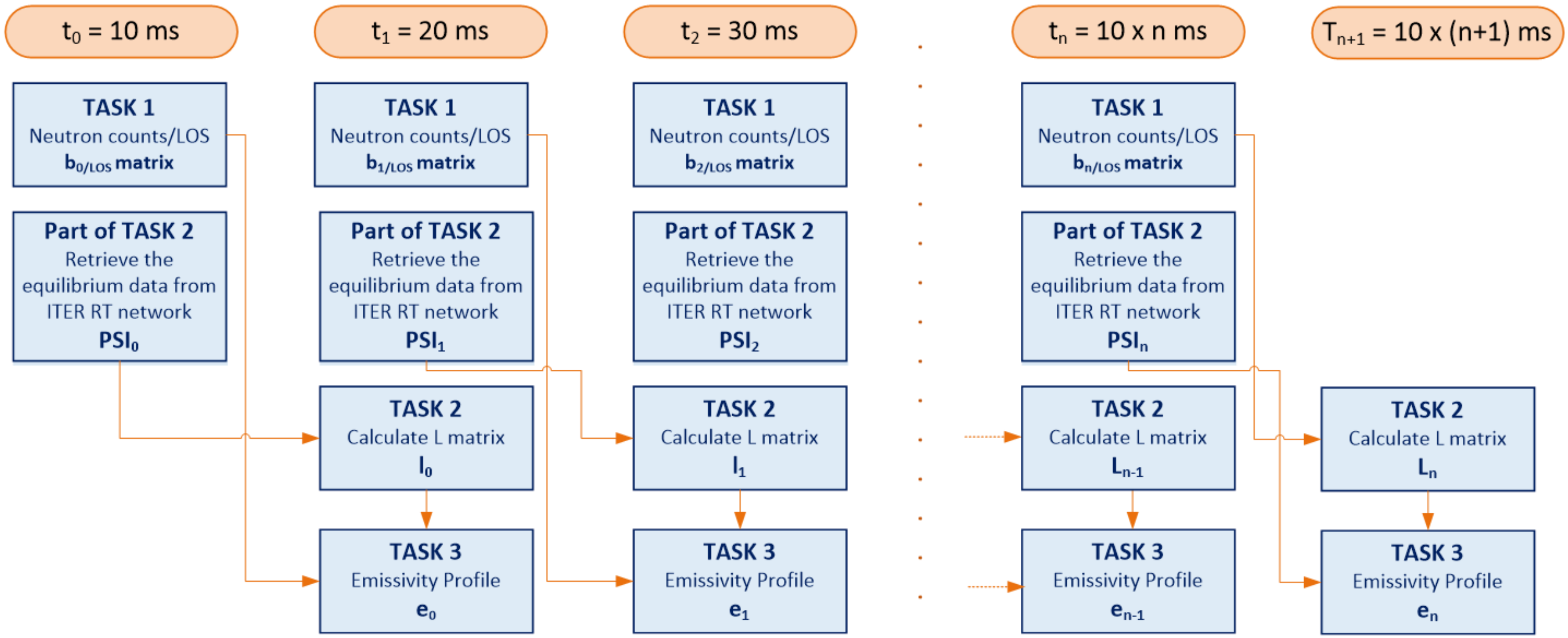
Emissivity Profile Measurement Data Processing



To obtain the real time measurement of the neutron emissivity profile 3 tasks were identified:

- Task 1 - Acquire and process neutron detector pulses;
- Task 2 - Retrieve and calculate the necessary inputs from the plasma equilibrium data;
- Task 3 - Calculate the neutron emissivity profile using Tikhonov inversion method.

Real Time Control Cycle



Tasks distribution to perform the control cycle under 10 ms:

- Task 1 must run before any other for the complete period of a control cycle;
- Task 2 uses data retrieved in the previous control cycle and calculates the necessary inputs from the plasma equilibrium data in the present control cycle;
- Task 3 calculate the neutron emissivity profile using all the available data from previous and present control cycle.
- The 3 tasks run in parallel using different CPUs.

