HiPEAC 2020 Conference
20-22 January 2020 - Bologna

Horizon 2020
European Union funding
for Research & Innovation

CERBERO

(Cross-layer modEl-based fRamework for multi-oBjective dEsign of Reconfigurable systems in unceRtain hybRid envirOnments)

PLANETARY EXPLORATION USE CASE

TASE: Pablo Sánchez de Rojas, Manuel Sánchez
UPM: Eduardo de la Torre, Eduardo Juárez, Alfonso Rodríguez, Leonardo Suriano, Daniel Madroñal
UNISS: Francesca Palumbo, Claudio Rubattu, Tiziana Fanni
UNICA: Carlo Sau, Luigi Raffo
INSA: Karol Desnos, Maxime Pelcat

http://www.cerbero-h2020.eu/

CERBERO requirements

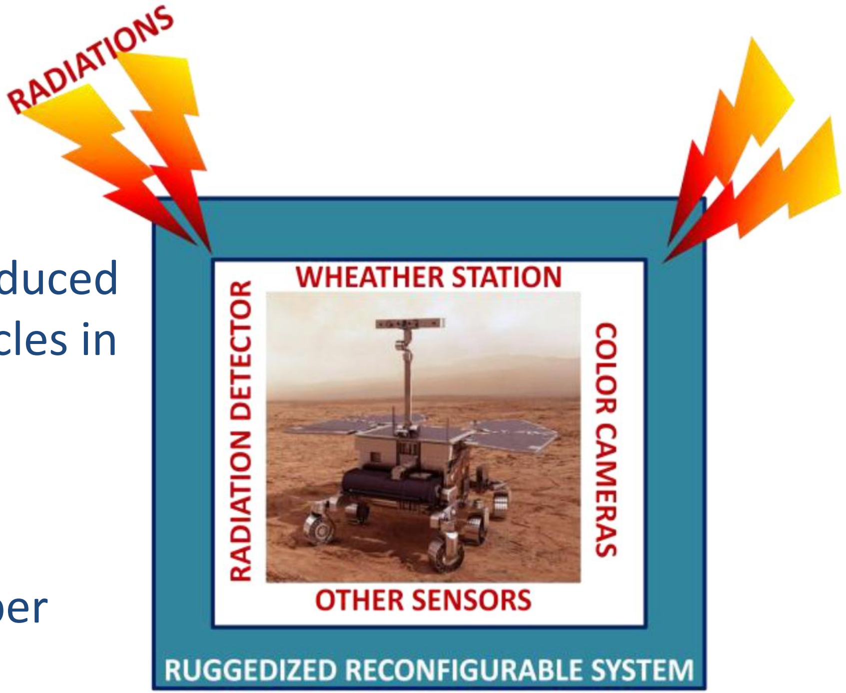
The CERBERO high level requirements related to the Self-Healing System for Planetary Exploration use case are:

- **Enable HW/SW co-design** for rad-tolerant control of robotic arm using adaptable COTS FPGA.
- **Development of integrated open-source** or commercially available toolchain **for development of robotics arms for space missions** with focus on multi-viewpoint system-in-the-loop virtual environment.
- **Development of self-adaption methodology** with supporting tools.

