Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Rapid prototyping of control modules for the DTT Plasma Control System

Sara Dubbioso[1,2], D. Ottaviano[2], F. Fiorenza[1,2] N. Ferron[3], G. Manduchi[3], R. Ambrosino[1,2,4], G. De Tommasi[1,2]

[1] Consorzio CREATE, via Claudio 21, 80125, Napoli, Italy
[2] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Napoli, Italy
[3] Consorzio RFX, C.so Stati Uniti 4, 35127 Padova, Italy [4] DTT S.C.a.r.l., via Enrico Fermi 45, Frascati, Rome, Italy

Motivations

Rapid prototyping by means of **automatic code generation (ACG)** is an essential tool in the context of HW/SW codesign of real-time systems [1].

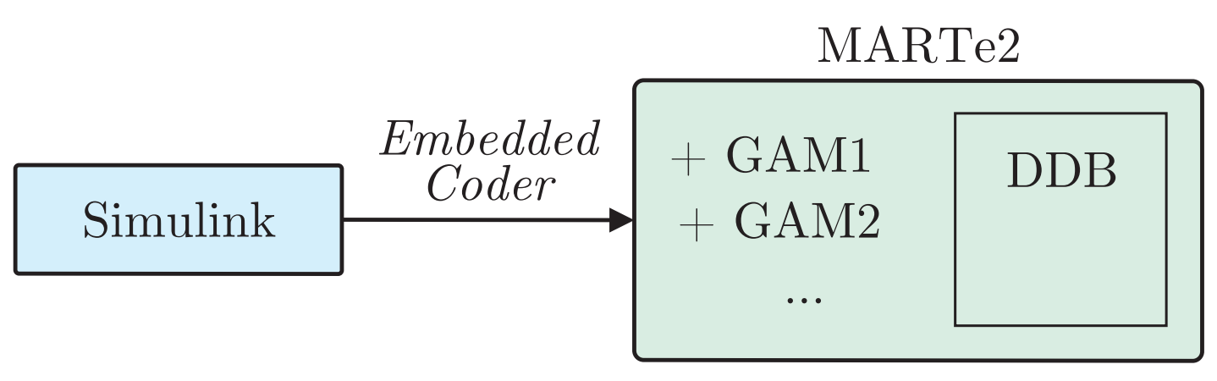
- it contributes to **reduce both the development cost and time**;
- it permits to easily **comply with quality standards** (such as MISRA [2]);
- it contributes to improve safety and reliability of the control software.

Enabling ACG for rapid prototyping is facilitated by a wide choice of commercial tools (Simulink Coder [3], dSPace platform [4],. . .), that can be customized to deal with practically any HW/SW platform.

DTT Framework for Rapid Prototyping of real-time control modules

The **DTT framework to deploy real-time control and data acquisition systems** relies on an **integrated environment** based on:

- **MDSplus [5]** – a data system widely used in the fusion community that provides:
  - a complete Application Programming Interface (API) for data management;
  - a set of tools for data visualization and configuration browsing.
- **MARTe2 [6]** – a framework for real-time applications originally developed at JET and currently used on several machines, including TCV;
- **Simulink Embedded Coder [3]**



References

[1] J. Teich *Hardware/software codesign: The past, the present, and predicting the future*, Proc. of the IEEE (2012)

[2] D. Ward, *MISRA Standards for Automotive Software*, 2^{nd} IEE Conf. on Automotive Electronics (2006)

[3] Simulink Embedded Coder https://www.mathworks.com/products/embedded-coder.html (2024)

[4] dSpace https://www.dspace.com (2024)

[5] G. Manduchi, A. Luchetta, C. Taliercio, A. Neto, F. Sartori, G. De Tommasi *Integration of Simulink, MARTe and MDSplus for rapid developmentof real-time applications*, Fus. Eng. Des. (2015)

