



CHALMERS
UNIVERSITY OF TECHNOLOGY



Experimental Commissioning and Performance Testing of Vehicle-to-Grid Operation Using Hardware-in-the-loop

Master's thesis in Embedded Electronic System Design

William Olofsson Maronen

Department of Microtechnology and Nanoscience
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

MASTER'S THESIS 2026

Experimental Commissioning and Performance Testing of Vehicle-to-Grid Operation Using Hardware-in-the-loop

William Olofsson Maronen



Department of Microtechnology and Nanoscience
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Experimental Commissioning and
Performance Testing of Vehicle-to-Grid Operation Using Hardware-in-the-loop
William Olofsson Maronen

© William Olofsson Maronen, 2026.

Supervisor: Lena Peterson, Microwave Electronics, Microtechnology and Nanoscience
Company advisor: Jonas Hellgren, Xiaoliang Huang, RISE
Examiner: Per Larsson-Edefors, Microwave Electronics, Microtechnology and Nanoscience

Master's Thesis 2026
Department of Microtechnology and Nanoscience
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in L^AT_EX
Gothenburg, Sweden 2026

Experimental Commissioning and
Performance Testing of Vehicle-to-Grid Operation Using Hardware-in-the-loop
William Olofsson Maronen
Department of Microtechnology and Nanoscience
Chalmers University of Technology

Abstract

The increasing proliferation of electric vehicles presents new opportunities for grid support through vehicle-to-grid (V2G) technology, enabling bidirectional power exchange between vehicles and the electrical grid. While the theoretical benefits of V2G are well established, experimental validation using standardized communication protocols under real-time conditions remains scarce. In particular, hardware-in-the-loop (HIL)-based V2G operation using the ISO 15118-20 communication standard has seen little experimental demonstration. This thesis addresses that gap by designing and commissioning a real-time direct current (DC) V2G HIL testbed built around a dSPACE MicroLabBox II, a dSPACE Smart Charging Interface (DS5366), and a Keysight communication tester (SL1550A) operating in electric vehicle communication controller (EVCC) mode. The testbed emulates the full DC charging chain, including power electronics, battery dynamics, and ISO 15118 -20 compliant electric vehicle (EV)–electric vehicle supply equipment (EVSE) communication, and supports both manual and automated experiment execution. The charging power is commanded through the ISO 15118-20 dynamic power request, and the experiments comprise step and ramp tests of the power response together with power-profile tests that replay grid frequency events relevant to frequency containment reserve for normal operation (FCR-N) and frequency containment reserve for disturbance (FCR-D). The testbed successfully demonstrates DC V2G operation under ISO 15118-20, confirming that HIL-based validation with standardized high-level communication is a viable and scalable alternative to costly vehicle-based field trials. Future work includes evaluating hardware-induced latencies and extending the testbed toward Nordic frequency containment reserve (FCR) prequalification testing.

Keywords: Vehicle-to-Grid, Hardware-in-the-Loop, ISO 15118-20

Acknowledgements

I would like to express my sincere gratitude to my industrial supervisors Jonas Hellgren and Xiaoliang Huang for their guidance, technical expertise, and support throughout the course of this work. Their insights and feedback have been invaluable in shaping both the direction and quality of this thesis.

I also wish to thank my academic supervisor Lena Peterson for her continuous support, constructive feedback, and encouragement during the project.

William Olofsson Maronen, Gothenburg, September 2026

Contents

List of Abbreviations	1
1 Introduction	3
1.1 Related Work	4
1.2 Purpose and Goal	5
1.3 Ethical, Societal and Ecological Aspects	6
1.3.1 Use of Generative AI	6
1.4 Thesis Outline	6
2 Technical Background	9
2.1 EV Charging	9
2.1.1 Charging Scenarios	9
2.1.2 Conductive AC	10
2.1.3 Conductive DC	10
2.1.4 Charging Connectors	11
2.1.5 Smart Charging	11
2.1.6 Bidirectional Charging	12
2.2 ISO 15118	12
2.2.1 Comparison with DIN 70121	12
2.2.2 Low-level Communication	13
2.2.3 High-Level Communication	15
2.2.4 Scheduled and Dynamic Control	16
2.3 Ancillary Services	16
2.3.1 Fast Frequency Reserve	17
2.3.2 FCR-N	18
2.3.3 FCR-D	18
2.3.4 Comparison and Performance Requirements	19
2.4 Model based Development	20
2.4.1 Model-in-loop	20
2.4.2 Software-in-loop	20
2.4.3 Hardware-in-loop	21
2.4.4 Physical Testing	22
3 Method	23
3.1 Simulation Platform and Test Equipment	23
3.1.1 dSPACE	23

3.1.2	Keysight	23
3.2	Hardware	24
3.3	Software	26
3.3.1	dSPACE Toolchain	26
3.3.2	Keysight Software	27
3.3.3	Model Parameters	27
3.4	Iterative Workflow	28
3.5	Experimentation	30
4	Design	31
4.1	Model Interconnects	31
4.2	Customized Simulink Models	32
4.2.1	Power-control Restbus	33
4.2.2	Charge-state Extraction	34
4.2.3	Charging-standard Selection	34
4.3	dSPACE ConfigurationDesk Setup	35
4.3.1	Intermodel Communication	35
4.4	Automated Testing Scripts	35
4.4.1	Frequency-to-power Profile Generation	35
4.4.2	Variable Access	36
4.4.3	Post-processing and Metrics	36
5	Results	37
5.1	Low-level Communication	37
5.2	Step-response	38
5.3	Ramp-response	40
5.4	Power Profile	40
5.4.1	FCR-N	41
5.4.2	FCR-D Up	44
5.4.3	FCR-D Down	47
5.5	Summary	50
6	Conclusion	53
	Bibliography	55
A	ISO 15118-20 Flowchart	I
B	SLAC	III
C	SECC Discovery Protocol	VII
D	V2G messages	IX
D.1	Communication setup sequence	XI
D.2	Identification, authentication, and authorization sequence	XI
D.3	Target setting and charge scheduling	XI
D.4	Charging control and re-scheduling	XII
D.5	End of charging process	XII

E	Automation Scripts	XIII
E.1	Frequency-to-power profile generation	XIII
E.2	ControlDesk playback	XIV
E.3	Ramp test	XVI
E.4	Measurement data extraction	XVII

List of Abbreviations

AC alternating current

C-HIL controller-hardware-in-the-loop

CAN controller area network

CCS2 combined charging system type 2

CP control pilot

DC direct current

DIN German Institute for Standardization

DS5366 dSPACE Smart Charging Interface

EV electric vehicle

EVCC electric vehicle communication controller

EVSE electric vehicle supply equipment

FCR frequency containment reserve

FCR-D frequency containment reserve for disturbance

FCR-N frequency containment reserve for normal operation

FFR fast frequency reserve

HIL hardware-in-the-loop

IEC International Electrotechnical Commission

IP Internet Protocol

IPv6 Internet Protocol version 6

ISO International Organization for Standardization

MDF measurement data format

MIL model-in-the-loop

OBC on-board charger

PE	protective earth
PEV	plug-in electric vehicle
PLC	powerline communication
PnC	plug and charge
PP	proximity pilot
PWM	pulse width modulation
SCE	Scienlab Smart Charging Emulation
SDP	SECC discovery protocol
SECC	supply equipment communication controller
SIL	software-in-the-loop
SL1550A	Keysight communication tester
SLAC	signal attenuation characterization
SoC	state-of-charge
TCP	Transmission Control Protocol
TLS	transport layer security
UDP	User Datagram Protocol
V1G	smart charging
V2G	vehicle-to-grid
V2H	vehicle-to-home
V2L	vehicle-to-load
V2X	vehicle-to-everything
WPT	wireless power transfer

1

Introduction

The transition toward electrified transportation presents both significant opportunities and substantial challenges for future power systems. Electric vehicles (EVs) are not only energy consumers but also mobile distributed energy storage units capable of injecting power back into the electrical grid. The bidirectional exchange of power between EVs and the grid, commonly referred to as vehicle-to-grid (V2G), has yet to achieve widespread adoption, and its practical large-scale feasibility remains largely unverified. The fundamental premise of V2G is to enable charging station operators to provide ancillary services such as frequency containment reserve (FCR) to the grid in return for financial compensation. By doing so, V2G has the potential to reduce charging costs and accelerate the adoption of electric transportation [1].

While the theoretical advantages of V2G are well documented in the literature, experimental studies focusing on real-time implementations are comparatively scarce. Many existing approaches rely exclusively on simulation-based models, and those that do include experimental validation often fail to capture the effects of hardware-induced delays, signal quantization, and communication latency inherent in real-time operation [2]. These non-idealities can significantly influence the stability, responsiveness, and overall efficiency of V2G control systems, potentially limiting their viability as ancillary service providers to the power grid [3].

Furthermore, conventional testing of grid-interactive EV functionality typically relies on full vehicle prototypes or large-scale field trials, which are costly, time-consuming, and difficult to reproduce. These limitations hinder rapid development and validation of V2G technologies and standards and contribute to the slow translation of simulation-based research into practical deployment. There is therefore a strong need for efficient, scalable, and repeatable testing methodologies that can realistically capture real-time behavior without requiring physical vehicles [2].

The objective of this thesis is to bridge the gap between simulation and deployment by developing a high-fidelity hardware-in-the-loop (HIL) testbed for V2G applications. Utilizing a dSPACE MicroLabBox II platform [4] for real-time grid and vehicle dynamics emulation, coupled with a dSPACE Smart Charging Interface (DS5366) [5] and the Keysight communication tester (SL1550A) [6], this work runs bidirectional direct current (DC) V2G charging sessions between an emulated EV and electric vehicle supply equipment (EVSE) in real time. Unlike purely numerical models, this setup exercises the real communication of a charging session, from the low-level control-pilot signaling, common to all European AC and DC charging, up to the high-level ISO 15118-20 messaging. ISO 15118-20 is the second-generation standard

and the first to support the bidirectional power transfer that V2G relies on. The setup therefore allows the communication sequences and any hardware-induced latencies to be measured, and provides a basis for the feasibility and performance testing of existing and new power profiles. By providing a repeatable environment to test ancillary service provision, this work seeks to provide a scalable alternative to costly vehicle-based field trials.

In the long term, successful validation of this approach can contribute to more efficient and standardized V2G testing practices, increase awareness of HIL-based validation methods, and support faster and more reliable deployment of V2G services. This, in turn, can strengthen the role of electric vehicles as active grid assets and enhance power system flexibility in future sustainable energy systems.

1.1 Related Work

Research on V2G systems has predominantly focused on control algorithm development and simulation-based evaluation. Nevertheless, several notable studies have extended this work through HIL testing, particularly in the context of power electronics and grid-interactive control.

Ru et al. [2] provide a comprehensive decadal review of V2G research, categorizing the technological evolution from basic unidirectional charging to sophisticated bidirectional grid-support strategies. Their analysis highlights that while optimization models and theoretical control frameworks have matured significantly, a critical gap remains in the practical validation of these systems under real-time constraints. The authors emphasize that future research must move beyond idealized simulations to address the real-world complexities of V2G, including the integration of standardized communication protocols and the mitigation of hardware-induced latencies. This review underscores the necessity of testing environments, such as the HIL testbed proposed in this thesis, to bridge the divide between theoretical grid-service models and reliable field deployment.

Yumiki et al. [7] propose an autonomous V2G control framework that simultaneously provides primary frequency control at the transmission level and voltage regulation in distribution networks. Their approach employs a physics-based ordinary-differential-equation model to regulate the spatiotemporal effects of large-scale EV charging and discharging on local grid conditions. The proposed framework is validated using numerical simulations with realistic grid models and synthetic operational data, complemented by HIL experiments to capture the effects of real-world communication and power equipment latencies. While this work explicitly acknowledges the presence of unavoidable delays introduced by communication and power electronics, the communication layer is abstracted as a generic delay source rather than being implemented using a standardized communication protocol, such as ISO 15118-20.

Dawood et al. [3] investigate the impact of communication network performance on the effectiveness of V2G frequency regulation services. Through simulation-based analysis under varying latency and packet loss conditions, the authors demonstrate

that increased communication delays significantly degrade service accuracy, as EVs fail to respond within the strict timing requirements imposed by grid operators. The study emphasizes that reliable, low-latency communication is essential for stable and effective V2G operation, particularly when coordinating multiple vehicles and aggregators. However, the analysis remains confined to simulated network conditions and does not include real-time embedded or protocol-level validation.

Makrygiorgou and Alexandridis [8] implement a comprehensive real-time dSPACE HIL setup to validate a unified control architecture for plug-in electric vehicles (PEVs) with V2G capabilities. However, while the HIL testbed successfully validates the internal control loops and transient response of the converters, it primarily functions as a single-vehicle emulation environment; it does not extend to multi-node network architectures or investigate the impact of standardized communication protocols (such as ISO 15118-20) on the overall system latency. Furthermore, their research focus remains on PEVs, diverging from the DC EV-EVSE interface focus of this thesis.

García-Martínez et al. [9] develop a specialized power EV emulator designed to validate EVSE functionality under V2G operating conditions. The proposed system utilizes a bidirectional power electronics architecture, comprising an alternating current (AC)/DC grid-side converter and a DC/DC vehicle-side stage, to emulate the high-power transients of an EV battery. Through experimental validation on a testbench compliant with CHAdeMO, the Japanese equivalent of ISO 15118-20, the authors demonstrate the emulator's ability to execute dynamic charging and discharging profiles while providing ancillary services like reactive power compensation. Similar to [7], [8] this implementation does not explicitly address the impact of network-layer vulnerabilities or the integration of higher-level V2G communication protocols like ISO 15118-20.

Simpson [10] evaluates the grid interconnection compliance and power quality of a 60-kW bidirectional charging station through a series of experimental tests based on the IEEE 1547.1 standard, including responses to abnormal voltage or frequency fluctuations. These procedures serve as a technical foundation for the experimental validation phase of this thesis, providing a benchmark for assessing the reliability of V2G systems under adverse grid conditions.

Although prior studies have addressed both simulation-based and HIL-based evaluation of V2G systems, embedded real-time validation using standardized EV-EVSE communication protocols remains limited. In particular, no existing work has demonstrated HIL-based real-time V2G control using ISO 15118-20 communication, which constitutes the primary focus and contribution of this thesis.

1.2 Purpose and Goal

The goals of this thesis project are first and foremost to commission a functioning real-time DC V2G HIL testbed on the dSPACE MicroLabBox II simulation platform that is able to conduct controlled bidirectional charging and discharging experiments. The second objective after that is to implement and evaluate bidirec-

tional DC V2G charging modes using ISO 15118-20 compliant EV-EVSE communication by interfacing with the SL1550A in electric vehicle communication controller (EVCC) mode.

By achieving these two goals, the thesis will provide quantitative experimental evidence of the capabilities and limitations of DC V2G systems in real-time operation, highlighting performance metrics, system behavior under dynamic conditions.

Some anticipated challenges in this work include developing appropriate test cases and stimuli to effectively evaluate system performance, including scenarios that may be difficult to simulate or measure. Another challenge is ensuring small enough step sizes as too large step sizes could compromise the validity of experimental results like the measured stability margins. This depends very heavily on the simulation models used and may very well prove to not be a problem at all if the very simple provided power simulations models are not replaced.