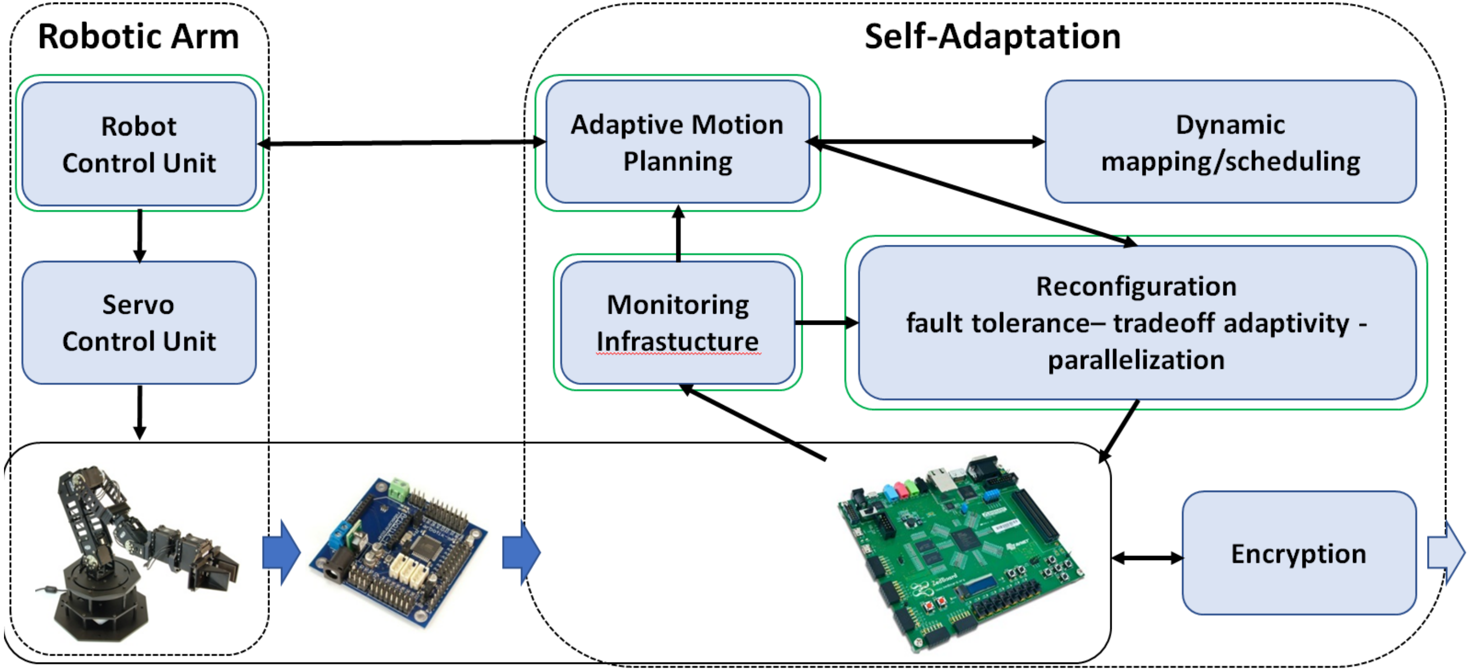
Use case goals

The main goals of the Self-Healing System for Planetary Exploration use case are:

- **Fault tolerance to single event effects** produced by the impact of subatomic radiation particles in electronic components
- **Adaption to harsh physical environment**
- **Power measurement and optimization** performed by HW/SW monitors on the cyber part of the computing platform



Skeleton of CERBERO tools for Planetary Exploration use case

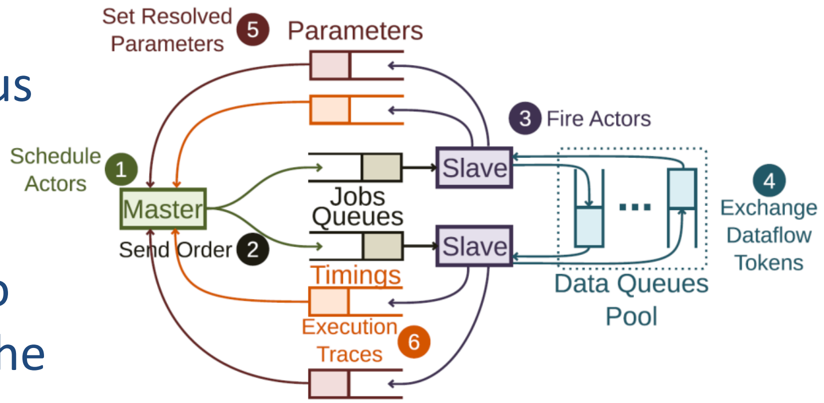


PREESM (INSA – Rapid prototyping and code generation tool):

- Allows algorithm graph description in PiSDF dataflow model and architecture graph description.
- Eases the design space exploration through graphic editors for all rapid prototyping inputs
- Provides static scheduling and memory optimization tasks
- Simulation and code generation tasks that provide metrics for system design and a prototype for testing the multicore execution of the system

SPIDER (INSA – Supporting tool for runtime adaption):