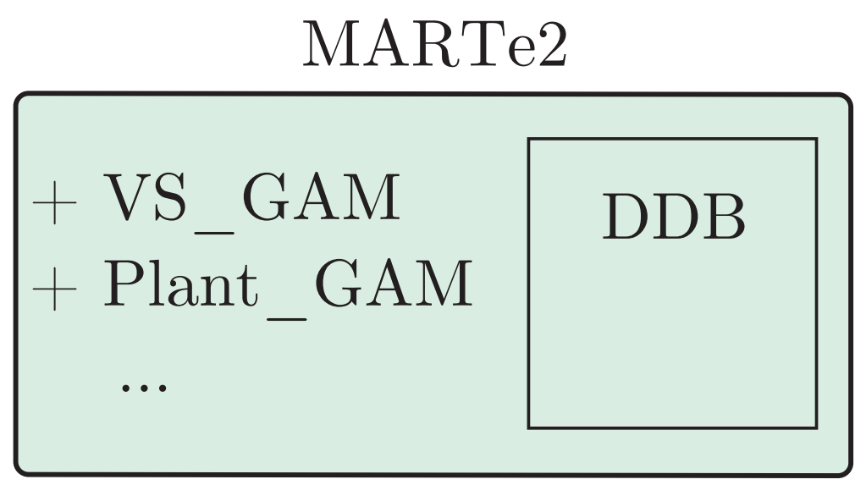
[6] A. Neto et al. *MARTe: a Multi-Platform Real-Time Framework* IEEE Trans. Nucl. Sci. (2010).

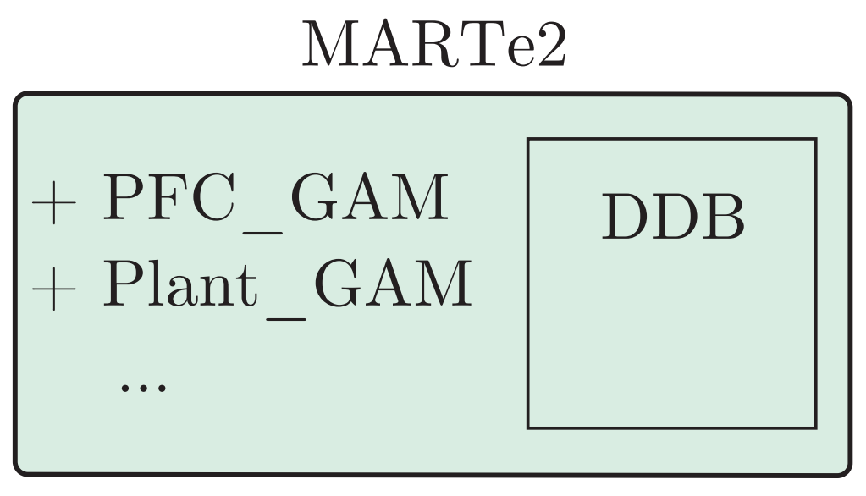
[7] D. Ottaviano, F. Ciraolo, R. Mancuso, and M. Cinque *The Omnivisor: A Real-Time Static Partitioning Hypervisor Extension for Heterogeneous Core Virtualization over MPSoCs*, ECRTS 2024.

The research leading to these results has been partially funded by the Project "TRAINER - Tokamak plasmas daTa-dRiven identificAtIon and magNEtic contRol" CUP E53D23014670001 funded by EU in NextGenerationEU plan through the Italian "Bando Prin 2022 - D.D. 1409 del 14-09-2022" by MUR

