HVDC Factory Acceptance Tests supported by test automation using RSCAD Python API

- Niklas Strunz
- Siemens Energy



RTDS USER'S GROUP MEETING

2026 MADRID USER'S GROUP MEETING
MADRID, SPAIN

Agenda

01 **Introduction**

Background and the HVDC RTS team

02 **HiL verification and automation**

Test environment, workflow and implementation

03 **Execution challenges**

Reconfiguration and automated evaluation

04 **Automation in Factory Acceptance Tests**

Scope, consistency, speed and customer benefits

Introduction

Background

M.Sc. Energy Science and Technology

FAU Erlangen, 2022

Joined Siemens Energy in 2022

HVDC Control and Protection

HVDC RTS team

Our team's focus

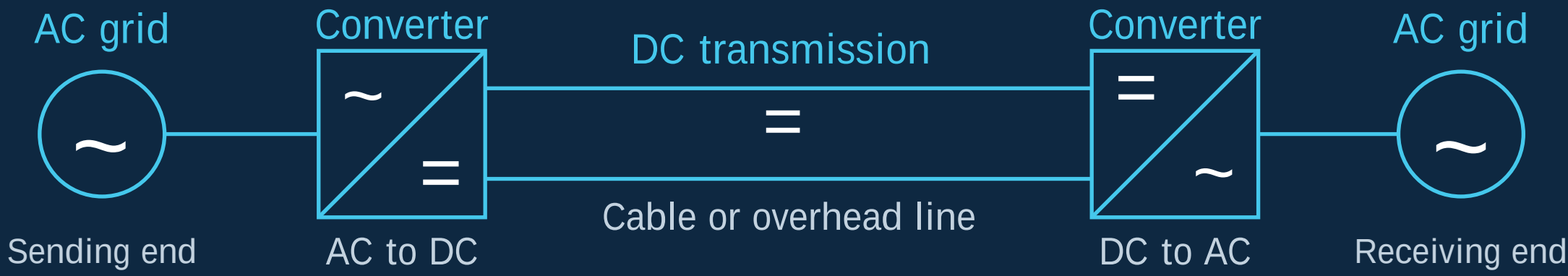
Dynamic grid and HVDC simulation

for HiL environments
HiL expertise

Replica planning and execution

Basic HVDC system

HVDC = High-voltage direct current



1 Convert

Change AC power into DC power.

2 Transmit

Carry power over the DC connection.

3 Convert back

Supply AC power to the receiving grid.

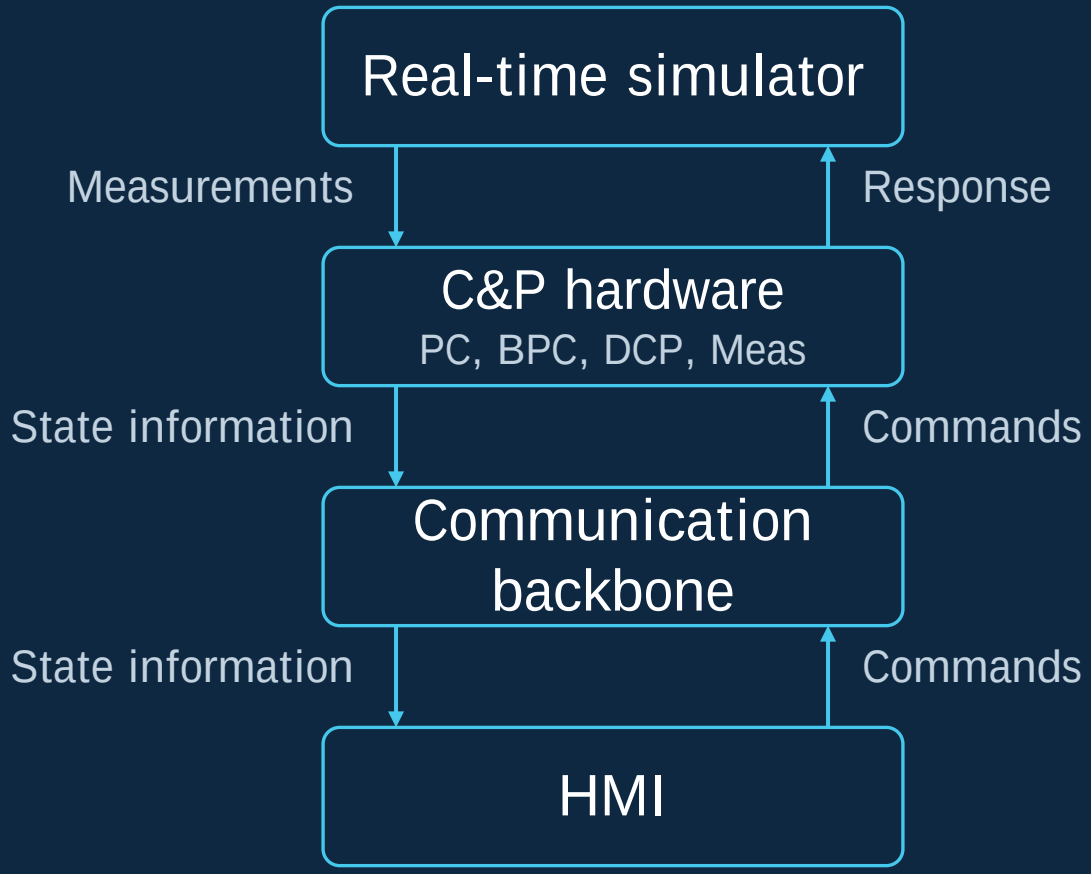
HiL verification of HVDC control & protection