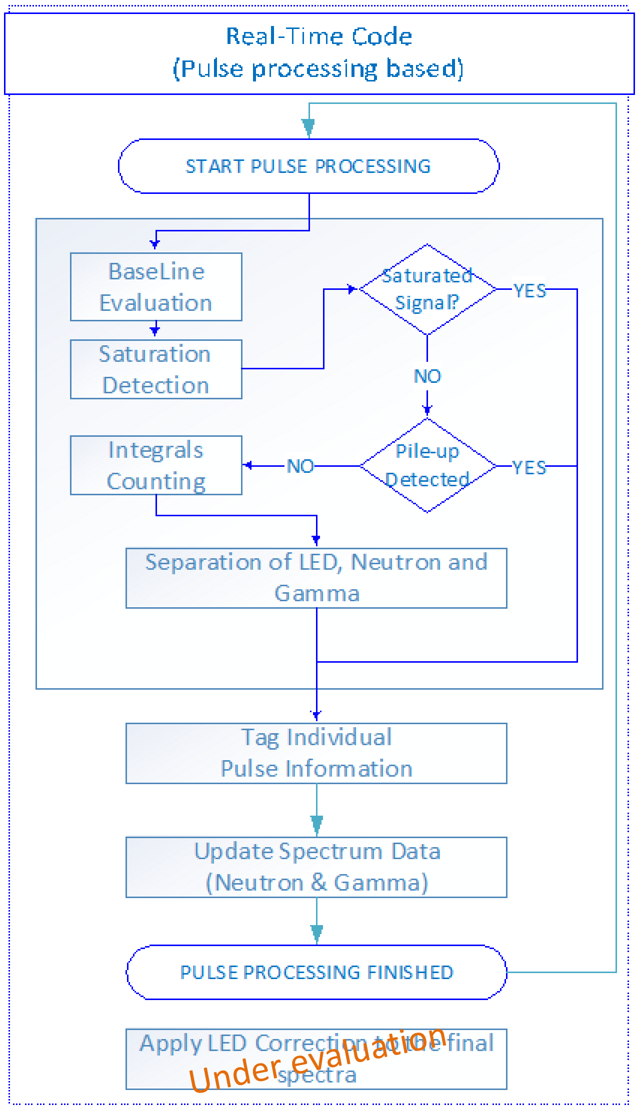
Pulse Processing

The real time pulse processing includes:

- Baseline evaluation
- Saturation detection
- Pile up detection
- Signal integration (energy calculation)
- Signal and particle separation

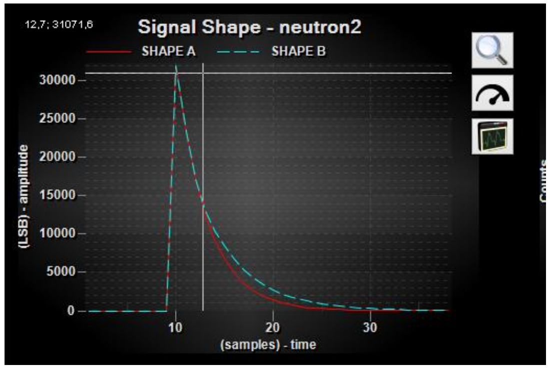
Real time spectrum calibration:

- The application of LED correction at the end of the control cycle to calibrate the energy spectra in real time is under evaluation.
- In former offline pulse processing, LED correction has been applied to each pulse individually.
- Pile-up and saturated signals detection is used for final spectrum count correction.

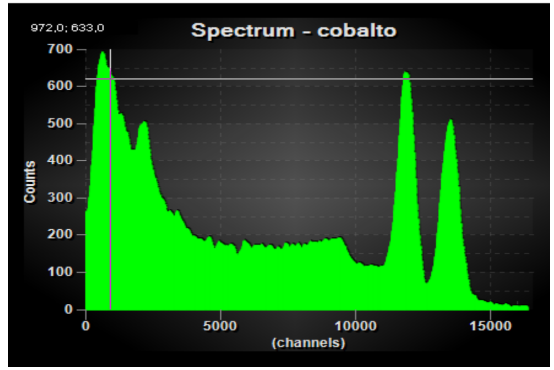


Validation
Pulse Processing

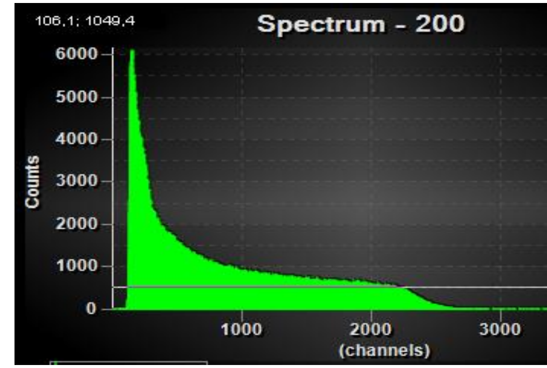
- Validation using known CAEN DT5800D generator