1.3 Ethical, Societal and Ecological Aspects

This work sits within the broader transition to electrified transport and renewable energy, which carries ethical, societal, and ecological implications.

Ethically, V2G raises questions of autonomy and privacy, as aggregators gain influence over when and how private vehicles charge, and the detailed charging data these services rely on is potentially sensitive and could be used to build profiles of EV owners. Additionally, since V2G uses a centralized power request sent from ancillary service providers to individual EVSEs or V2G aggregators, there is a real and almost unavoidable threat to grid stability in the event of malfunction or malicious use, so rigorous testing is needed before full adoption.

Societally, V2G can turn parked EVs into a distributed resource that helps balance the grid, absorb variable renewable generation, and defer costly investment in new generation and network capacity. When this value is shared with vehicle owners, it can lower the cost of ownership and, in turn, encourage wider EV adoption.

Ecologically, V2G can cut the reliance on fossil-fueled peaking plants, but the frequent charge and discharge it requires cycles the battery more than simple charging, accelerating degradation and bringing forward the environmental cost of battery manufacture which may in turn offset other ecological gains.

1.3.1 Use of Generative AI

Claude Opus 4.8 was used as an aid in proofreading, general troubleshooting and for helping with writing scripts. Every output was reviewed, verified, never generated completely from scratch and, in the case of scripts, tested before inclusion.

1.4 Thesis Outline

Chapter 1 presents the technical background, covers the key aspects of EV charging, including charging scenarios, AC and DC charging, connector types, smart charging,

and bidirectional charging. The ISO 15118 standard family is introduced, covering the low-level control-pilot signaling it builds on and its high-level communication, and is briefly compared with DIN 70121. The ancillary services fast frequency reserve (FFR), frequency containment reserve for normal operation (FCR-N), and frequency containment reserve for disturbance (FCR-D) are described along with their performance requirements, and the chapter closes with an overview of model-based design, from model-in-the-loop through software-in-the-loop and HIL to physical testing. Next, Chapter 2 describes the hardware and software of the testbed, the iterative workflow used to commission it, and how the experiments were conducted. Afterwards, Chapter 3 documents the system architecture, the customizations made to the pre-validated Simulink models, their interconnection in ConfigurationDesk, and the automated testing scripts. Finally, Chapter 4 presents the measured behavior of the testbed across the low-level communication, step, ramp, and power-profile tests, and the conclusion summarizes the findings and outlines directions for future work.

2

Technical Background

This chapter presents the technical background required to understand the purpose and scope of the work. It begins by introducing key concepts related to EV charging. The communication protocol evaluated by the testbed, ISO 15118, is then described to support the analysis. To provide additional context on the potential services enabled by V2G, FFR and FCR are briefly introduced and compared. The principles of model-based development and HIL are then outlined to support the understanding of the methods and results presented in subsequent chapters.

2.1 EV Charging

This section provides an overview of the key aspects of EV charging, including charging scenarios, AC and DC charging methods, connector types, smart charging strategies, and bidirectional energy transfer capabilities.

2.1.1 Charging Scenarios

Fig. 2.1 illustrates a hierarchical classification of EV charging scenarios, categorizing them into three primary methods: inductive, conductive, and battery swapping.

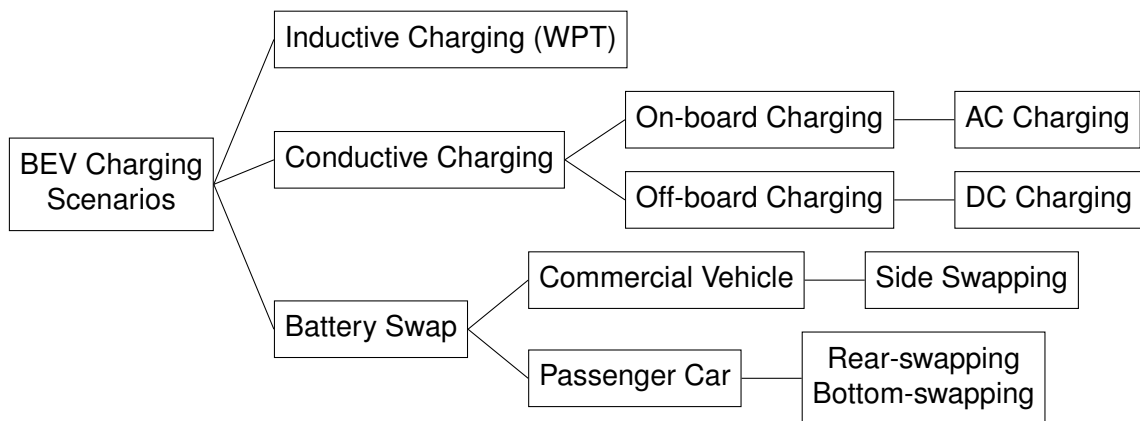


Figure 2.1: Classification of EV charging scenarios into inductive, conductive, and battery-swap methods. Sourced from [11].

Inductive charging is represented as wireless power transfer (WPT), while conductive charging is subdivided into on-board systems and off-board systems.

Of these methods, only conductive charging is relevant to this work. The ISO 15118-20 standard evaluated by the testbed in fact defines both WPT and conductive charging, but WPT is still uncommon in production vehicles and its bidirectional variant remains largely experimental, while battery swapping transfers energy by exchanging the pack rather than through a grid-connected charging interface. Conductive charging is instead the dominant and standardized method for electric vehicles today, and the bidirectional DC charging that this work evaluates is realized through it. The remainder of this chapter therefore focuses on conductive charging, and in particular on the off-board DC case used by the testbed.

2.1.2 Conductive AC

Fig. 2.2 shows the entire chain of AC charging. The charging begins with the connection of the charger plug to the inlet. This begins the charging communication between the EVSE and EV, with the protocol varying between connector types. After some initialization the contactors are closed and AC power starts flowing through the inlet. This incoming AC is then rectified by the on-board charger (OBC) which also regulates the incoming power, voltage and current into the battery according to control signals sent by the EVCC or OBC. The EV's ability to regulate this themselves makes it so that communication between the EV and EVSE does not need to be very complex.

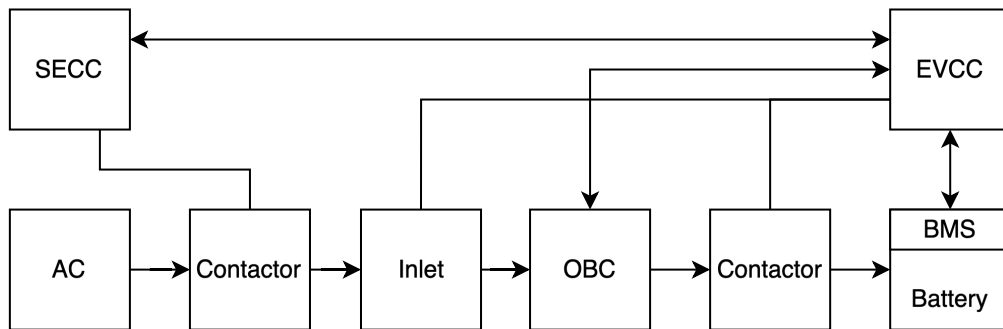


Figure 2.2: Components used in the EVSE and EV during AC charging. Bidirectional arrows refer to control/communication signals. Unidirectional arrows refer to power flow. Simple lines connectors refer to switches.

2.1.3 Conductive DC

Fig. 2.3 shows the entire chain of DC charging. DC charging shifts the primary power electronics from the EV to the EVSE, whose power supply rectifies the grid AC to DC. This DC is then supplied directly through the vehicle inlet and secondary contactors, often passing through an internal DC/DC booster if the vehicle's battery voltage exceeds the station's output capability. This allows for significantly higher power throughput due to eliminating the limitations of the OBC at the cost of requiring much faster and more advanced active monitoring by the EVSE. This requires higher-level of charging communication, and fast at that, to be able to regulate power better as to not overcharge the batteries.

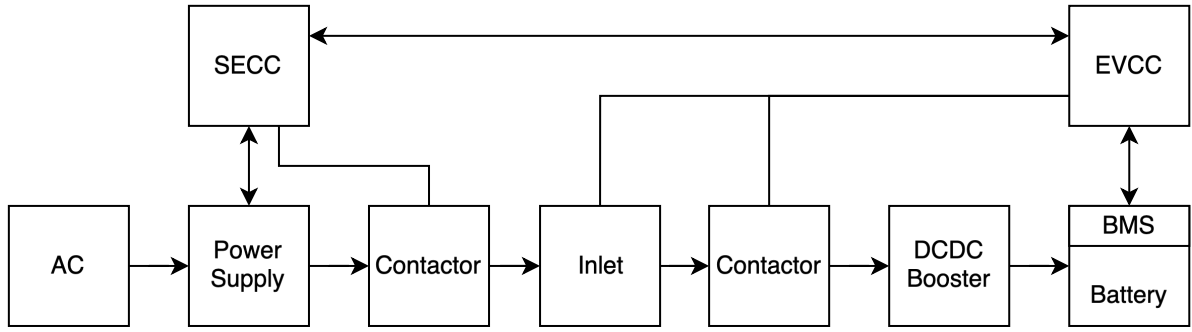


Figure 2.3: Components used in the EVSE and EV during DC charging. Bidirectional arrows refer to control/communication signals. Unidirectional arrows refer to power flow. Simple lines connectors refer to switches.

2.1.4 Charging Connectors

Table 2.1 summarizes the main AC and DC EV charging connectors used in North America, Japan, China, and Europe. For AC charging, Type 1 is used in North America and Japan, Type 2 in Europe, and GB/T in China. Tesla also operates proprietary AC connectors in some regions.

Table 2.1: AC and DC charging connectors worldwide. Sourced from [11].

	North America	Japan	China	Europe
AC	Type 1 Tesla	Type 1	GB/T Tesla	Type 2
DC	CCS 1 Tesla	CHAdeMO Tesla Chaoji	GB/T Tesla Chaoji	CCS 2

For DC fast charging, Table 2.1 shows that CCS 1 is used in North America and CCS 2 in Europe, while Japan uses CHAdeMO and China uses GB/T. Tesla’s proprietary DC connectors appear across multiple regions. The emerging Chaoji standard, listed for Japan and China in Table 2.1, is a next-generation ultra-fast charging system designed to support significantly higher power levels and improved interoperability between the two regions.

It is important to note that *CCS* and *Type* connectors are not communication protocols on their own, but rather connector systems that rely on underlying standards. Type 1 and 2 use IEC 61851-1 as its communication protocol and CCS 1 and 2 ISO 15118-2, the newer ISO 15118-20 or the somewhat older and less feature-rich DIN 70121 as its communication protocol. The other connector types define both the physical interface and their own dedicated communication protocols.

2.1.5 Smart Charging

Smart charging (V1G) refers to the controlled and optimized charging of electric vehicles by adjusting charging power and timing based on external factors such as grid

conditions, electricity prices, and user preferences. Instead of charging immediately at full power, smart charging systems manage when and how fast a vehicle charges to improve efficiency and reduce costs. The more advanced communications, such as ISO 15118-2/20, also support plug and charge (PnC) that automatically bill EV owners when they plug in the connector to charge their car [11].

2.1.6 Bidirectional Charging

Bidirectional charging extends these capabilities by allowing energy to flow not only from the grid to the vehicle but also back from the vehicle to external systems such as your home, in which case it is called vehicle-to-home (V2H), or an arbitrary load, in which case it is called vehicle-to-load (V2L). These terms commonly fall under the umbrella term vehicle-to-everything (V2X).

The most discussed among these is V2G which refer to EVs being able to supply energy back to the grid and support it in the form ancillary services [11].

2.2 ISO 15118

In 2010, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) formed the ISO 15118 Joint Working Group. This initiative brought together specialists from the automotive sector as well as the utility and energy industries to collaborate on developing a global communication standard for EV charging. Their work resulted in the widely recognized standard known as ISO 15118 that was first published in 2014 [12].

Prior to ISO 15118, its preceding standard IEC 61851 [13] did define a protocol for DC charging but was unable to conduct DC charging because it did not define the necessary high-level communication [11]. But by building on top of this old standard through the overlaying of powerline communication (PLC) based higher-level communication over the low-level analog communication of IEC 61851, ISO 15118 sought to enable DC fast charging and more advanced functionality [14]. Examples of these include PnC, V1G even V2G in the later ISO 15118-20 standard [15].

2.2.1 Comparison with DIN 70121

Before the full international standard ISO 15118 was finalized, the industry required a functional framework to support the rapid rollout of high-power infrastructure. This led German Institute for Standardization (DIN) to create DIN 70121, a German pre-standard based on early drafts of ISO 15118, that were released in 2012. While it lacks more advanced features such as AC charging support, PnC security, or V1G, it still remains for many public DC fast chargers today, ensuring backward compatibility for older EVs [11].

2.2.2 Low-level Communication

Fig. 2.4 illustrates the combined charging system type 2 (CCS2) connector standard on the vehicle side. In addition to the pins used for AC charging (L1, L2, L3, and N) and DC charging (DC+ and DC-), the connector includes two communication pins: proximity pilot (PP) and the control pilot (CP) both referenced electrically to protective earth (PE).

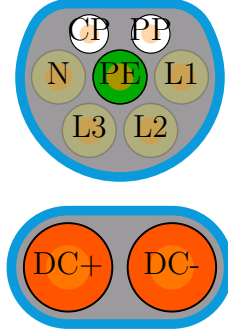


Figure 2.4: Pinouts of the CCS2 connector. From [16], CC BY-SA 4.0.

The PP pin is used to detect a valid connection between the EV and EVSE. As per IEC 61851-1, the maximum current rating of the charging cable is encoded in the resistance value between PP and PE [13]. The relationship between that measured resistance and maximum current is shown in Table 2.2.

Table 2.2: Maximum cable current rating as a function of PP resistance.

Resistance (Ω)	Maximum Current (A)
1500	13
680	20
220	32
100	63

The CP interface manages low-level communication through a 1 kHz pulse width modulation (PWM) signal with an amplitude switching between ± 12 V. The EV communicates its operational state by adjusting the positive voltage level as measured by the EVSE through a net of pull-down resistors inside the EV. The EVSE communicates the available current through the PWM duty cycle. The EV states, defined by the voltage levels between CP and PE, are summarized in Table 2.3 and the mapping of the duty cycle to maximum available current in Table 2.4.

Table 2.3: Operational states for CP signal voltage levels.

State	Voltage	Connected	Description
A	+12 V	No	Standby; no charging permitted.
B	+9 V	Yes	Vehicle detected; charging pending (B1: DC signal; B2: PWM active).
C	+6 V	Yes	Vehicle ready; charging authorized (C2: PWM active).
D	+3 V	Yes	Vehicle ready; ventilation required for indoor charging.
E	0 V	Yes	Power unavailable; potential short circuit to PE.
F	-12 V	Yes	Error state; EVSE unavailable.

Table 2.4: Control Pilot duty cycle interpretation.