Verification scope

Actual control and protection hardware and software

The real-time simulator supplies secondary measurement signals

Original hardware and I/O



Test automation in HiL environments

Conventional software tests

Requirements and test instructions

become executable test code

A controlled setup typically provides

a known starting state

HiL automation

Test logic also maps actions and checks

to different communication interfaces

The initial system state may be unknown

and needs explicit verification

HiL test automation sequence

1 Observe
Read the current state

→ **2 Reconfigure**
Establish the desired state

→ **3 Verify**
Confirm test preconditions

→ **4 Execute**
Run only when ready

→ **5 Evaluate**
Automated analysis of results

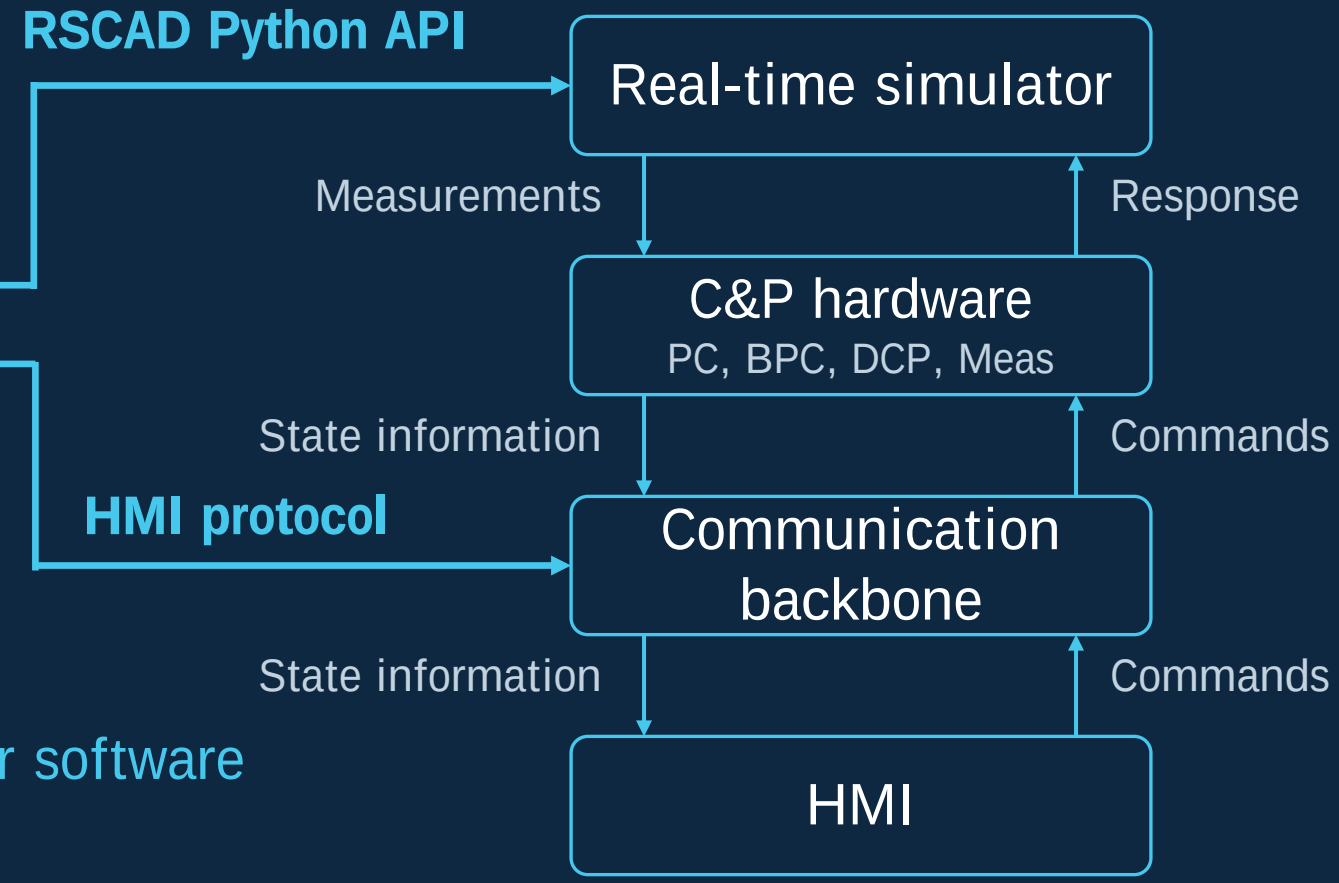
Automation in our HVDC environment

Simulator and operator-level control

Python automation

Direct backbone access using the same protocol and operator commands as the HMI

No C&P hardware and only minor software changes
Less interference with the system



Challenges in automated test execution

Reconfiguration and evaluation need the most attention

1	**Observe**	Established	Read required data through the established link.
2	**Reconfigure**	**Key challenge**	Project-specific topologies and transition paths.
3	**Verify**	Established	Check test preconditions using backbone data.
4	**Execute**	Established	Trigger events using simulator and system control.
5	**Evaluate**	**Key challenge**	Changing dynamics limit automatic result analysis.

Current assessment for this automation environment

Reconfiguration between consecutive tests

Project topology determines which transitions are possible

Option 1: Reset and replay

Reset to a defined state