Pulse shape

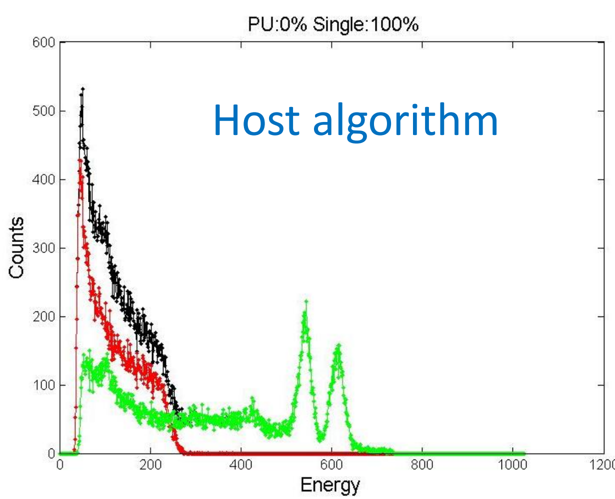


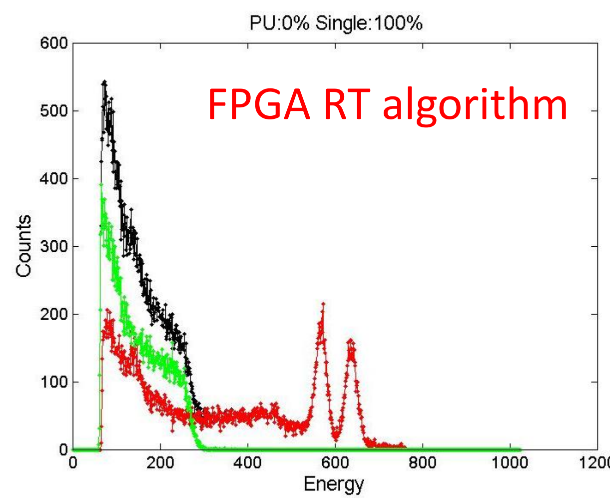
Gamma spectrum emulation



Neutron spectrum emulation

- Cross validation with FPGA code results (see next oral presentation)