Duty Cycle	Functional Meaning
< 3%	Charging prohibited
3% – 7%	High-level (digital) communication required
8% – 10%	Available current = 6 A
10% – 85%	Available current = (Duty Cycle $\times$ 0.6) A
85% – 96%	Available current = (Duty Cycle - 64) $\times$ 2.5 A
> 97%	Charging prohibited

Fig. 2.5 illustrates how the CP signal evolves over a complete charging session, combining the state and duty-cycle information of Table 2.3 and Table 2.4. Initially, in state A, the line rests at a steady +12 V and no vehicle is connected. When the cable is plugged in, the EV’s pull-down network decreases the positive level to +9 V, entering state B; at first the signal is still DC (B1), and once the EVSE begins toggling the PWM it becomes *EVSE ready* (B2). The EV then signals its readiness to charge by pulling the positive level further down to +6 V (state C), after which power transfer is authorized. Throughout states B and C the negative half of each cycle remains clamped at -12 V, while the width of the positive pulses, i.e. the duty cycle, encodes the maximum current the EV may draw, as listed in Table 2.4. When charging completes, the sequence reverses through state B back to the idle +12 V of state A.

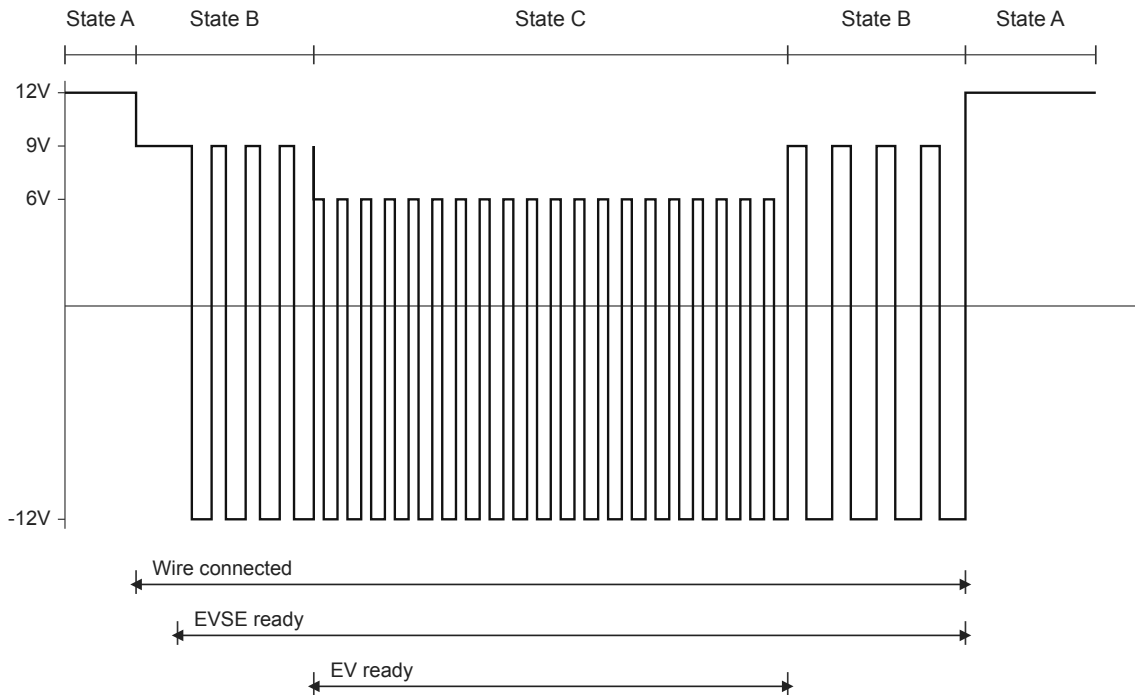


Figure 2.5: Control Pilot signal across the connection states over a charging session, showing the EV pulling the positive level from +12 V (A) to +9 V (B) and +6 V (C) while the negative half stays at -12 V.

2.2.3 High-Level Communication

After connecting the EV to the EVSE and reaching the B2 state (see Table 2.3 and 2.4) high-level communication will initiate. This stage utilizes PLC based on the HomePlug Green PHY standard [17] to transmit V2G messages over the CP line. Fig. A.1 shows the full charging sequence for ISO 15118-20.

The signal attenuation characterization (SLAC) procedure is initially executed to establish a reliable point-to-point link between the EV and EVSE. It ensures that the EV is logically paired with the physically connected EVSE by analyzing signal strength, thereby preventing cross-talk in multi-charger installations. See the full explanation and sequence in Appendix B.

After having matched the EV with its corresponding EVSE the EV needs to learn of the matched EVSE's Internet Protocol (IP) which is done via the SECC discovery protocol (SDP). See the full explanation of the protocol in Appendix C.

When the IP is known the Transmission Control Protocol (TCP) and subsequent mandatory transport layer security (TLS) handshakes can be made to in preparation for the proper V2G message session.

Finally, once the secure connection is established, the proper V2G message session begins. These messages are XML-structured data encoded using the Efficient XML Interchange (EXI), framed according to V2GTP, encrypted using TLS, and transported via TCP/IP over PLC. See the full sequence and a summary of each message in Appendix D.

2.2.4 Scheduled and Dynamic Control

ISO 15118-20 defines two control modes for the energy transfer loop that differ in where the authority over the charging power sits [15], [18].

In scheduled control mode the EVSE offers the EV one or more charging schedules of power limits over time, and the EV steers its own charging within them [15]. Authority therefore rests with the EV, while the EVSE only supplies the boundaries and monitors compliance.

In dynamic control mode the authority moves to the EVSE and the backend behind it [15]. The EV reports only its operating limits and present voltage, and the EVSE regulates its DC output to steer the delivered power toward a target set by that backend. An external actor such as an aggregator or grid operator can thus move the power in near real time [2], [7], which is what makes fast grid services feasible, since reserves such as FCR and FFR require the delivered power to change within seconds of a frequency event [19], [20].

2.3 Ancillary Services

An electrical power system must maintain a continuous balance between the power that is generated and the power that is consumed. This balance is reflected in the system frequency, which in the Nordic synchronous area is nominally 50 Hz. This area comprises the interconnected transmission grids of Sweden, Norway, Finland, and eastern Denmark, which operate as a single synchronous system at a shared frequency. A surplus of generation drives the frequency above the nominal value, while a deficit pulls it below. Because a large share of the generation still comes from synchronous machines, sudden imbalances are initially buffered by the rotational inertia of those machines. The frequency must then be actively restored, as an uncorrected deviation can drift outside safe limits and trigger protective disconnections and, ultimately, cascading blackouts [21].

To keep the frequency within acceptable bounds, the Swedish transmission system operator, Svenska Kraftnät, procures ancillary services from market participants, in particular frequency reserves. These reserves automatically adjust their power output in response to measured frequency deviations, and they are organized into a hierarchy that differs in how large a deviation each reserve reacts to and how quickly it must respond, as illustrated in Fig. 2.6. The smaller and more frequent deviations around the nominal value are handled by continuously acting reserves, whereas larger, rarer disturbances call on faster-acting reserves that engage only once defined frequency thresholds are crossed [22].

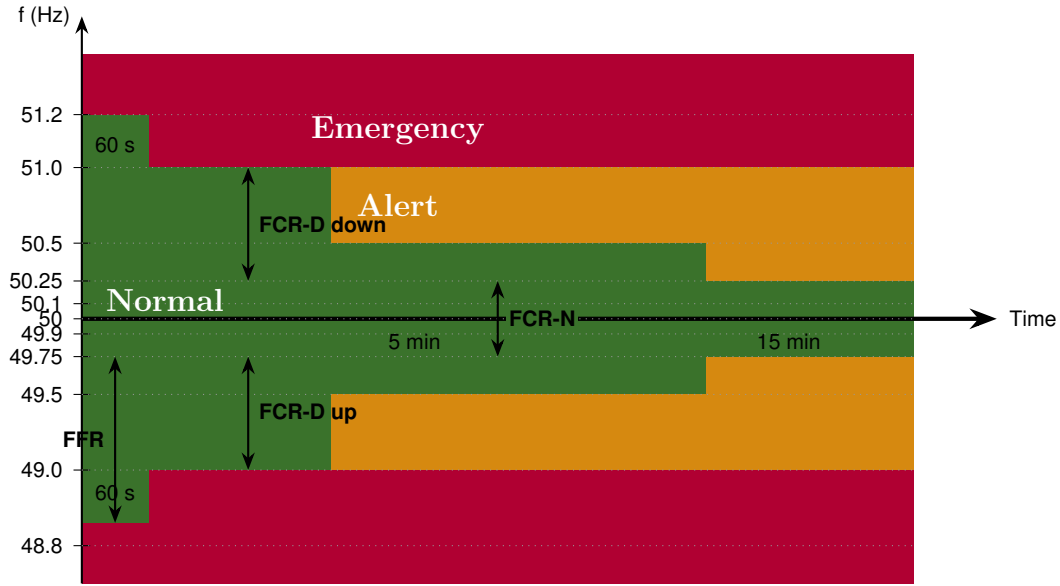


Figure 2.6: Hierarchy of FCR. Colored regions indicate the transition from Normal operation (green) to Alert (orange) and Emergency (red) states based on frequency thresholds (Hz)

This section describes the three fast reserves most relevant to this work. The FFR counters the very first and steepest part of a frequency drop, while the FCR-N and FCR-D continuously contain smaller and larger deviations around the nominal value, respectively. Their required response times, activation ranges, and endurance are compared at the end of the section.

2.3.1 Fast Frequency Reserve

According to Svenska Kraftnät, FFR is an ancillary service designed to stabilize the power system during the very first moments after a sudden imbalance, such as the unexpected loss of a large generator or transmission line. Unlike the frequency containment reserves, it provides only upward regulation, delivering its full power as a single step once the frequency drops below a chosen activation threshold, as seen in Fig. 2.7 [20].

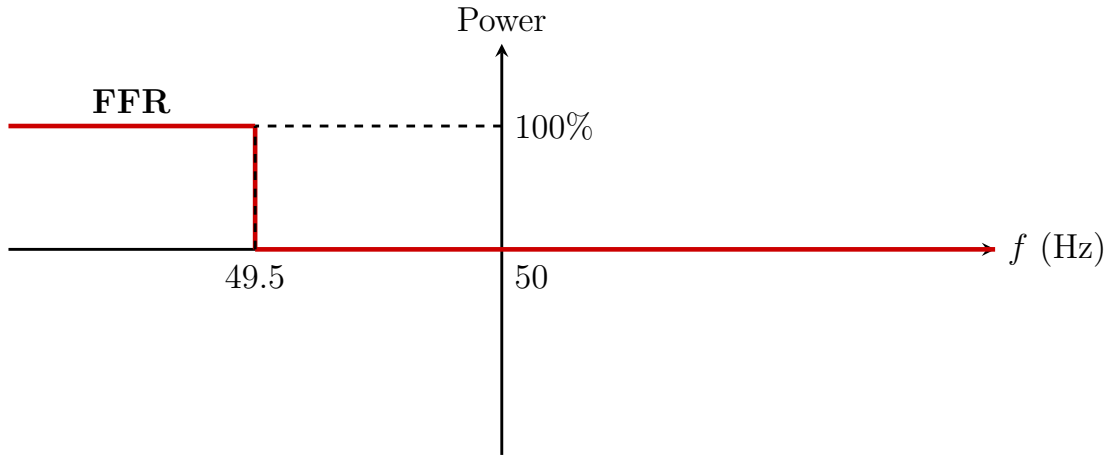


Figure 2.7: Activation power of FFR as a function of grid frequency. Full activation is delivered once the frequency falls below the chosen activation threshold, shown here at 49.5 Hz.

When system frequency begins to drop rapidly, FFR shall be fully activated within 0.7 seconds at 49.5 Hz, 1.0 seconds at 49.6 Hz, or 1.3 seconds at 49.7 Hz [20].

2.3.2 FCR-N

FCR-N is used to manage small, continuous deviations around the nominal system frequency of 50 Hz. It operates symmetrically as seen from Fig. 2.8, requiring participating resources to both increase and decrease their power output depending on whether the system frequency falls below or rises above the nominal value [19].

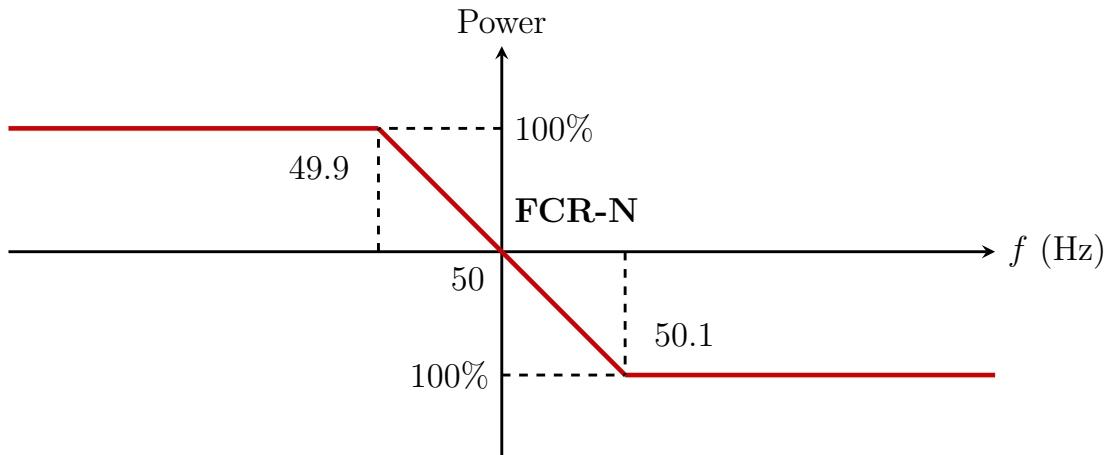


Figure 2.8: Activation power of FCR-N as a function of grid frequency.

2.3.3 FCR-D

FCR-D is activated in response to larger frequency deviations than what FCR-N can deal with. As seen from Fig. 2.6 and 2.9 FCR-D is divided into upward and

downward regulation where FCR-D Up counteracts frequency drops and FCR-D Down addresses frequency rises [19].

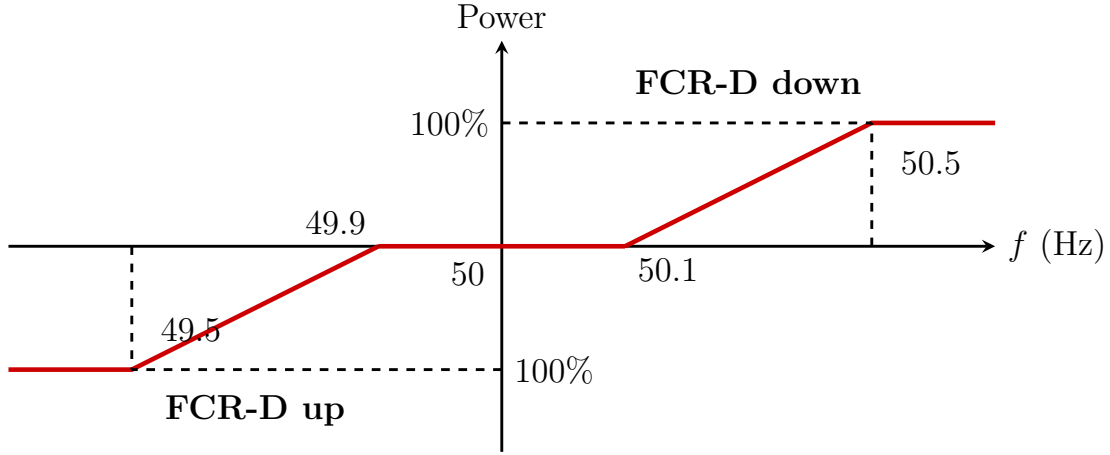


Figure 2.9: Activation power of FCR-D Up and Down as a function grid frequency.