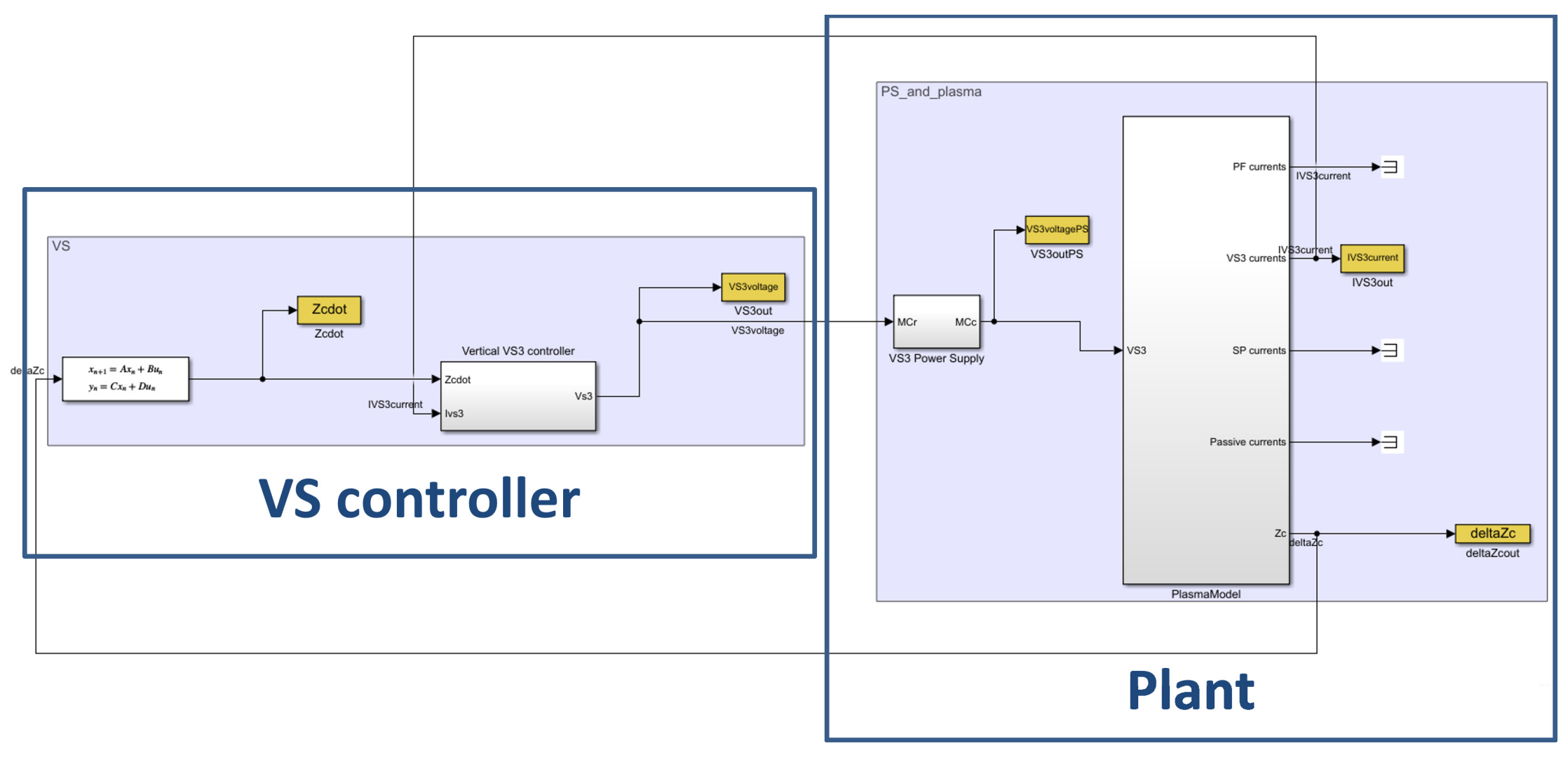
Rapid prototyping of DTT magnetic control modules

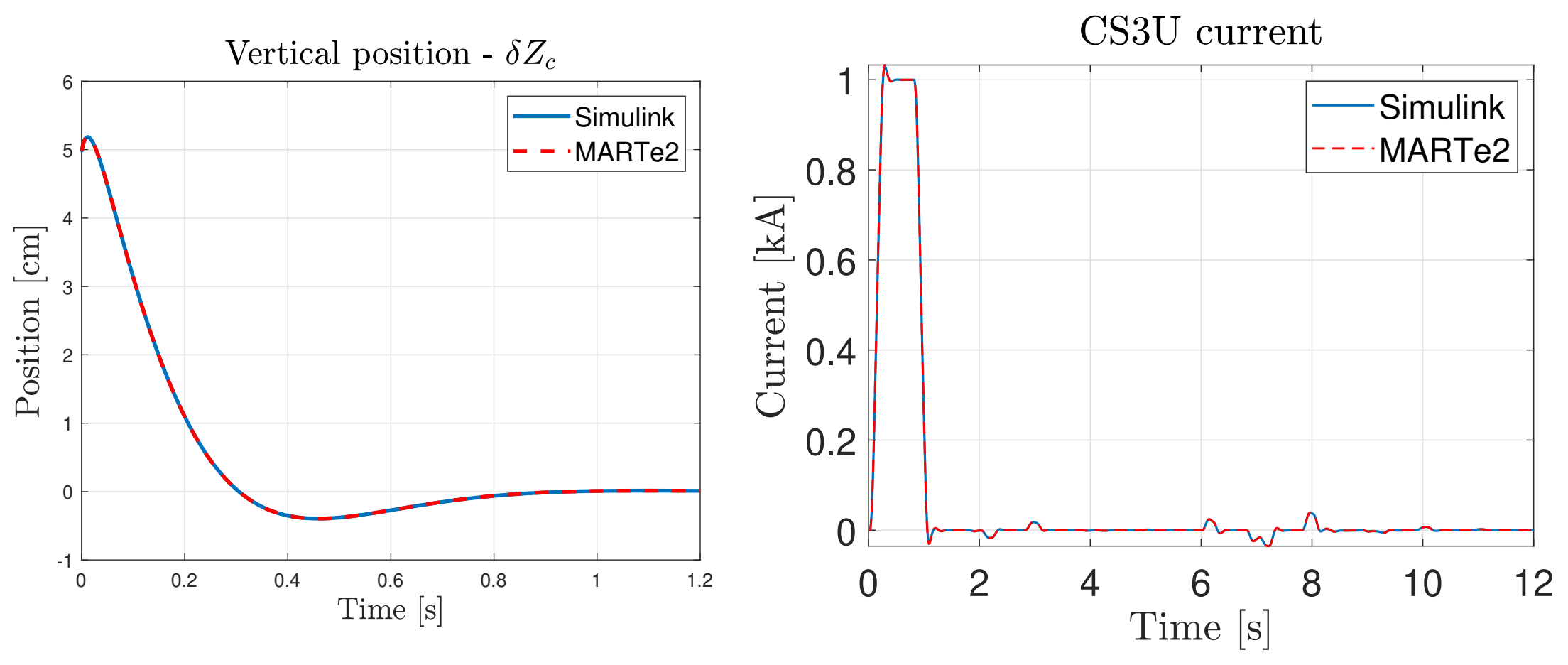
The first validation of the automatically generated code is done by running both the plant and the controller within a single MARTe2 instance