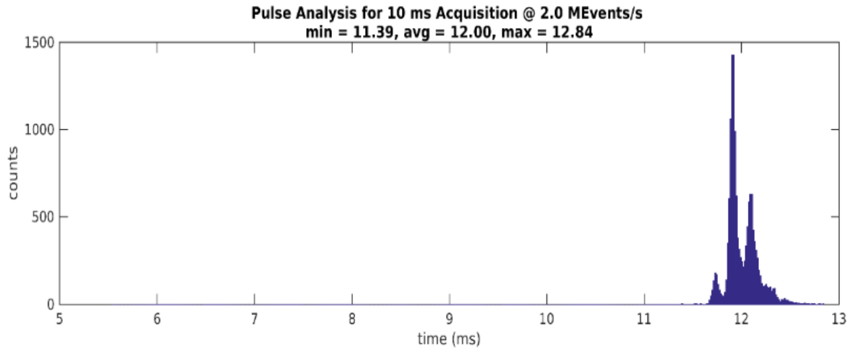


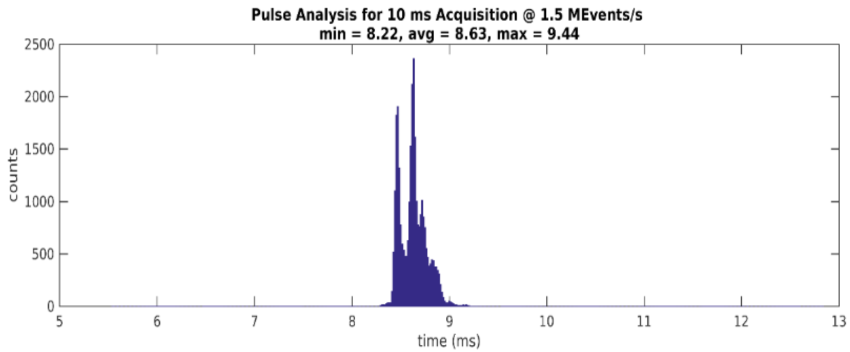


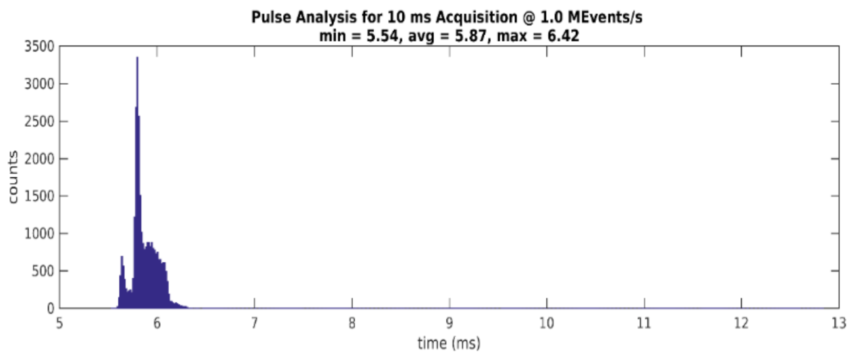
- Validation and performance measurements using Frascati Neutron Generator*

* To be presented R. C. Pereira et al., 30th Symposium on Fusion Technology (Sept 2018)

Performance Results
Pulse Processing







Processing time for different event rates using 1 logical core:

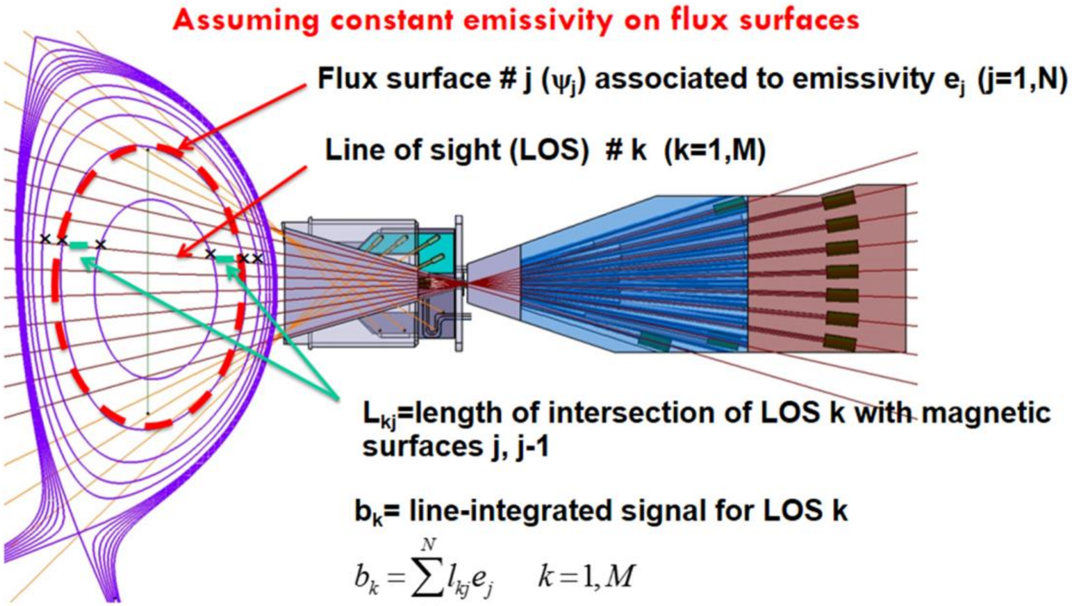
- 2.0 Mevents/s
- 1.5 Mevents/s
- 1.0 Mevents/s

Performance results:

- @ 1.0 & 1.5 Mevents/s processing times were always under 10 ms;
- @ 2.0 Mevent/s: all cycles needed more than 10 ms to process the data.

- To achieve 2.0 Mevents/s processing time under 10 ms, 2 logical cores must be used for pulse processing.

Inversion Algorithm



$$b_k = \sum_{j=1}^{N} l_{kj} e_j \qquad k = 1, M$$

$$b = L\,e \qquad (1)$$

Tikhonov regularization method

Solve the linear system (1) using the procedure, based on Tikhonov regularization (ref. [1] for a specific application to RNC) minimizing the functional

$$\left\| Le - b \right\|_2 + \alpha \left\| D\,\mathrm{e} \right\|_2$$

Regularization parameter to be minimized

Regularization operator to impose physical constraints to the solution

Algorithms
Development Strategies

The **GNU Scientific Library** was selected for algorithm implementation*.