2.3.4 Comparison and Performance Requirements

Table 2.5 compares the three reserves on their minimum bid, full-activation time and endurance requirements.

Table 2.5: Comparison of Nordic Frequency Reserves specifications. Sourced from [22].

Category	Type	Min. Bid	Full Activation	Endurance
Remedial	FFR	0.5 MW	0.7–1.3 s	5–30 s
Containment	FCR-D (Up)	0.1 MW	Linear (49.9–49.5 Hz)	≥ 20 min
	FCR-D (Down)	0.1 MW	Linear (50.1–50.5 Hz)	≥ 20 min
	FCR-N	0.1 MW	Linear (49.9–50.1 Hz)	1 hour

For a battery-backed provider such as a V2G-capable EV, sustaining a reserve draws on the battery’s state-of-charge (SoC), the share of its total energy capacity that is currently stored, so the endurance requirement translates directly into an energy demand on the battery. In the context of V2G the lower endurance requirements are attractive for battery preservation and availability for the user. This makes FFR appear most favorable on paper. But the five times higher minimum bid [22] require larger fleet aggregation which makes it more complex and therefore less attractive. This higher entry barrier may shift the economic preference toward FCR-N instead. In contrast FCR-D presents a middle ground, offering a minimum bid similar to FCR-N but with a three times shorter endurance requirement and less activation time overall due to the deadband between 49.9 and 50.1 Hz as seen from Fig. 2.9. Consequently, FCR-D is regarded in many studies as the most plausible option.

2.4 Model based Development

Traditional development relies on full-scale physical prototypes for its testing. Although this approach provides accurate results, it is costly, time-consuming, inflexible, and sometimes even dangerous. Model based development addresses these limitations by progressively integrating software and hardware around interconnected control and plant models which makes HIL testing much easier to fit within the development cycle. The following explanations are framed around MATLAB/Simulink, the toolchain used in this work and shown in the figures, but the model-in-the-loop (MIL), software-in-the-loop (SIL) and HIL concepts they describe are general and transfer to equivalent tools from other vendors. In Simulink a model is built graphically from blocks, predefined functional units such as integrators, gains and lookup tables that are wired together by signal lines. This keeps the models visual and modular, and it lets the same block diagram be both simulated directly and turned automatically into code, which is what enables the progression through the stages below.

2.4.1 Model-in-loop

The development begins with a fully simulated system, illustrated in Fig. 2.10. The dynamics and physics of the system are mathematically modeled through so called plant models that interface with the control logic, and here both are implemented as Simulink models. Because the whole system is simulated offline, this stage is called MIL.

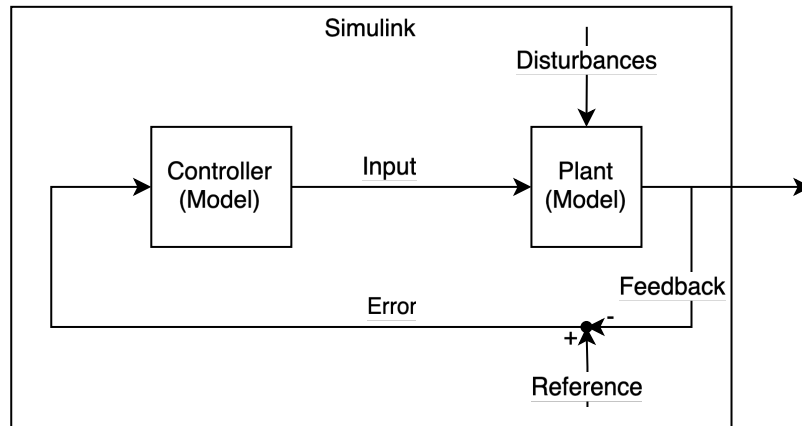


Figure 2.10: Model-in-the-loop, where both the plant models and the control logic run as simulated models.

2.4.2 Software-in-loop

The control logic can then be replaced by C code that Simulink Coder [23] generates automatically from its Simulink blocks, as shown in Fig. 2.11. The generated code is embedded back into the model as an S-function, Simulink's way of defining a block from compiled C code rather than from graphical blocks, so that it runs in place

of the original controller model. This kind of simulation is called SIL and tests the functional behavior of the generated code against the still-simulated plant.

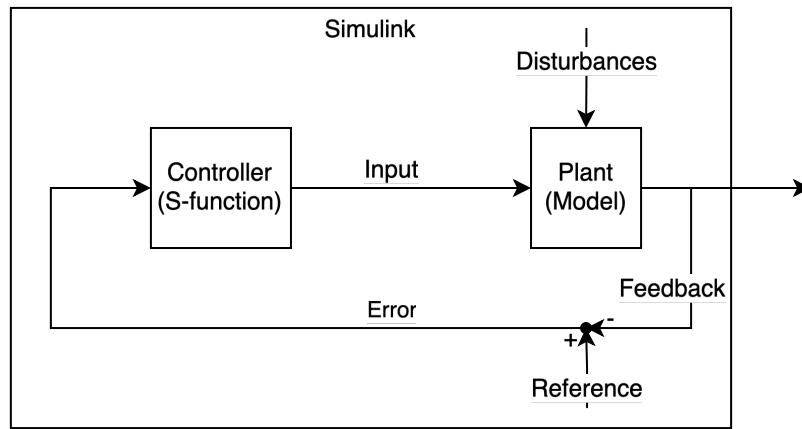


Figure 2.11: Software-in-the-loop, where the control logic is replaced by code generated from the control model and tested as software against the simulated plant.

2.4.3 Hardware-in-loop

HIL goes further still and brings real hardware into the loop, as illustrated in Fig. 2.12. The hardware brought in is usually the controller, in which case the setup is called controller-hardware-in-the-loop (C-HIL), although in principle any model in the system can be replaced by its physical counterpart.

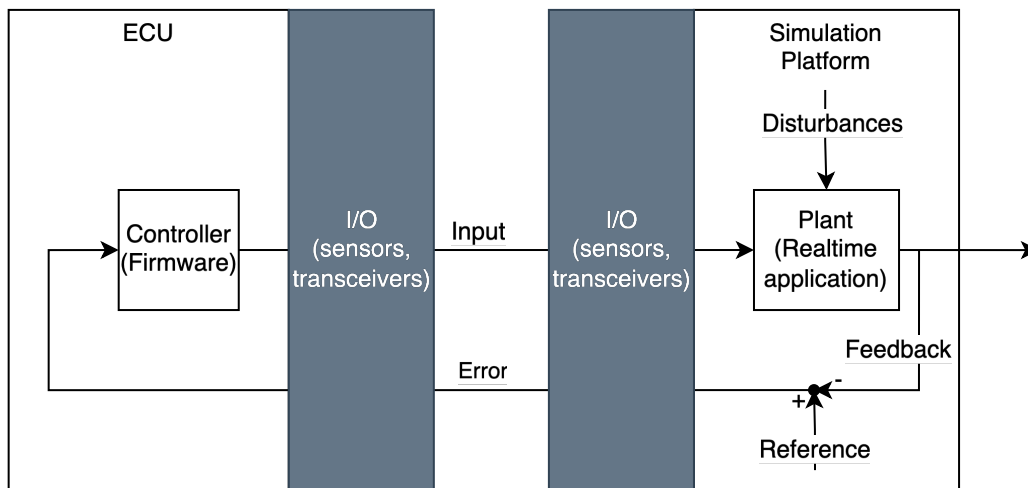


Figure 2.12: Hardware-in-the-loop, where the control logic runs on a physical controller that exchanges the simulated plant signals over a bus such as CAN.

In the C-HIL case shown here, the control logic is replaced by a physical controller loaded with the functionally validated firmware generated during the SIL stage, while the plant models now run as a real-time application on the simulation platform. Like the controller code, this application is generated from the Simulink plant models, but

compiled to run on the platform's dedicated real-time hardware at a fixed time step, so the plant advances in step with the physical controller rather than in the offline time of the earlier stages. The simulated signals and values from that platform must now travel across a bus such as CAN to reach the physical controller, which adds latency to the loop. This resembles reality more closely, since these signals typically originate from sensors that communicate with the controller over a shared bus.

2.4.4 Physical Testing

Following HIL testing, validation typically proceeds to full system prototype testing and ultimately real-world operation, where both controller and plant are entirely physical and no longer rely on real-time simulation. As the testing or development progresses the simulated components are gradually replaced with real physical parts until the full product is realized, with each hardware component typically tested individually in isolation. In the case of this testbed, that step would amount to full vehicle tests, replacing the simulated battery and power electronics and the emulated vehicle side with an actual EV.

3

Method

This chapter presents the hardware and software used to commission the HIL testbed. It first introduces the simulation platform and test equipment, then the available components and how they are interconnected, the software toolchain used to configure the models and upload them onto the system, and the iterative workflow followed during commissioning. A final section explains how the experiments were conducted.

The testbed applies the HIL principle of Fig. 2.12, though in a more elaborate form than that minimal example. Instead of a single controller it has two: the supply equipment communication controller (SECC) and the EVCC, which are both real controllers connected through an interface and exchanging genuine ISO 15118-20 communication. One of them is hosted inside the simulation platform, alongside the plant models of the battery and power electronics that are simulated in real time.

3.1 Simulation Platform and Test Equipment

This section briefly introduces the two companies whose commercial equipment makes up the testbed. dSPACE provides the real-time simulation platform and its toolchain, while Keysight provides the communication tester used on the vehicle side.

3.1.1 dSPACE

dSPACE is an engineering company specializing in HIL simulation, rapid control prototyping, and test and validation systems, founded in 1988 and headquartered in Paderborn, Germany [24]. Its real-time platforms integrate tightly with MATLAB/Simulink, which is a key reason for their wide adoption in the automotive sector [25], [26].

3.1.2 Keysight

Keysight is a test and measurement company whose Scienlab product line targets EV charging and V2G [27]. In this testbed its SL1550A emulates the EV side of the charging session [6], complementing the dSPACE platform that hosts the EVSE side.

3.2 Hardware

The HIL testbed is built from three main units, interconnected as shown in Fig. 3.1. The dSPACE MicroLabBox II is the real-time simulation platform. It runs the pre-validated dSPACE Simulink models of the battery, contactors and power electronics as the real-time plant application, and it also hosts the SECC, the controller on the charger side. The SECC communicates with the SL1550A, the physical controller that acts as the EVCC on the vehicle side, through the DS5366 that sits between them.

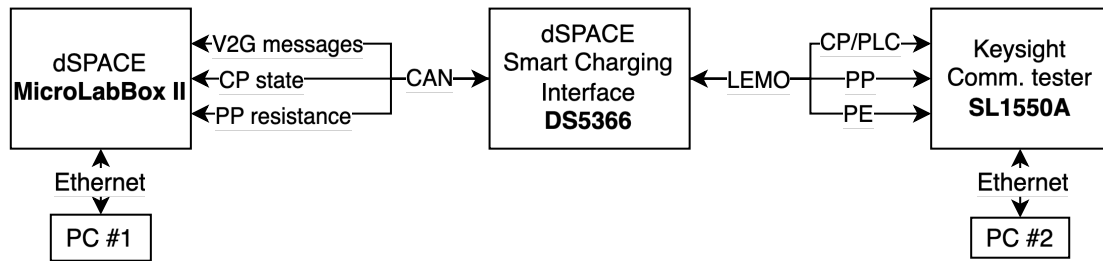


Figure 3.1: Physical interconnects of the hardware components in the testbed.

The two controllers communicate through high-level V2G messages carried over PLC according to ISO 15118-20, together with the low-level signaling of IEC 61851-1 (Section 2.2.2). The SL1550A has a built-in PLC modem and the necessary hardware for CP and PP signaling, whereas the MicroLabBox II does not. The two are therefore interfaced through the DS5366, which provides that electrical interface and connects to the MicroLabBox II over controller area network (CAN) and to the SL1550A over a LEMO cable.

Across this interface the DS5366 converts the outgoing V2G messages from CAN to TCP/IP and the incoming ones back from TCP/IP to CAN. It also measures the CP voltage and duty cycle and the PP resistance and transmits them to the MicroLabBox II over the same CAN link.

Each side is operated from its own host PC. PC 1 runs the dSPACE toolchain and controls the MicroLabBox II, while PC 2 runs the Keysight software and controls the SL1550A.

Fig. 3.2 shows the assembled physical testbed. The MicroLabBox II and the SL1550A form the left stack, with the DS5366 and its benchtop supply to the right. The split LEMO cable is open ended, with its CP, PP, and PE conductors terminated in banana connectors that plug into the SL1550A.

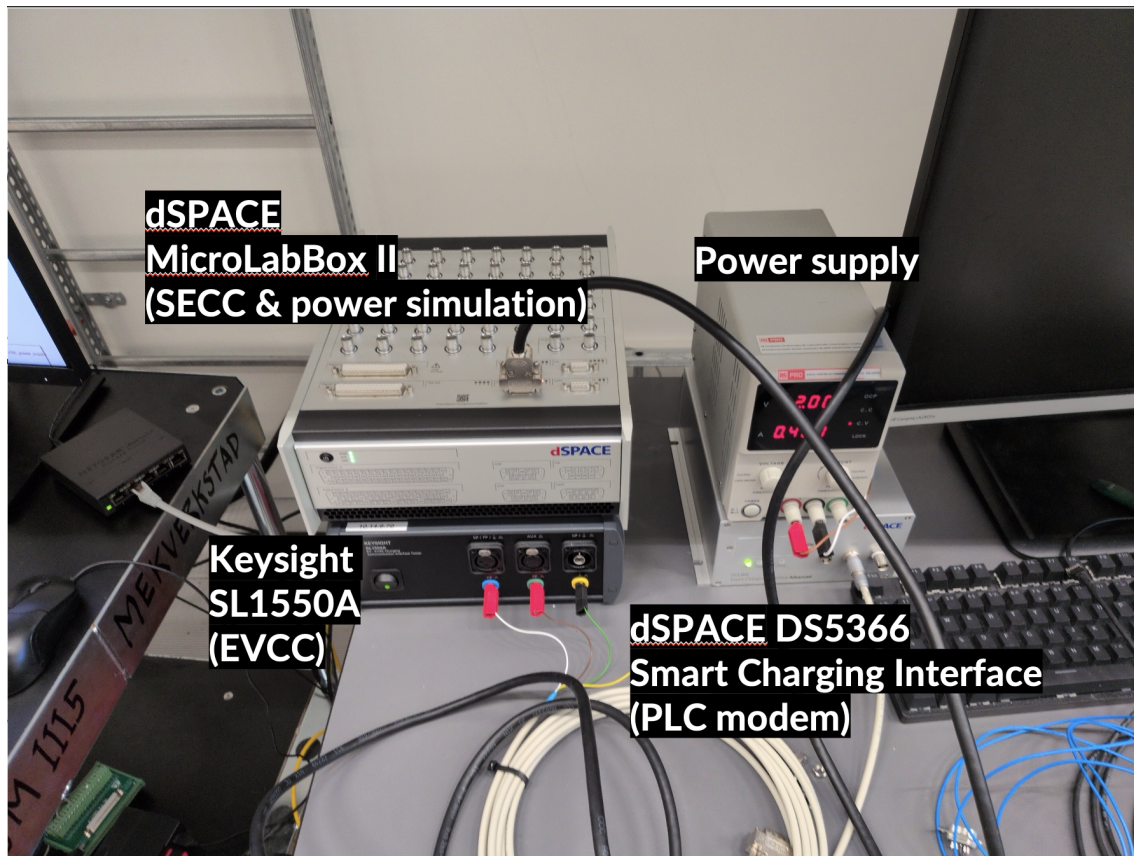


Figure 3.2: Picture of the physical testbed with annotations.