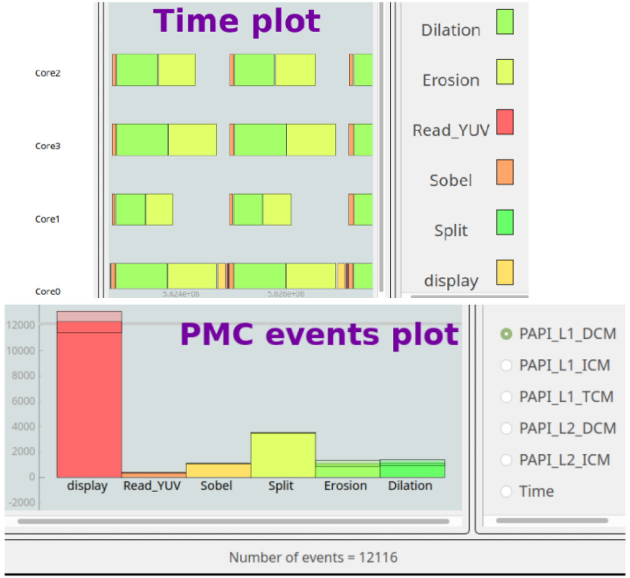
- Runtime manager for execution of reconfigurable PiSDF on heterogeneous MPSoCs
- Performs dynamic mapping and scheduling of the dataflow actors onto the different processing elements of the computing platform



INSA

PAPIFY (UPM – Automatic instrumentation and monitoring):

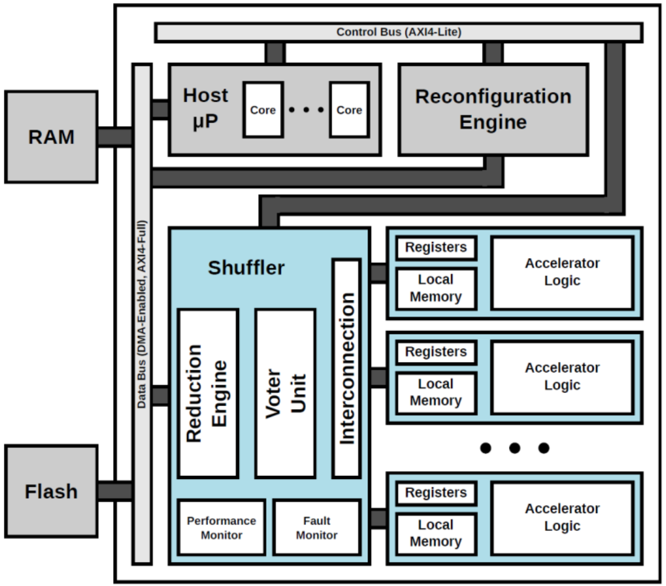
- Configuration of instrumentation for dataflow applications
- Automatic inclusion of monitoring function calls
- Performance monitoring of static and dynamic scheduling executions



POLITÉCNICA

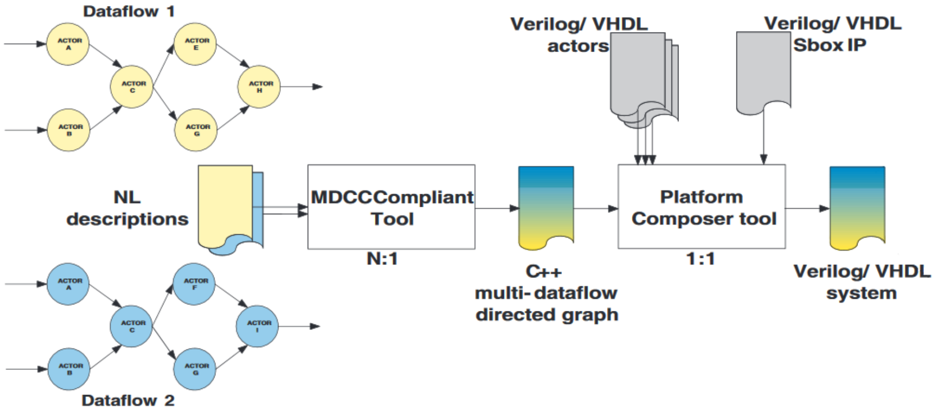
ARTICo³ (UPM – Framework for embedded system design):

- Architecture for flexible hardware acceleration
- Automated toolchain to build FPGA-based reconfigurable system
- Runtime execution environment to manage running applications



MDC (UNISS/UNICA – CGR hardware composition):

- Merges together different datapaths into one unique dataflow application by the insertion of switching modules
- Derives the RTL description of the CGR datapath from the multi-dataflow