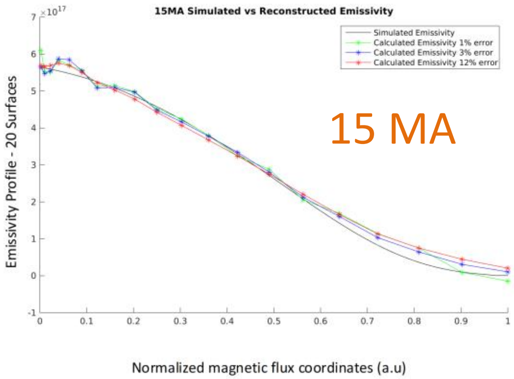
Main features:

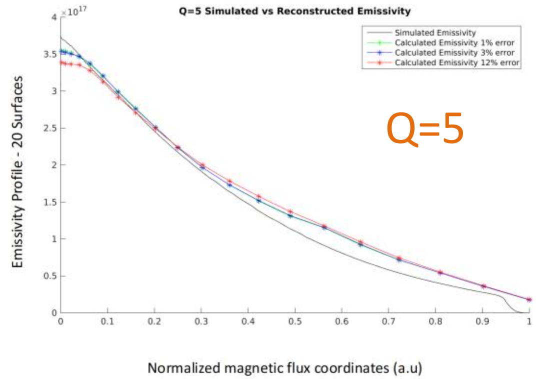
- Wide range of mathematical applications
- **Well structured, well documented, well supported**
- Globally the **best performant tool**
- GSL uses the ATLAS (Automatically Tuned Linear Algebra Software) implementations of BLAS, providing high performance.
- To obtain **higher performance**, GSL may use static linked ATLAS library and its CBLAS interface.

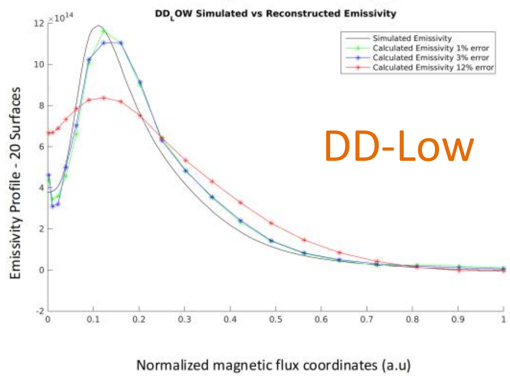
The algorithms are implemented as **Linux shared libraries** with documented API that can be used by other real time environments, according to the guidelines from F4E and ITER.

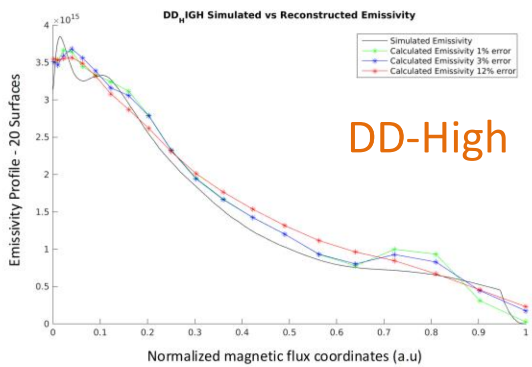
* Other options included Numerical recipies, Basic Linear Algebra Subprograms (BLAS), Linear Algebra Package (LAPACK),

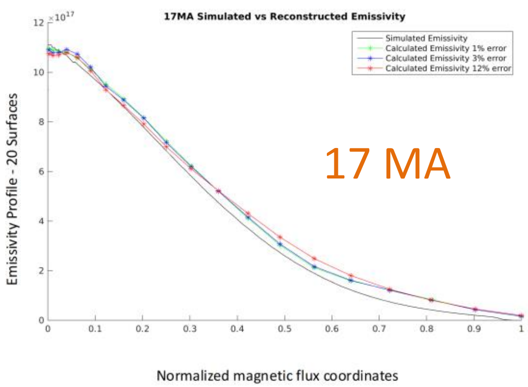
Validation
Inversion Algorithm

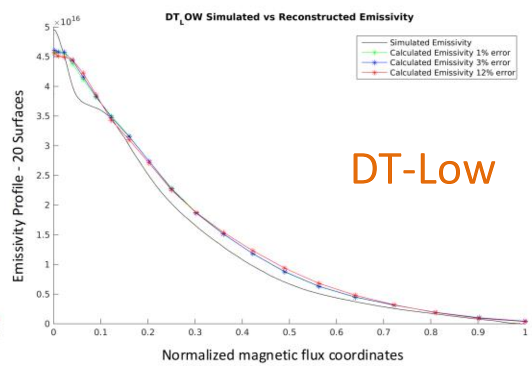












Validation for relevant ITER scenarios of the algorithm results was the first step to understand if the code run and converge with the correspondent simulated emissivity.