The components introduced above each have a distinct role in the testbed. Their functions are summarized in Table 3.1.

Table 3.1: Function of each hardware component in the testbed.

Component	Function
dSPACE MicroLabBox II	Real-time simulation platform. Hosts the SECC and simulates the battery, contactors and power electronics.
SL1550A	Communication tester that emulates the EVCC, with a built-in PLC modem and the CP and PP signaling hardware.
DS5366	Electrical interface between the MicroLabBox II and the SL1550A. Converts the V2G messages between CAN and TCP/IP and measures the CP voltage and duty cycle and the PP resistance.
Benchtop power supply	Provides the supply voltage for the DS5366 interface.

Continued on next page

Component	Function
CAN cable	Connects the MicroLabBox II to the DS5366, carrying the CAN messages together with the measured CP and PP values.
LEMO cable	Open-ended cable carrying the CP, PP and PE signals from the DS5366 to the SL1550A.
PC 1	Host PC that runs the dSPACE toolchain and controls the MicroLabBox II.
PC 2	Host PC that runs the Keysight software and controls the SL1550A.

3.3 Software

The software toolchain provides the environment for model development, hardware configuration, and real-time experiment control. The dSPACE side of it covers the whole path from a model to a running experiment, while the Keysight side configures the vehicle emulation.

3.3.1 dSPACE Toolchain

The dSPACE toolchain follows a model-based design flow that goes from a Simulink model to a real-time application running on the hardware, monitored from an operator interface. The three tools involved, Simulink for the models, ConfigurationDesk for the build and ControlDesk for the experiments, are all software applications that run on the host PC rather than on the MicroLabBox II itself.

The control logic and power simulations are provided as pre-validated Simulink models developed by dSPACE. These models capture the behavior of the SECC, the battery, the contactors and the power electronics, and define what the MicroLabBox II executes in real time. Their parameters, such as the battery capacity and the current and voltage limits, are set inside the models before the build. Each model exchanges signals with the outside through *inport* and *outport* blocks, which define its inputs and outputs. These ports are what get mapped to the hardware and what connect one model to another, and the design chapter refers to them when describing the customizations.

dSPACE ConfigurationDesk connects these models to the physical hardware. This configuration is held in a ConfigurationDesk *project*, which maps the model signals onto the I/O pins of the MicroLabBox II and the DS5366 interface and generates, compiles and downloads the real-time application onto the platform so that the model runs at a fixed step size. This is the step that turns the Simulink models into code executing on the MicroLabBox II rather than on a host PC.

Once the application is running, dSPACE ControlDesk provides the operator interface. Its instruments are arranged into *layouts*, the on-screen panels of gauges and controls, which together make up a ControlDesk *experiment*. From these layouts

the model variables are monitored and changed in real time, which is how power requests and limits are issued during manual operation. ControlDesk reads the signals directly from the running model and records the selected ones to an measurement data format (MDF) file for offline analysis, as described in Section 3.5. It also hosts the Python automation used to run the scripted tests of Appendix E.

3.3.2 Keysight Software

The vehicle side is controlled from the Keysight software on PC 2. Scienlab Smart Charging Emulation (SCE) drives the SL1550A, the physical EVCC, and stands in for the car. It starts and stops the charging and switches between charging and discharging.

SCE is also where the EV parameters are set, such as the battery voltage limits and the maximum charging and discharging power. The battery itself is simulated inside the MicroLabBox II, so these values act as the limits the car asks to stay within.

3.3.3 Model Parameters

The parameters used for the experiments are listed in Table 3.2. The EVSE limits are set in the ControlDesk configuration interface on the dSPACE side, while the EV limits are set on the SL1550A acting as the EVCC. The charging standard is fixed to ISO 15118-20 over CCS2 in DC mode. The EVSE is configured with generous limits so that it does not constrain the tests. The EV limits represent a modern 400 V electric vehicle with an 80 kWh pack, comparable to a Tesla Model 3, and define the operating envelope that the reserve tests work within. The battery itself is simulated inside the MicroLabBox II, so its capacity, internal resistance and 20% starting SoC are parameters of that model rather than values set on the communication tester.

Table 3.2: EVSE and EV parameters set in the simulation model for the experiments.

Side	Parameter	Value
EVSE	Maximum charge current	500 A
	Maximum discharge current	500 A
	Maximum voltage	1000 V
	Maximum power	150 kW
EV battery	Capacity	80 kWh
	Start state of charge	20%
	Maximum voltage	410 V
	Minimum voltage	300 V
	Maximum charge current	400 A
	Maximum discharge current	400 A
	Maximum power	150 kW
Pack	Total resistance	0.01 Ω

Important to note is that SL1550A does not simulate the battery nor any other power

simulation. It therefore does not report its internal battery voltage and other internal measurements to the dSPACE MicroLabBox II throughout the charging loop cycles as it should. It does however present its present battery voltage during precharge phase(see appendix D according to the “initial battery voltage” parameter set inside the Keysight software. This makes the EVSE simulated inside the MicroLabBox II to match that voltage during the precharge, which if it deviates from the internal simulated battery voltage, causes a large rush of current when the contactor closes after it goes over to the charge loop phase. Therefore the parameters set by battery model inside dSPACE and the initial battery voltage set by Keysight must be the same.

3.4 Iterative Workflow

Fig. 3.3 shows the overall iterative workflow followed to commission the testbed. The given ConfigurationDesk project is built into a real-time application (RTA) and, once the build succeeds, loaded together with the provided ControlDesk experiment. The provided ControlDesk experiment includes diagnostic layouts that report the state of the hardware, such as whether the DS5366 is connected and whether its firmware is up to date. Once these diagnostics pass, a complete charging communication cycle is established and a simple charging experiment is run to check the charging behavior. The Simulink models are then reconfigured and the loop is repeated until the behavior is correct, after which the final experiments are conducted.

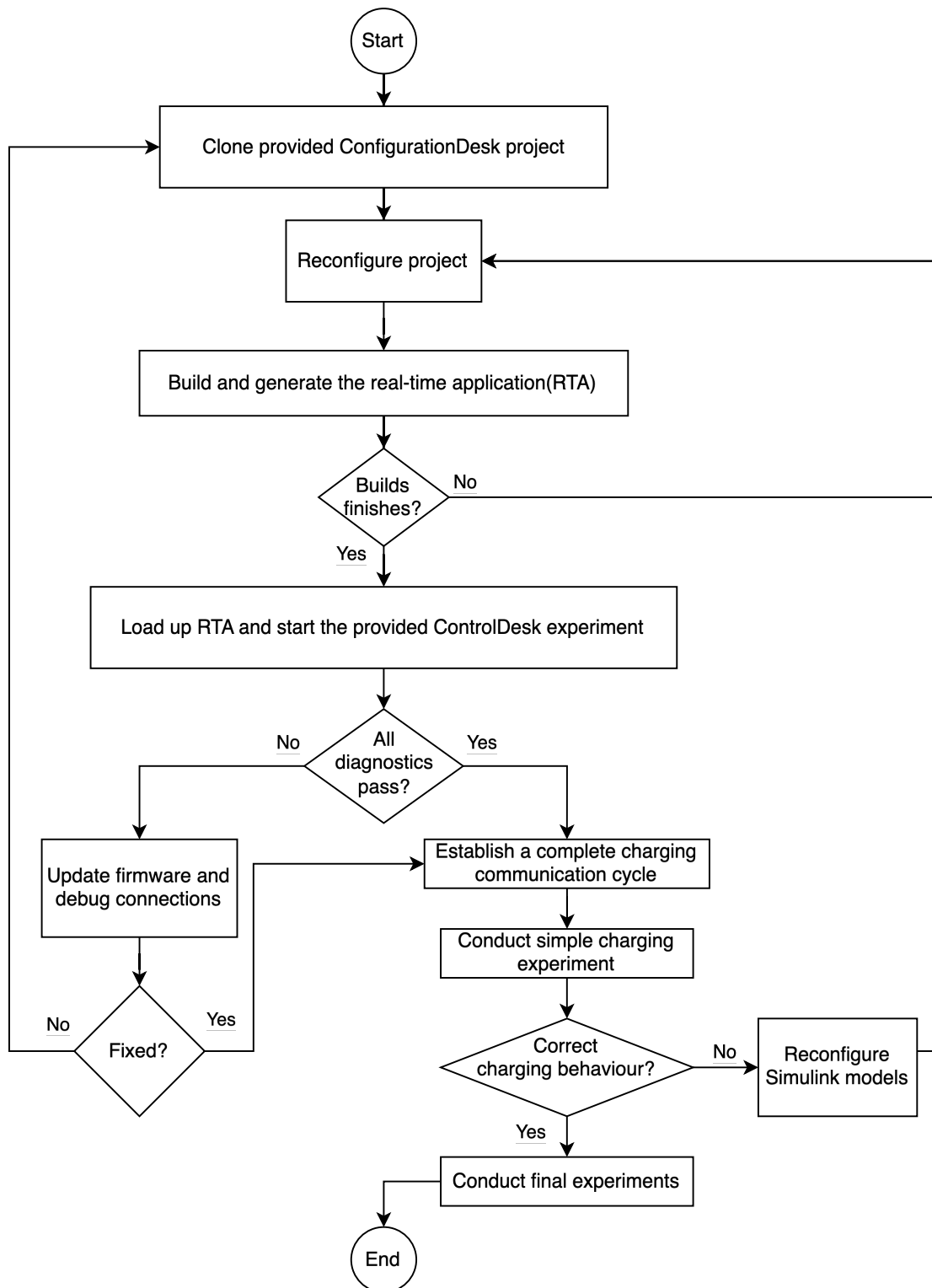


Figure 3.3: The iterative workflow followed to commission the testbed and conduct the experiments.

3.5 Experimentation

The experiments used the ISO 15118-20 dynamic control mode, commanding the charging power through the EVSE dynamic external power request so that the unit's tracking of a fast-changing reference could be measured. They were conducted in two ways, manually and automatically, and in both the ControlDesk recorder captured the measured signals for offline analysis.

In the manual runs the charging power was commanded from ControlDesk by writing directly to the dynamic power request. Under the ISO 15118-20 dynamic control mode this single setpoint drove power in both directions, so a positive request charged the EV from the grid and a negative request discharged it back. The basic charge and discharge, step, and ramp tests were carried out this way, by holding, stepping or ramping the request.

The automated runs instead drove the request from a Python script that replayed a power profile in real time. A day of recorded grid frequency was mapped, sample by sample, to a profile that emulated an EV providing FCR-N, FCR-D or FFR, as described in Section 4.4. Replaying a full day this way exercised the ancillary services and revealed quantities such as the SoC at the end of the day, and any failure to meet the required response time.

4

Design

This chapter presents the design of the real-time system that runs on the testbed. It introduces the Simulink models and how they are interconnected, documents the customizations made to adapt the pre-validated models to the physical SL1550A, and describes how the models are connected in ConfigurationDesk. A final section covers the automated scripts used to drive the experiments and process their results.

4.1 Model Interconnects

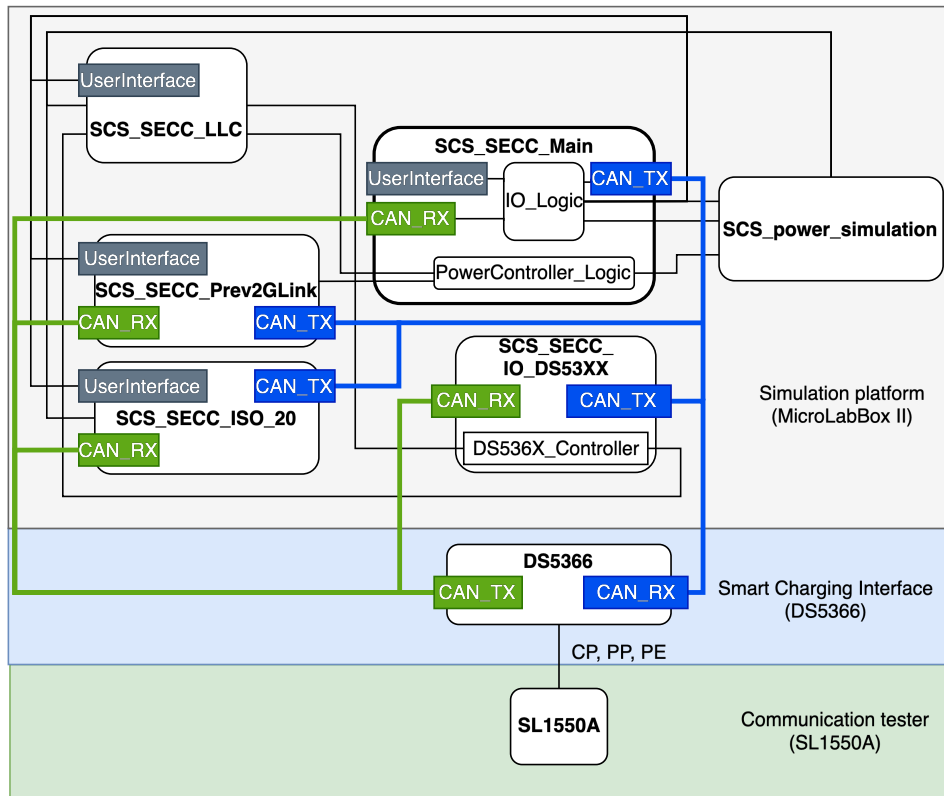


Figure 4.1: CAN interconnects between the Simulink models on the MicroLab-Box II and the DS5366. Blue links carry the CAN sent out towards the SL1550A and green links the CAN returning from it; the **CAN_TX** and **CAN_RX** labels denote the sending and receiving models rather than physical ports.

The SECC is realized as several Simulink models that run together on the MicroLabBox II, whose hardware and interconnection are described in Section 3.2. Most of them implement the SECC logic, while one emulates the battery and power electronics it drives. Table 4.1 lists each model and its role; since they were provided pre-validated by dSPACE, they are summarized rather than documented in full.

Fig. 4.1 shows how these models are wired to one another and how they exchange CAN with the DS5366, the smart charging interface that the SL1550A is connected to. The models never reach the SL1550A directly; the DS5366 sits between them and converts the CAN to and from the ISO 15118-20 traffic it exchanges with the SL1550A (Section 3.2). The blue links carry the CAN sent out by the MicroLabBox II towards the SL1550A, and the green links the CAN returning from it, so the two colors separate the outbound and inbound traffic. The `CAN_TX` and `CAN_RX` labels are not physical ports but mark which models originate and which receive these messages.

Table 4.1: Summary of the Simulink models that make up the SECC simulation.