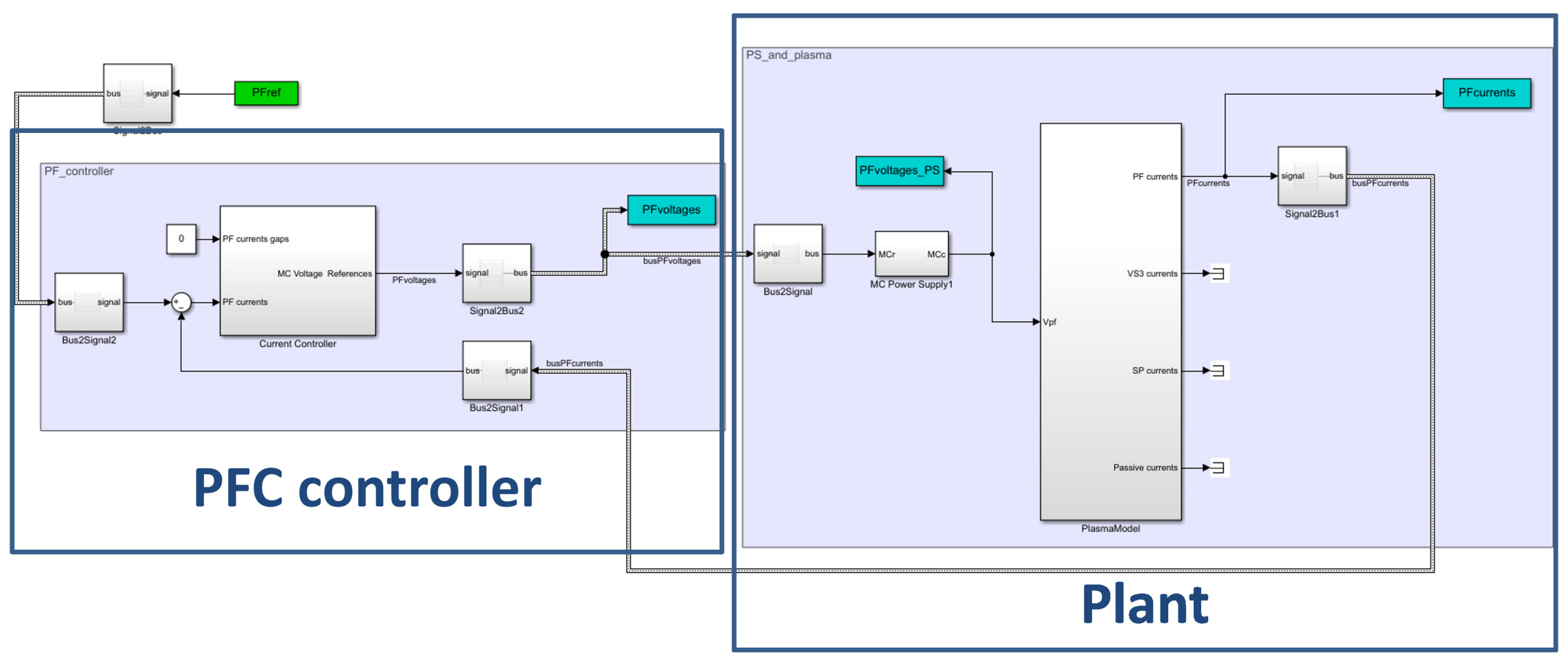


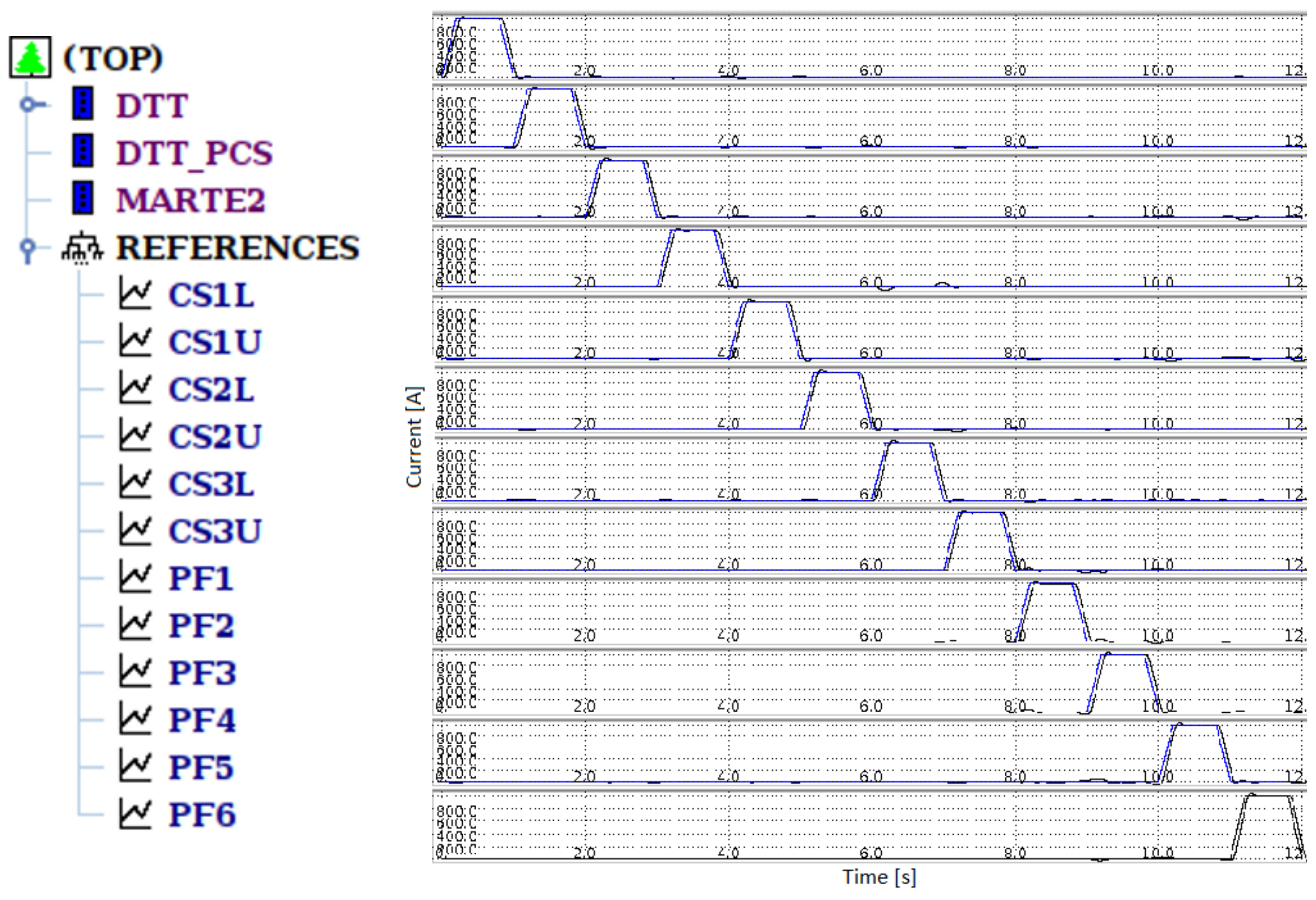
Test case 1: Vertical Stabilization controller