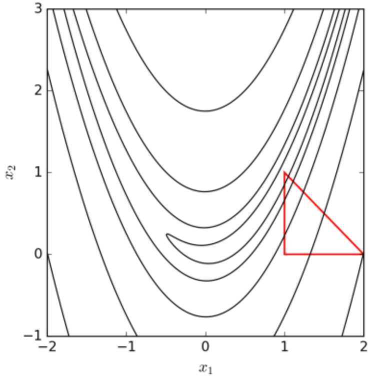


UNIVERSITÀ DEGLI STUDI DI SASSARI

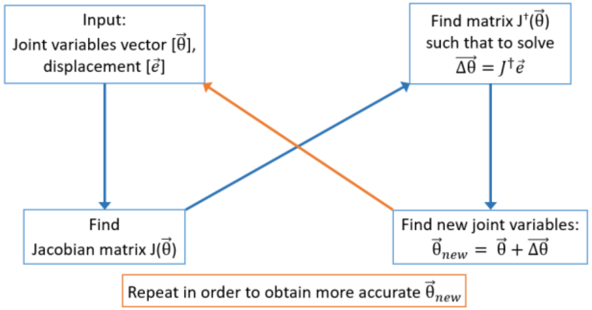
Algorithm diversity:

- Two motion planning algorithms: Nelder-Mead (N-M) and Damped Least Squares (DLS), with corresponding HW and SW versions
- **N-M guarantees given accuracy** but computation time is not predictable
- **DLS guarantees computation time** once accuracy and trajectory length has been defined

Nelder-Mead optimization

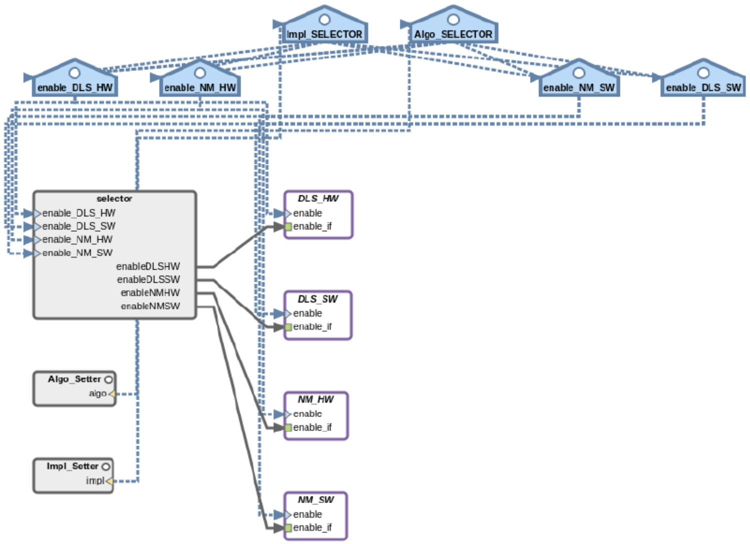


Damped Least Squares method



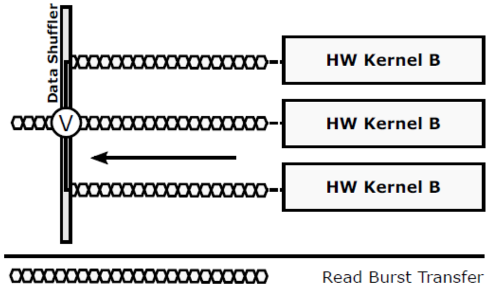
Parallelization and adaptable redundancy:

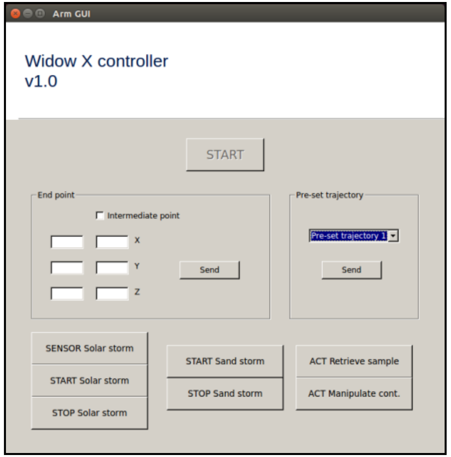
- Different levels of parallelization: acceleration through instantiation of multiple cost-function cores, and direct parallelization of the algorithm
- Different scalability possibilities: execution in software, with increasing number of accelerators and in redundancy mode (DMR/TMR)



Monitoring and optimization:

- Performance monitoring of internal metrics in both HW and SW components at execution time
- Emulation of radiation failures, where error counters may be accessed through PAPI components
- External event emulation and user commands received from a Graphic User Interface





DEMO TIME