Model	Role
<code>SCS_SECC_Main</code>	Central EVSE controller that handles the DS5366 I/O logic and the power-controller logic.
<code>SCS_SECC_PreV2GLink</code>	Manages the connectivity to high-level communication, hosting the SDP server and the SLAC pairing and establishing the TCP connection.
<code>SCS_SECC_LLC</code>	Implements the state machines for low-level (Control Pilot) communication.
<code>SCS_SECC_ISO_20</code>	Handles the high-level ISO 15118-20 protocol stack and the bidirectional V2G message exchange.
<code>SCS_power_simulation</code>	Plant model of the battery and power electronics (power supply, DC/DC booster, and contactors).

4.2 Customized Simulink Models

The provided Simulink models were pre-validated to work but only as a self-contained set, with dSPACE emulating both sides of the charging session. The models therefore had to be customized to create an interface for the external SL1550A EVCC. At the same time, the control signals that the missing dSPACE EVCC would otherwise produce had to be supplied instead, either as constant block values or rewired from elsewhere. Finally, to ensure correct charging behavior, the charge-progress flag has to be wired to the contactor inside the power simulation model.

Fig. 4.2 shows the `SCS_power_simulation` model and its submodels after all the customizations, and the following subsections describe the changes made to it.

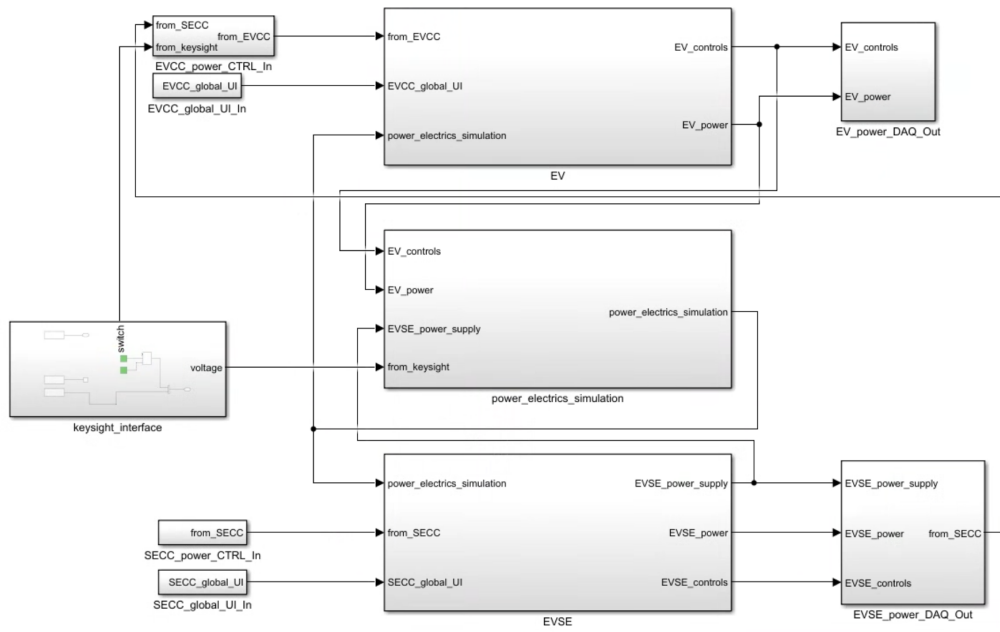


Figure 4.2: The SCS_power_simulation model and its submodels.

4.2.1 Power-control Restbus

The power-control signal from the EVCC is handled in its own subsystem, shown in Fig. 4.3. Since the EVCC now lives in the SL1550A, the model's own EVCC signals arriving on the `from_EVCC` input are terminated, and restbus signals take their place. The Keysight EVCC instead supplies the maximum DC current and the maximum and minimum DC voltage over the `from_EVSE` input and the power-relay command over `from_keysight`. The values it does not provide are fixed as constants, with the maximum AC current set to 20 A, a nominal value since no AC charging takes place, the insulation monitoring device enabled and the AC/DC configuration set to DC.

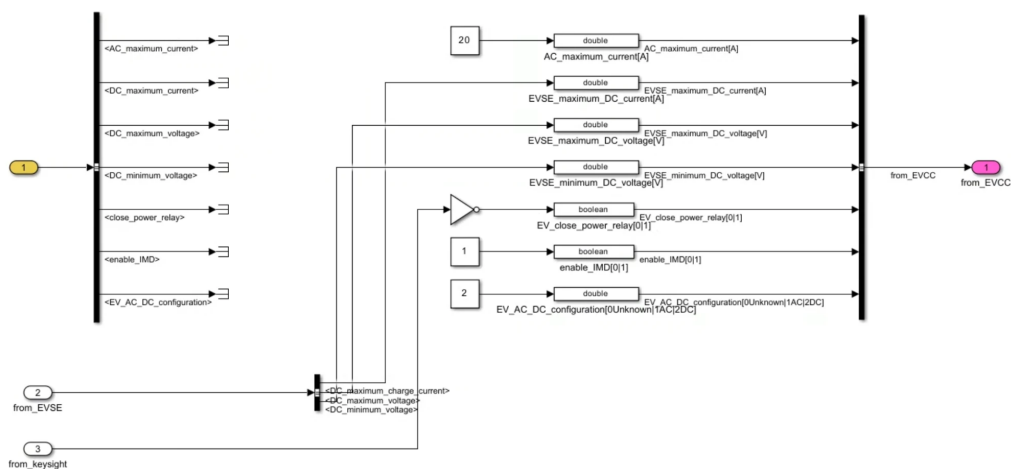


Figure 4.3: Restbus simulation of the power-control signal from the EVCC.

4.2.2 Charge-state Extraction

Because the EVCC now resides in the SL1550A, the power simulation has no direct knowledge of a charge driven by that external unit. To recover it, the charge progress and the EV present voltage are unpacked from the decoded ISO 15118-20 request messages (Appendix D) and exposed to the `SCS_power_simulation` model through ConfigurationDesk outputs (Section 4.3.1), as shown in Fig. 4.4. The charge-progress flag is then inverted to drive the relay state in the power-control subsystem of Fig. 4.3, so that the simulated contactor follows a session driven by the SL1550A.

The present-voltage path was meant to feed a battery emulated on the SL1550A back over the charge loop, but the SL1550A reports it only during precharge, so it was left unused and the power simulation keeps its own battery model.

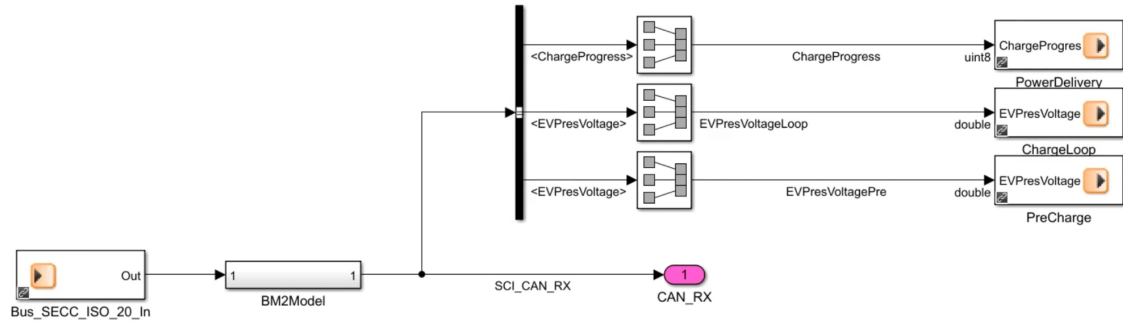


Figure 4.4: Extraction of the charge-control signals from the ISO 15118-20 message bus received by the SECC.

4.2.3 Charging-standard Selection

The charging standard has to agree with the one the SL1550A is configured for before a session can be established. Fig. 4.5 shows the `EVCC_global_UI_in` subsystem, where the `Ctrl_charging_standard` input from the user interface is terminated and a constant is cast to the `SCS_EV_charging_standard` enumeration. This fixes the standard to ISO 15118-20 DC charging, so the model always starts in the mode the physical setup uses rather than depending on a manual selection. The remaining interface signals pass through unchanged to the output bus.

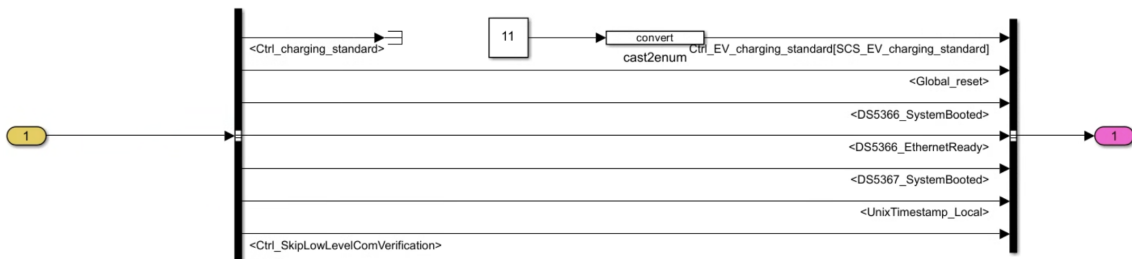


Figure 4.5: The `EVCC_global_UI_in` subsystem, which pre-selects ISO 15118-20 DC charging.

4.3 dSPACE ConfigurationDesk Setup

ConfigurationDesk maps the Simulink model I/O onto the physical MicroLabBox II and DS5366 channels and generates the real-time application executed on the platform.

4.3.1 Intermodel Communication

The outputs added in Section 4.2 only expose the extracted signals. They still have to be connected to the model that consumes them, which is done in ConfigurationDesk rather than in Simulink. Fig. 4.6 shows the resulting intermodel communication, where each data output block of the SCS_SECC_ISO_20 model is wired to the matching data input block of the SCS_power_simulation model.

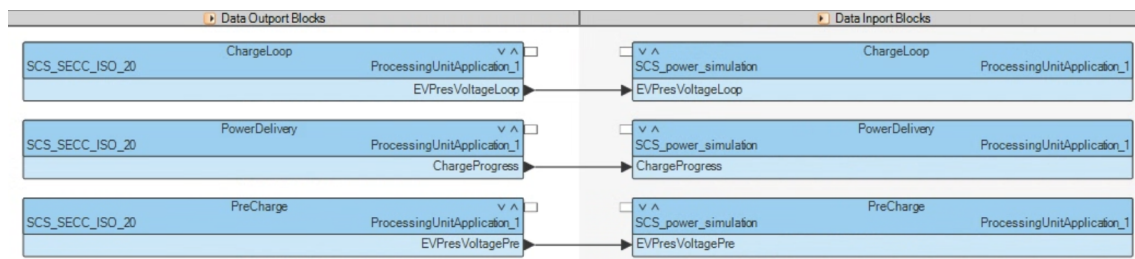


Figure 4.6: Intermodel communication between the SCS_SECC_ISO_20 and SCS_power_simulation models in ConfigurationDesk.

4.4 Automated Testing Scripts

ControlDesk exposes a Python automation interface that drives the real-time application without manual interaction. This section outlines the scripts developed to execute the experiments and to acquire their results. The two scripts that make up this pipeline are listed in full in Appendix E.

4.4.1 Frequency-to-power Profile Generation

The automated ancillary-service tests are driven by a power profile derived from real grid frequency measurements. A day of recorded frequency data is read from a CSV file and mapped, sample by sample, to an activation level for each reserve following the characteristics introduced in Section 2.3. For each sample the per-reserve activations are combined into a net request and scaled by the rated power to give the power setpoint, which is written to a CSV file that the ControlDesk playback script streams to the EVSE's dynamic power request (Appendix E.2). The full generation script, including the activation function of each reserve, is given in Listing E.1.

4.4.2 Variable Access

Model parameters and signals are reached through the same automation interface. Each variable is retrieved by its path and accessed through its converted value, so a setpoint such as the dynamic power request is applied by assigning to that value, and a measurement is obtained by reading it, as shown in Listing 4.1.

```
1 evse = Application.ActiveExperiment.Platforms[0].Platforms[0].  
    ↪ ActiveVariableDescription.Variables  
2 power_req = evse.ItemByPath("urn:dspace:variable?ModelVariable=SCS_SECC_ISO_20  
    ↪ /.../Dynamic_ext_power_request")  
3  
4 power_req.ValueConverted = 5.0          # write a 5 kW setpoint  
5 value = power_req.ValueConverted        # read back the current value
```

Listing 4.1: Accessing and changing a model variable through the ControlDesk automation interface.

4.4.3 Post-processing and Metrics

Each experiment is captured by the ControlDesk recorder as a single MDF file holding the requested power together with the measured signals. This file is the starting point for all post-processing. It is read back offline with the extraction script of Appendix E.4, which selects the signals of interest and resamples them onto a common time base to give a table indexed by time, from which the tracking metrics reported in Chapter 5 are computed and the result figures are plotted.

5

Results

Four types of experiments were conducted to characterize the behavior of the testbed. The basic charge and discharge experiments exercise the low- and high-level communication and validate the charging behavior. Step tests evaluate the transient response to sudden changes in power demand, ramp tests assess the ability to track a linearly varying power reference, and power-profile tests validate the ability to follow a predefined time-varying trajectory representative of real ancillary-service provision such as FCR-N and FCR-D. The chapter closes with a summary that compares the measured performance against the reserve requirements.

5.1 Low-level Communication

Fig. 5.1 shows a complete charging session, with the control pilot states defined in Table 2.3. The session starts in A1 with no vehicle connected, moves to B1 on connection, and reaches B2 once the EVSE applies the PWM signal, which is what allows the high-level communication to begin. The hatched interval carries the SLAC pairing, the SDP discovery, the application protocol handshake and the TCP connection of Appendices B and C, run without TLS. These exchanges stay between the DS5366 and the SL1550A, so the interval is inferred from the gap before the first V2G message rather than measured. The state then changes to C2 and the DC contactor closes.

The cable check then raises the supply target to 1000 V, reaching about 720 V, to verify the insulation before the battery is connected, and the precharge matches the supply to the battery voltage of about 372 V. The request is held at zero for this run, so the run exercises the communication sequence rather than the transfer of power. The current stays at zero apart from a brief peak at the start of the charge loop, driven by the small remaining difference between the supply and plug voltages.