Crosscheck of the reconstructed emissivity, for different level of random noise (1%, 3% and 12%) using 20 magnetic surfaces, with the correspondent simulated emissivity was done for ITER relevant scenarios.

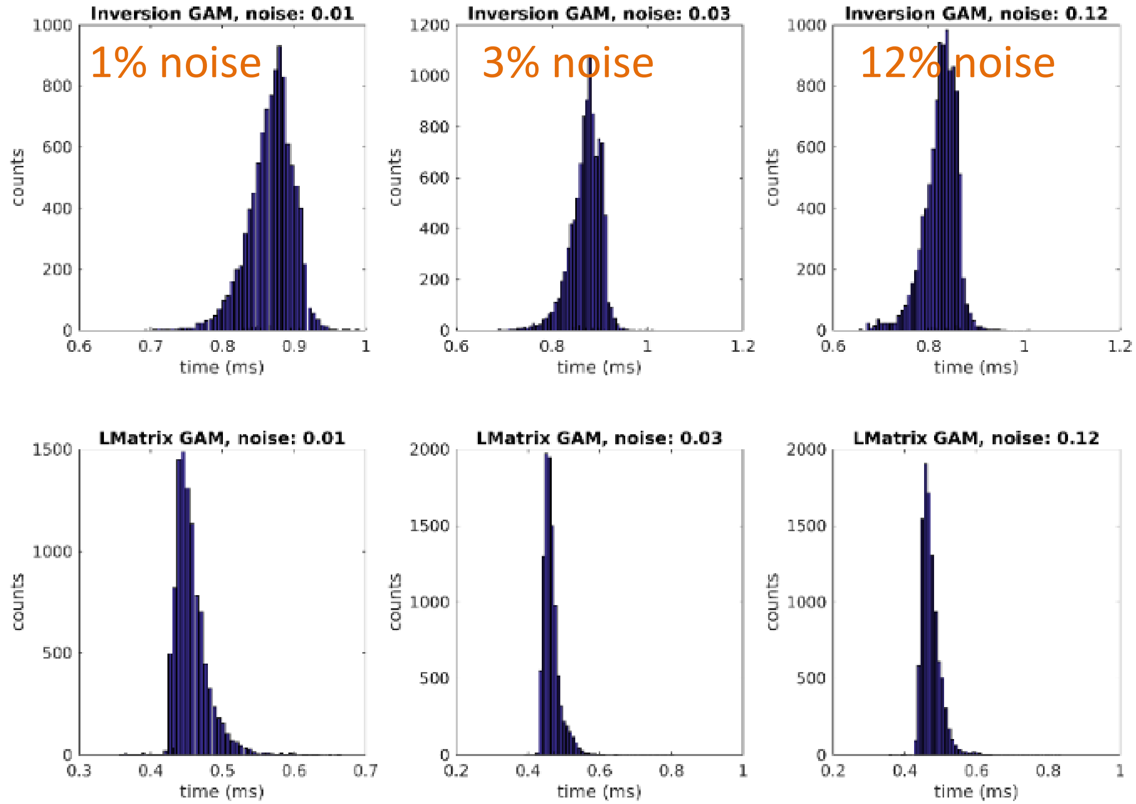
ITER Scenarios:

- 15 MA scenario;
- Q=5 scenario;
- DD-Low scenario;
- DD-High scenario;
- 17 MA scenario;
- DT-Low scenario.

Performance Results
Inversion Algorithm

Inversion Algorithm

L-Matrix Calculation



Run time calculations for the 2 processing modules of the real time inversion algorithm

Complete task runs under 1.5 ms

Summary

- A system architecture has been presented to help the final RNC system design
- A system prototype has been used to
  - Implement the most critical algorithms
  - Validate the results of the algorithm
  - Measure the performance of the system
- Performance tests helped to retrieve valuable information to size the optimal system configuration in terms of processing needs and number of acquisition channels per CPU
- The inversion algorithm to calculate the neutron emissivity profile can run in less than 2 ms
- The pulse processing algorithm runs in 12 ms using 1 CPU core for peak event rate of 2.0 Mevents/s
- FPGA pulse processing algorithm already implemented* is essential to improve the performance of the system

* A. Fernandes et al, next talk in this conference.