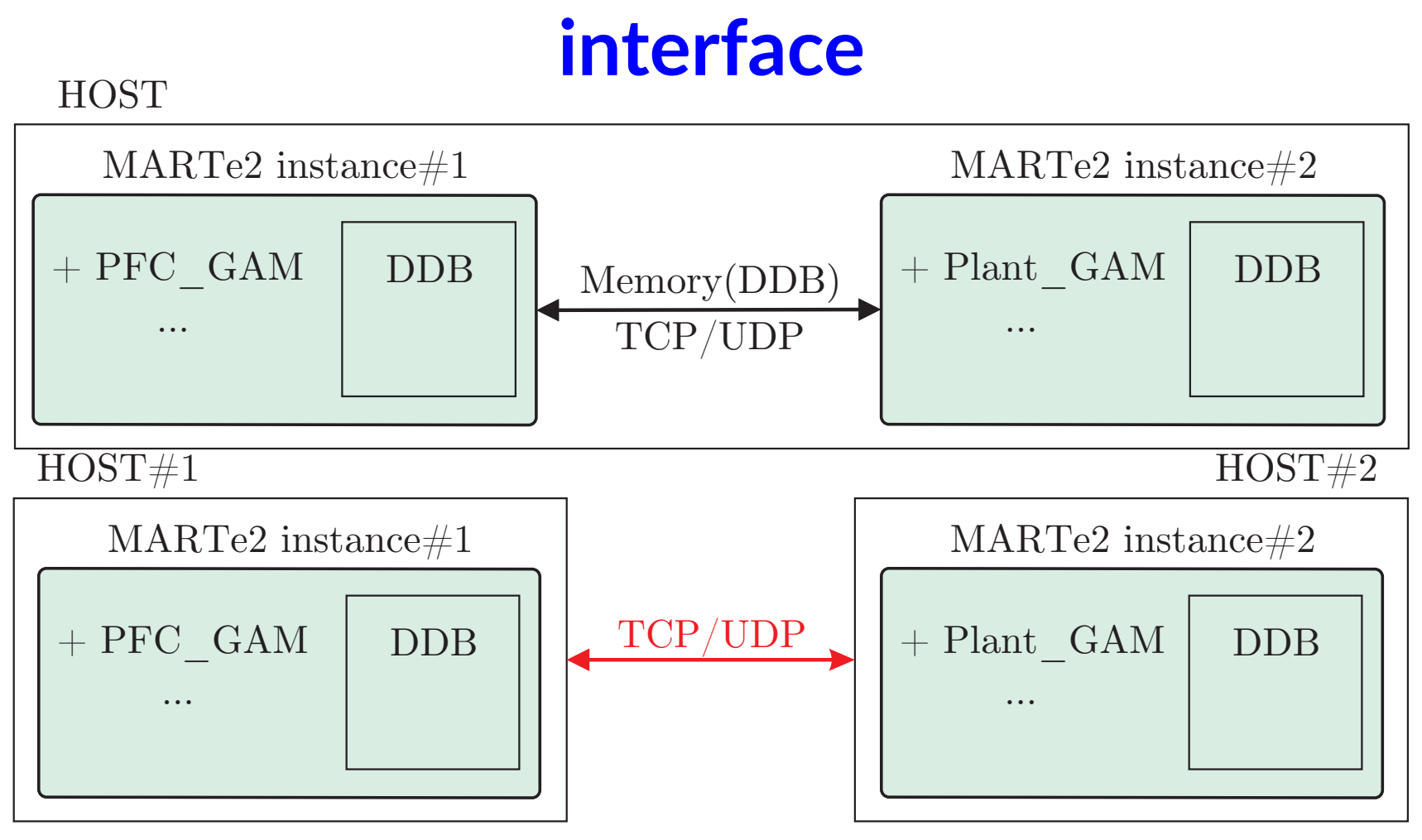


Test case 2: PF current decoupling controller



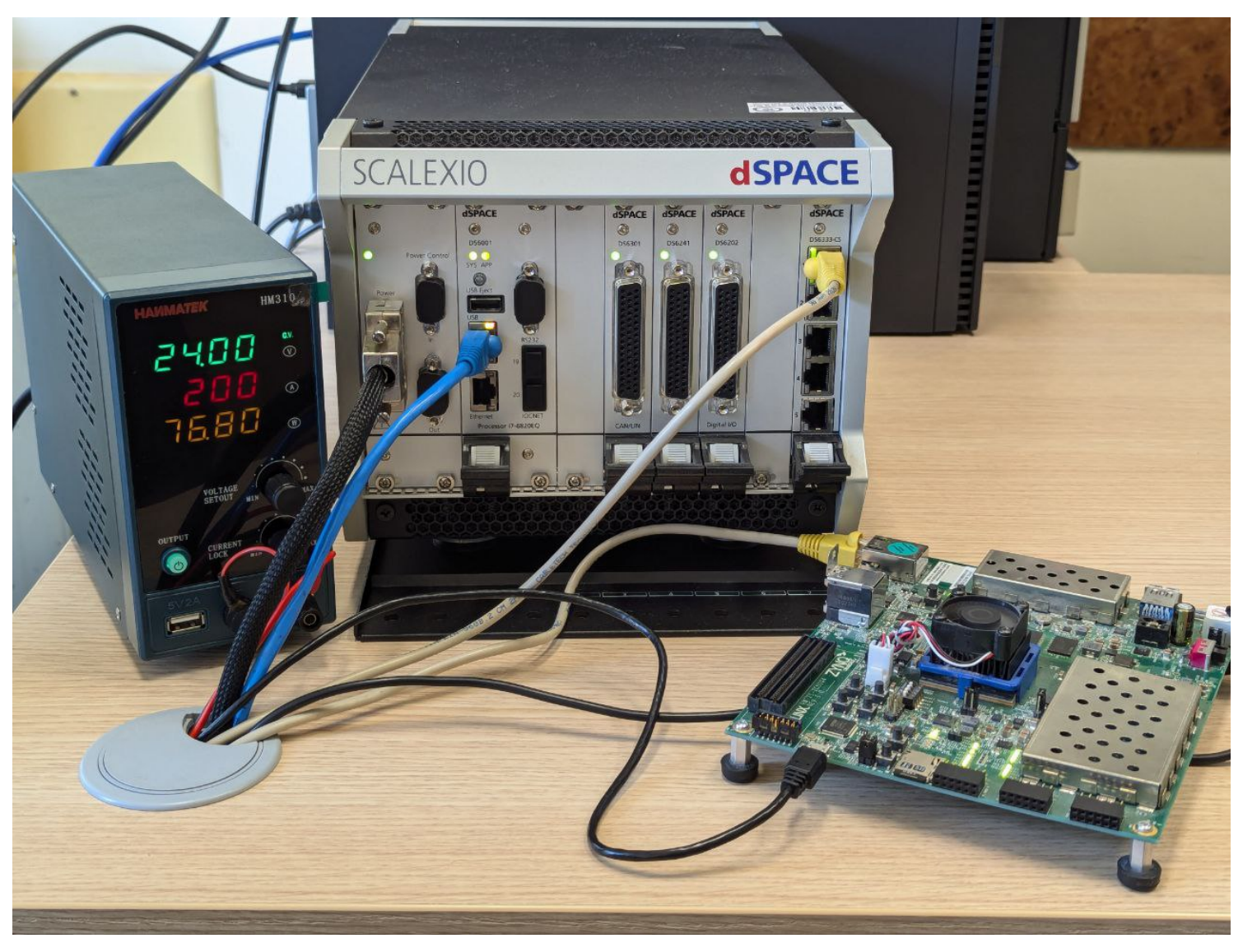


Setup for Test & Validation of the communication interface



Such setup can be exploited to assess performance against network connection reliability