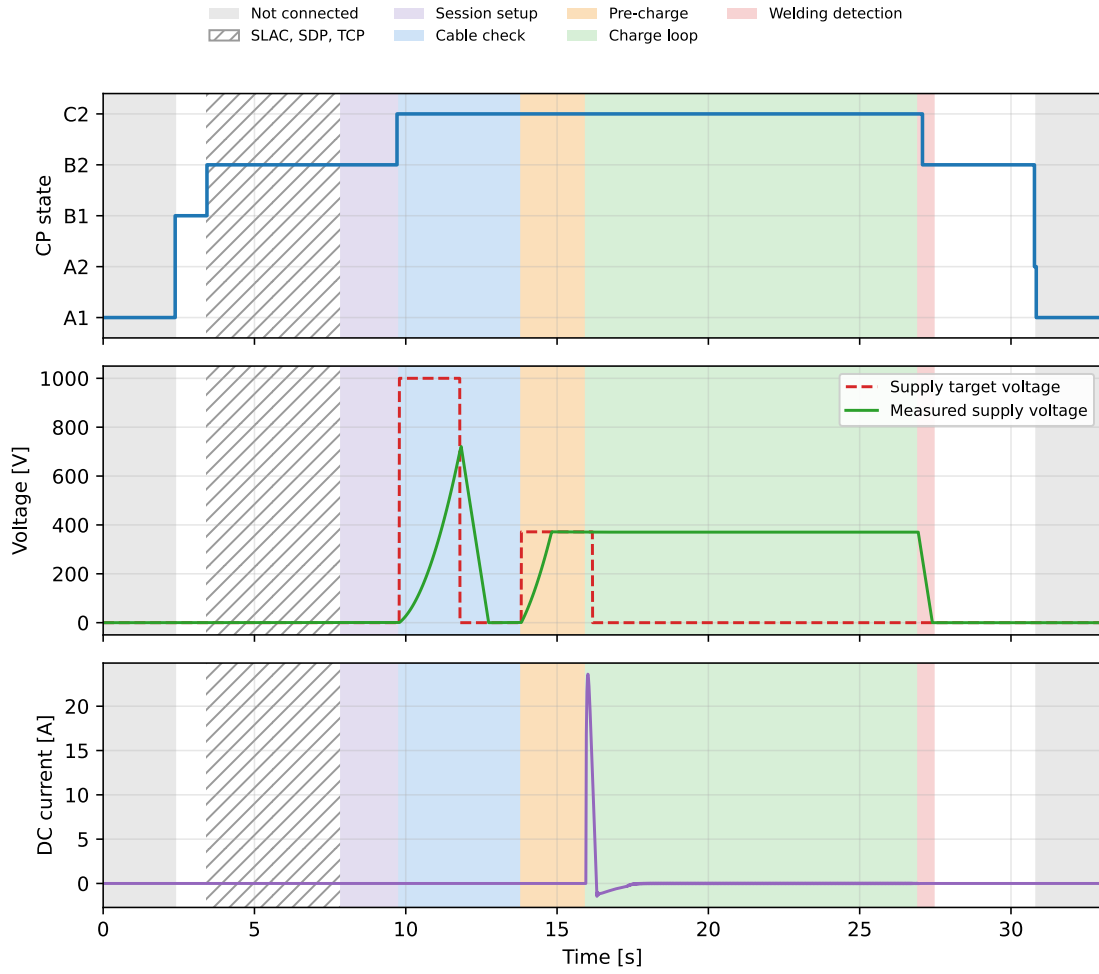
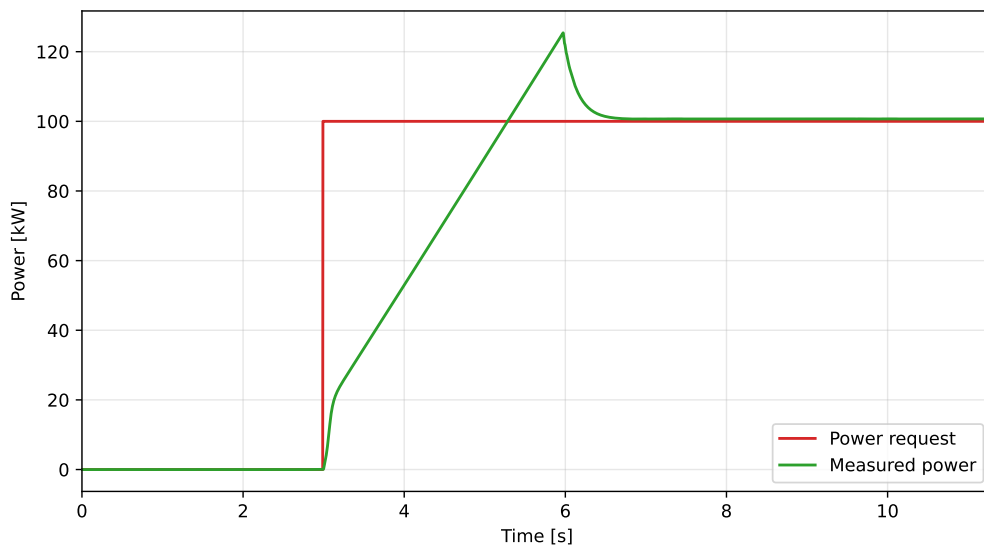


Figure 5.1: Control pilot states, DC supply target and measured supply voltage, and DC current over a complete charging session. The shaded intervals mark the phases of the session, the gray intervals the periods with no vehicle connected, and the hatched interval the exchanges that the recorded signals do not cover. The color shadings are explained at the top of the figure.

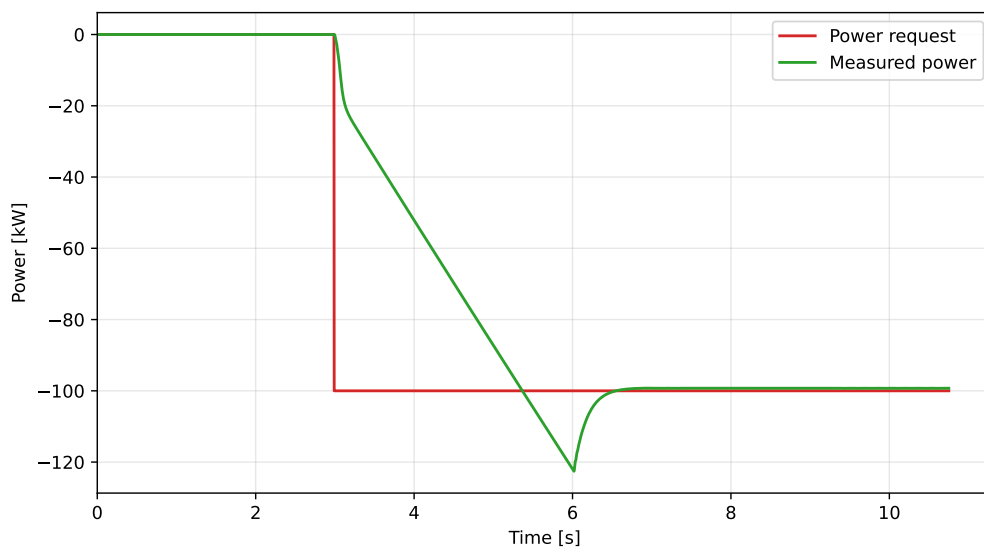
5.2 Step-response

Fig. 5.2a shows the response to a step from zero to 100 kW in the dynamic power request, the power that the ancillary service provider asks of the EVSE to charge or discharge the battery of the connected EV. A positive request charges the battery from the grid and a negative request discharges it to the grid. The measured power does not follow the step directly. It rises at a limited rate of about 37 kW/s and first reaches the requested 100 kW roughly 2.3 s after the step. It then overshoots to about 125 kW, close to 25% above the final value, before settling within 2% of that value about 3.4 s after the step. The steady output is 100.7 kW, so the request is met with an offset of less than 1%. The shape of this response, the limited slew rate together with the overshoot and settling, is determined by the built-in slew-rate limit and PID-parameters of the power controller model.

Fig. 5.2b shows the corresponding step in the discharging direction, from zero to -100 kW, drawn on the same time axis so that the two responses can be compared directly. The behavior mirrors the charging case. The measured power slews at about 36 kW/s, first reaches -100 kW roughly 2.4 s after the step, overshoots to about -123 kW, and settles within 2% about 3.4 s after the step at -99.3 kW, again within 1% of the request. The near-identical slew rate, overshoot and settling time confirm that the same slew-rate limit and PID-parameters of the simulated power supply shape the response in both directions.



(a) Charging



(b) Discharging

Figure 5.2: Step-response of the EVSE for (a) charging and (b) discharging, on the same time axis. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

5.3 Ramp-response

The ramp test increases the dynamic power request linearly and records the measured output, shown in Fig. 5.3. The request rises from zero to 10 kW at about 2 kW/s and the measured power follows it over the whole ramp and reaches the same 10 kW, so no power limit is met within the tested range. While the request is ramping the measured power sits below it by about 0.27 kW. At a constant rate of change this offset is the product of the ramp rate and the response lag, which gives a lag of about 130 ms. Shifting the measured power forward by that amount reduces the root-mean-square deviation from 0.23 to 0.03 kW, so the deviation during the ramp is almost entirely the delay of the power stage and not an error in amplitude. The offset closes once the request settles at 10 kW.

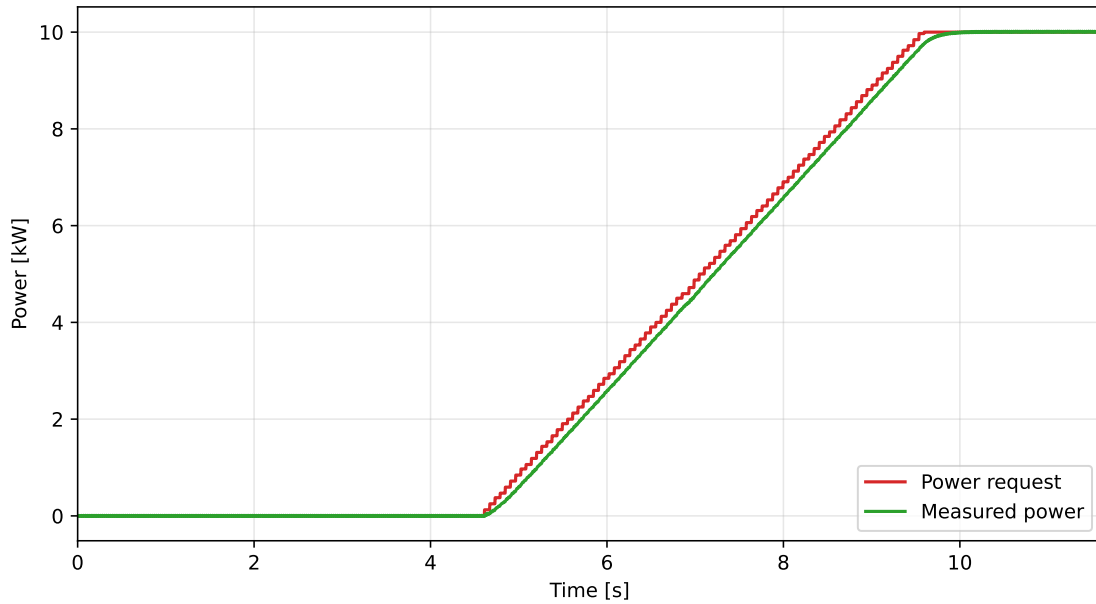


Figure 5.3: Ramp-response of the EVSE. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

5.4 Power Profile

The power-profile tests replay a full day of ancillary-service operation and records how the EVSE follows the requested profile. Each profile is derived from the recorded grid frequency of 2026-02-09, shown in Fig. 5.4, by applying the reserve characteristics of Section 4.4.1. The frequency stays close to the nominal 50 Hz for most of the day and leaves the FCR-N band only briefly, which determines how often each reserve is activated. The frequency and FCR-N figures span the full day from 00:00 to 24:00, while the FCR-D figures are zoomed to the intervals in which those reserves activate.

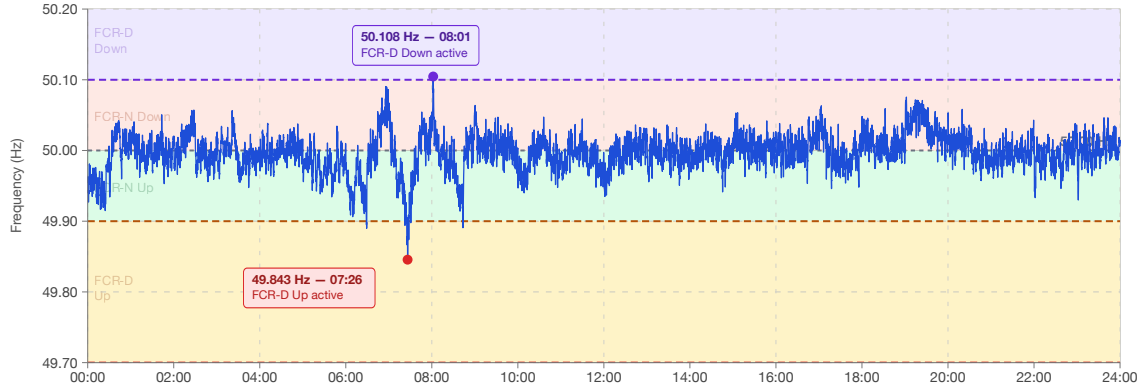


Figure 5.4: Recorded Nordic grid frequency over the full day of 2026-02-09, with the shaded bands marking each reserve’s activation range. Frequency data from [28].

5.4.1 FCR-N

Fig. 5.5 shows the resulting FCR-N request over the day. Since FCR-N counters the small continuous deviations around 50 Hz, it is active almost everywhere and tracks the frequency symmetrically, which gives a nearly continuous alternation between charging and discharging.

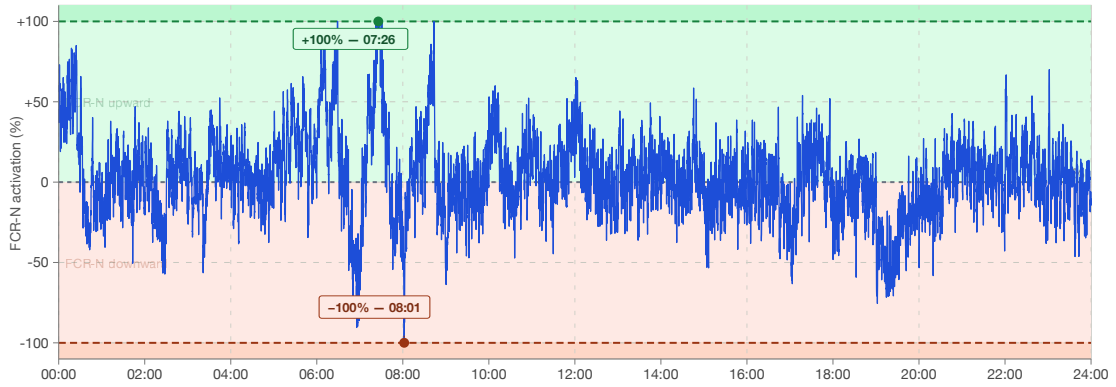


Figure 5.5: FCR-N activation over the full day of 2026-02-09, as a percentage of rated power derived from the frequency, with the shaded bands marking upward and downward regulation. Frequency data from [28].

A four-minute window around 13:05 was replayed onto the dynamic power request at the rated power of 100 kW with the same raised EV limits, and the measured EVSE power recorded, shown in Fig. 5.6. The window was chosen because the reserve alternates continuously there, with 96 changes of direction and roughly equal

time spent charging and discharging, so it exercises the transition between the two directions that the FCR-D tests do not. The request swings between about -24 and $+43$ kW and the measured power follows it over the whole window, with a mean absolute deviation of 0.74 kW and a root-mean-square deviation of 0.98 kW, which is 1.7% and 2.3% of the peak. The energy delivered is within 0.3% of the energy requested.