(e.g. de-energized).

Follow a default path to the next test state.

Main limitation

Abrupt, non-realistic changes can conflict with the real hard- and software.

Option 2: State machine

Model valid reconfigurations.

Use system responses to identify allowed transitions and plan a sequence.

Proposed approach

Reach the desired state through valid transitions rather than forced resets.

Automated evaluation: capabilities and limits

Complex dynamic tests

Dynamics vary from test to test. Our current implementation cannot yet evaluate complex dynamic tests automatically.

Selected regression tests support automatic checks

A fixed acceptance rule, such as a constant ramp rate or a prohibited event, makes the result check explicit.

Power ramp

Check a constant ramp rate of 50 MW/min

Protection tests

Compare test to expected high level behavior

Automation scope in FAT

Software integrity remains the priority

Retain HMI-based functional tests

Basic operating sequences

Integrity of operator inputs

These tests must include the HMI.

Direct backbone access bypasses it and cannot verify its behavior.

Automate repetitive test cases

Fault ride-through scenarios

Protection tests

Here, the HMI only helps establish the test state. Automation can perform that setup and execute the test.

Repetitive FAT: consistency and speed

More consistent execution

Repeat programmed sequences consistently, reducing manual entry errors and variation between runs.

Faster configuration

Programs can set hundreds of values per second.

Manual HMI input is slower and ties up test personnel.

Same test outputs and report format

Automation retains the outputs and test report used for manual execution. Complex results still require expert analysis.

Q&A