Setup for Hardware-In-the-Loop (HIL) validation



Relies on a dSpace SCALEXIO real-time target

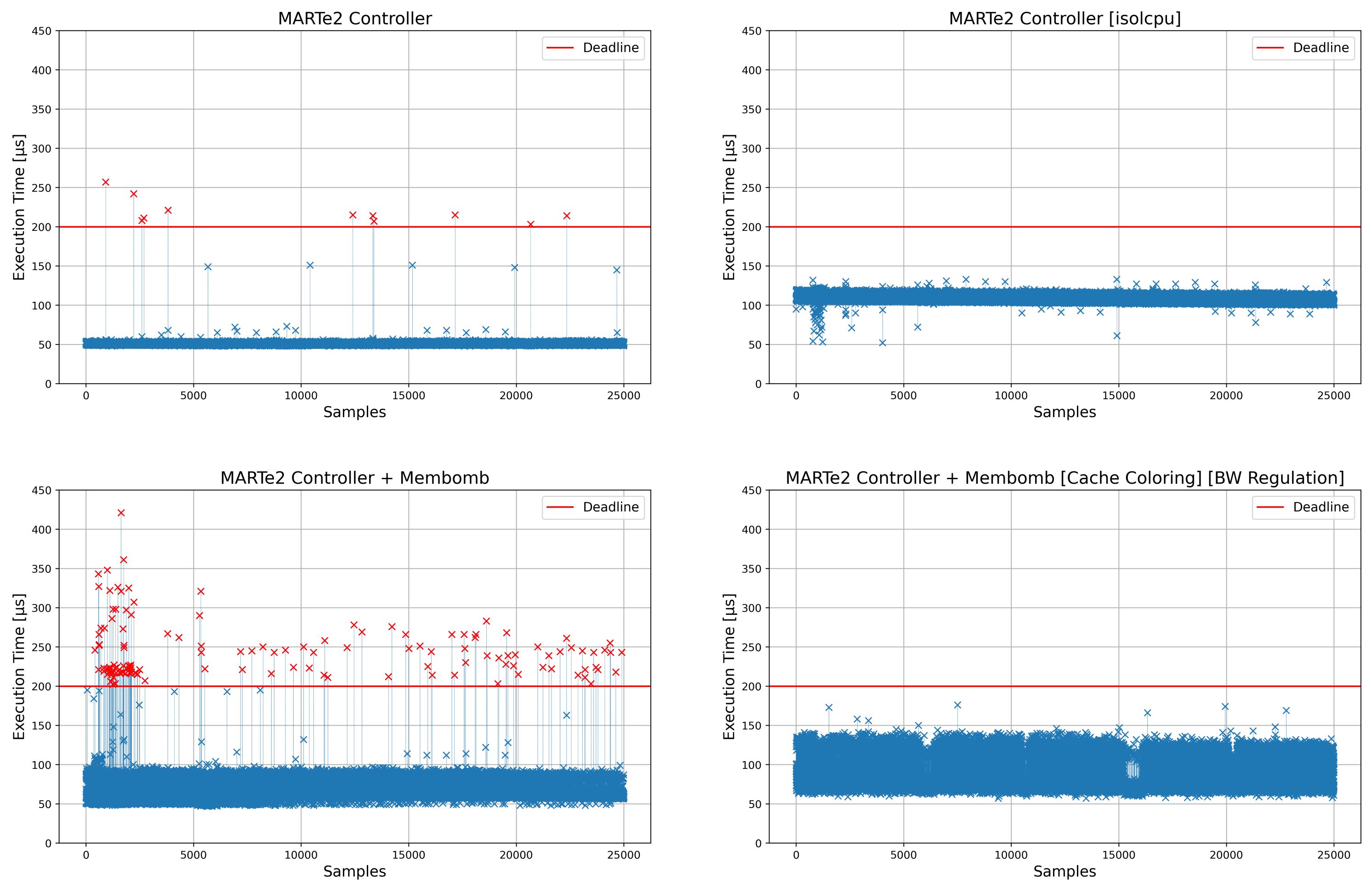
HIL performance assessment - Case study

A HIL setup is deployed to assess the performance of a MARTe2 controller on a ARM-based Multiprocessor System-on-Chip (MPSoC):

- The **dSPACE SCALEXIO** real-time target is used to emulate the plasma model
- The **Zynq™ UltraScale+™ MPSoC** board from Xilinx runs the MARTe2 controller at 5 kHz

The assessment has shown different predictability results by **varying the software configuration**

- **Solo Controller**: the controller is initially executed on a Linux system without any co-located application. The execution time is in the needed range in average but the non-real-time nature of Linux causes sporadic deadline misses during execution.
- **IsolCpu**: by using the Linux *PREEMPT-RT* patch and the *isolcpu* feature the controller achieves more predictable results without any deadline misses at the cost of higher average execution time. However, the board is underutilized since we use only one processor.
- **Membomb**: by co-locating applications with high memory bandwidth utilization, despite the *isolcpu* and *PREEMPT-RT* mechanisms, the controller is subjected to strong interferences causing deadline misses.
- **Regulation**: *by containing the application in Virtual Machines, partitioning the cache, and regulating the memory bandwidth*, the execution time is more stable without any unexpected interference [7].


MARTe2 Controller
MARTe2 Controller [isolcpu]
MARTe2 Controller + Membomb
MARTe2 Controller + Membomb [Cache Coloring] [BW Regulation]
Deadline
Execution Time [us]
Samples