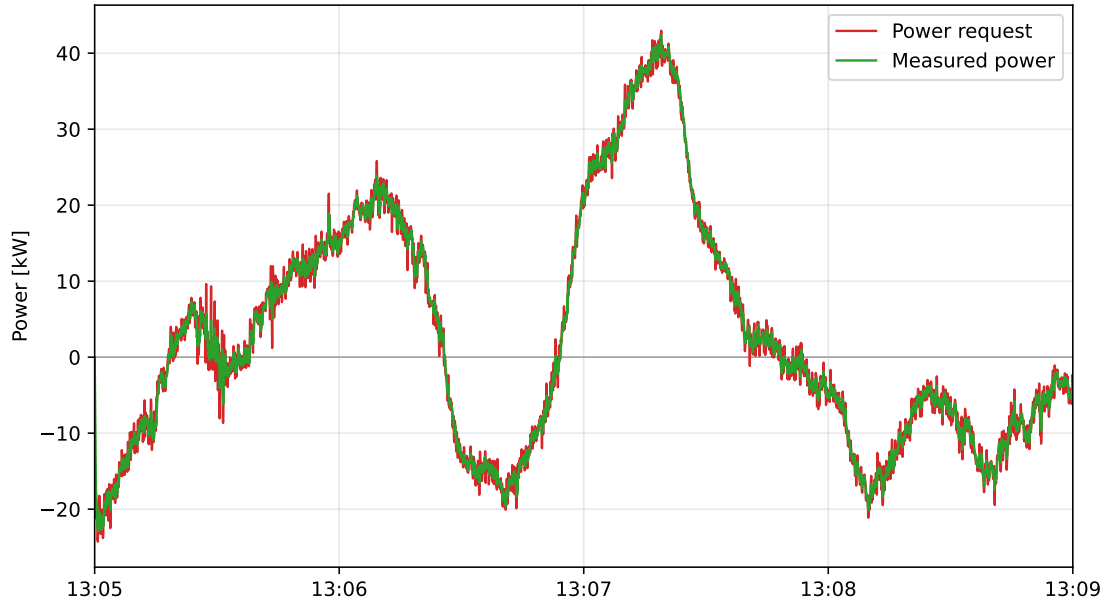


Figure 5.6: Measured FCR-N response for the four-minute window around 13:05 of the 2026-02-09 activation. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

Fig. 5.7 plots the difference between the two, the tracking error taken as the measured power minus the request. It stays within about ± 7 kW and centers on zero, and its largest excursions line up with the reversals: within a quarter of a second of a change of direction the mean deviation rises to 1.11 kW against 0.71 kW elsewhere, and the worst case of 7.0 kW also falls at a reversal. Reversing between charging and discharging is therefore the least accurate part of the operation, although each excursion is short lived and the energy is still delivered. The opening two seconds are excluded from the figure and from these numbers, since the window opens at an activation of about -20 kW rather than at zero, so the replay begins with a step that the supply cannot follow at once, an artifact of where the window was cut rather than a property of the reserve.

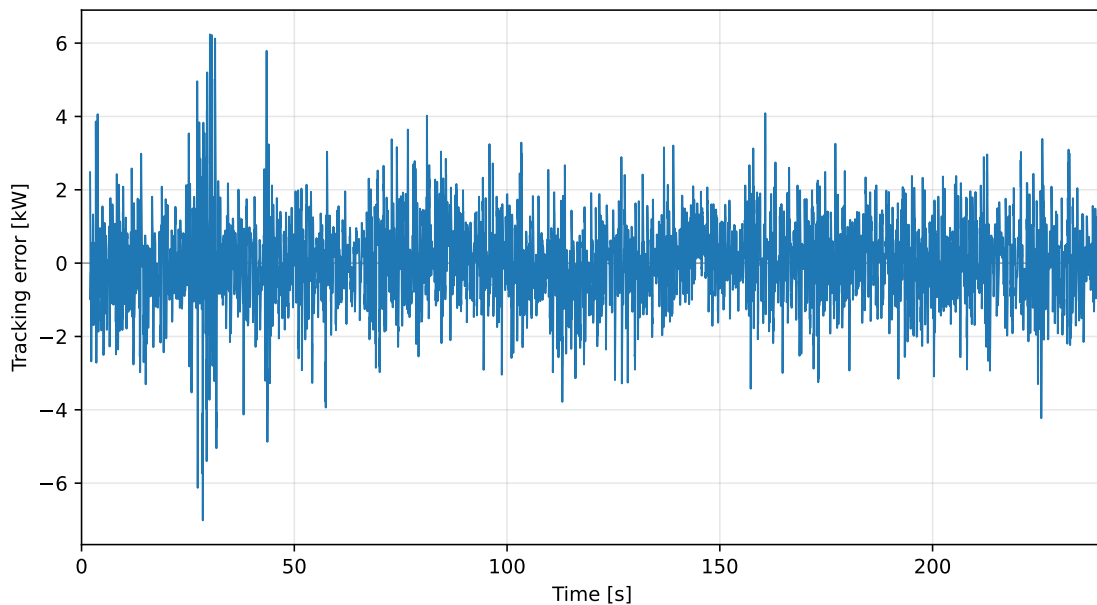


Figure 5.7: FCR-N tracking error over the four-minute window, taken as the measured EVSE power minus the dynamic power request. The opening two seconds are excluded to omit the startup step of the replay.

Fig. 5.8 shows the battery state of charge over the same window. It follows the integral of the delivered power, rising while the reserve charges and falling while it discharges, so the alternation of the request appears as a slow oscillation of a few tenths of a percentage point. Over the four minutes the state of charge moves from 18.60% to 18.81%, a net gain of about 0.22 percentage points, because the energy requested over this particular window is slightly positive rather than balanced. FCR-N is symmetric in how it responds to frequency but it is not necessarily neutral in energy over a short window, so the state of charge drifts. Over a longer period of operation that drift is what would limit how long the reserve could be sustained.

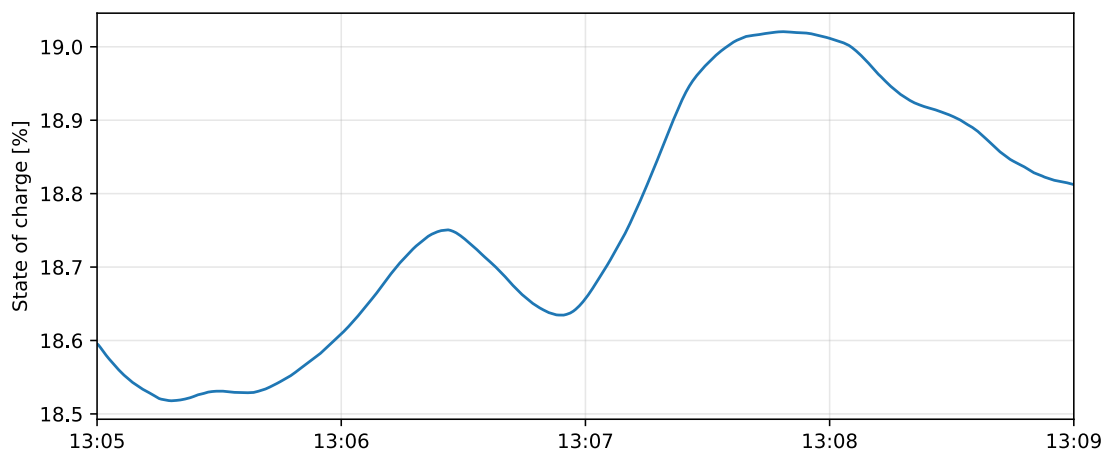


Figure 5.8: Battery state of charge over the FCR-N window.

5.4.2 FCR-D Up

Fig. 5.9 zooms in on the FCR-D up events. FCR-D up responds only to under-frequency below 49.9 Hz, so it is activated rarely and in short bursts, and the figure shows the intervals around these events rather than the whole day. In this recording the events fall in the morning, between about 06:30 and 08:45.



Figure 5.9: FCR-D Up activation on 2026-02-09 as a percentage of rated power derived from the frequency, shown over the morning interval that contains its events. Frequency data from [28].

The most prominent of these events, around 07:26, was replayed onto the dynamic power request at the rated power of 100 kW, using the same raised EV current limit as the ramp test in Section 5.3. The measured EVSE power is shown in Fig. 5.10. FCR-D up is an under-frequency reserve, so the request is negative and asks the unit to discharge, feeding power to the grid. The request follows the frequency-derived profile and reaches about -14.2 kW, and the measured power tracks it across the whole event down to the same -14.2 kW, so the unit delivers the full FCR-D up activation at this power level.

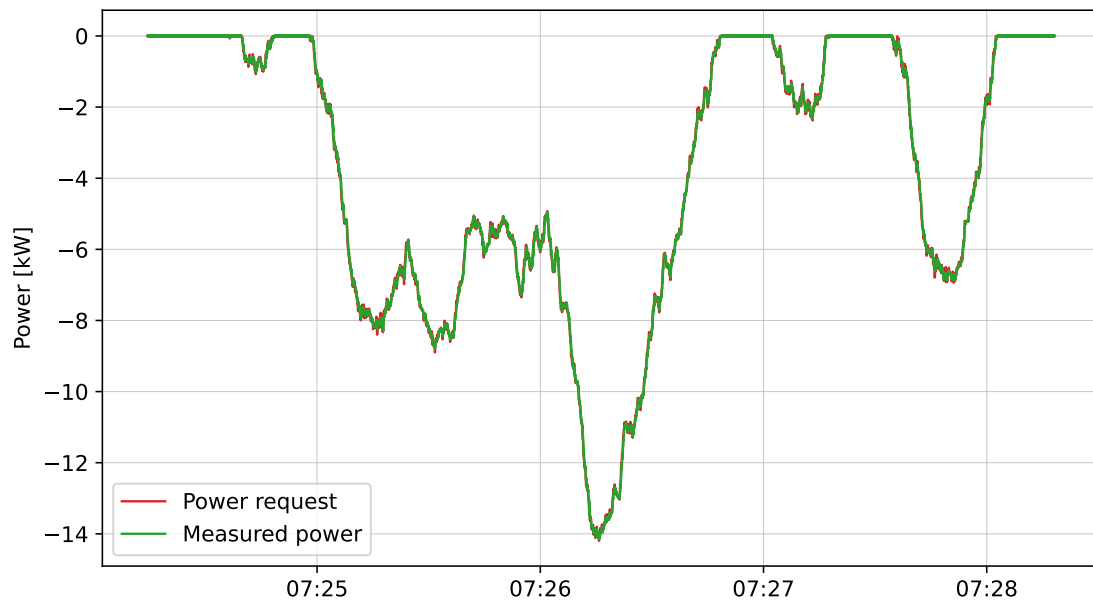


Figure 5.10: Measured FCR-D Up response. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

The tracking error is shown in Fig. 5.11. It stays within about ± 0.5 kW throughout, appearing only as small transients on the steepest parts of the activation, and is an order of magnitude smaller than for FCR-N because the request moves slowly and in a single direction without the reversals that drive the FCR-N error.

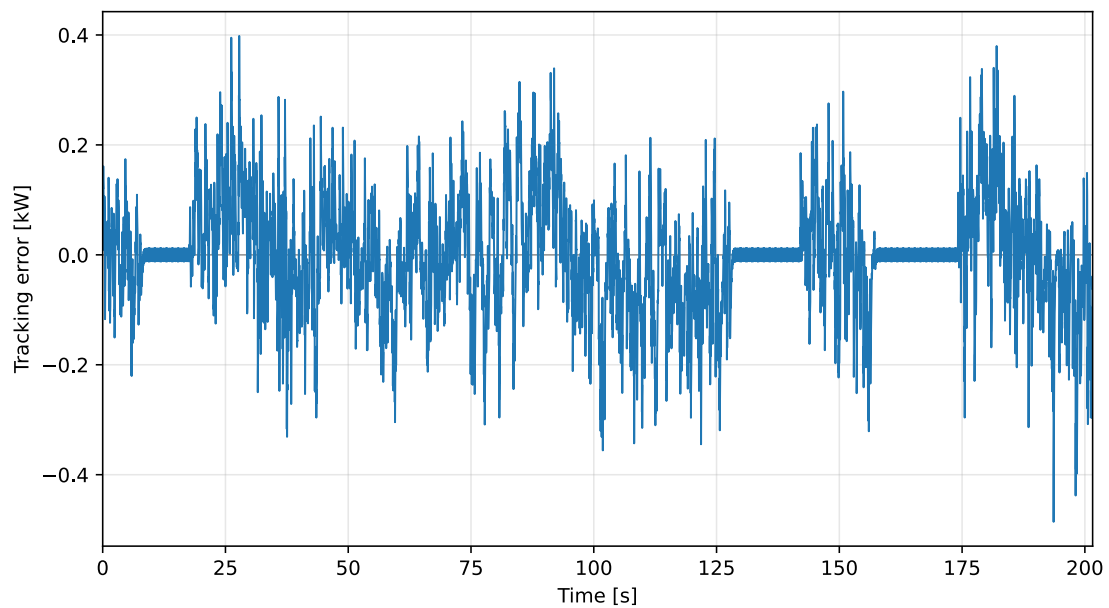


Figure 5.11: FCR-D Up tracking error over the event, taken as the measured EVSE power minus the dynamic power request.

Fig. 5.12 zooms to a single second on the activating edge, showing that the tracking holds through the rapid transients and not only on the slow envelope. The request advances in discrete steps about 60 ms apart, the update interval of the playback script, while the measured power follows as a smooth ramp lagging by a fraction of a second and staying within a fraction of a kilowatt of the request. Because each value is written at its elapsed time, this stepping does not distort the timing of the replayed profile.

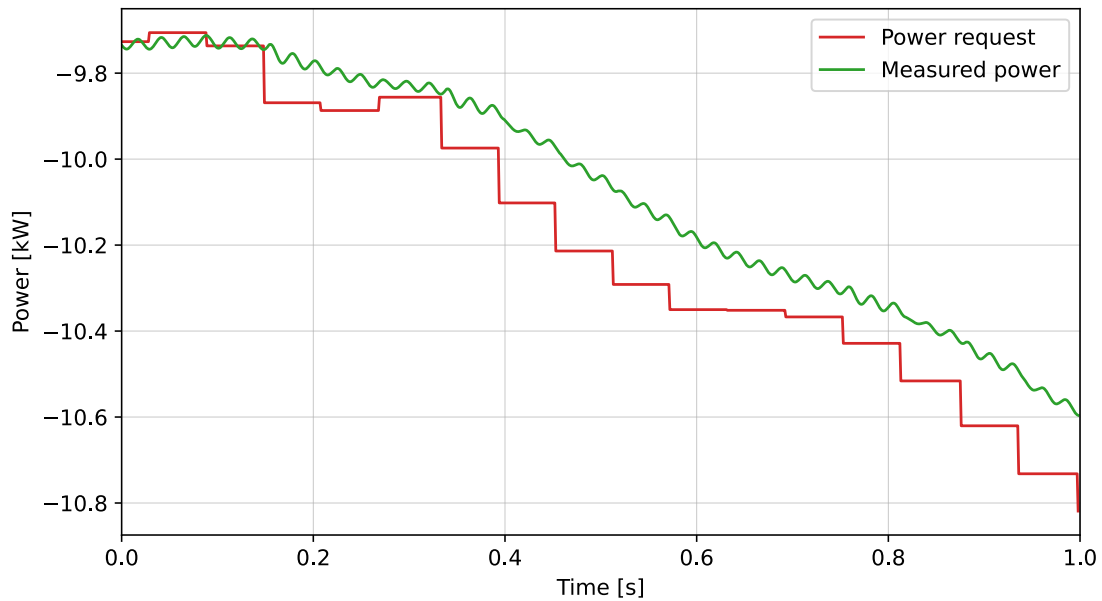


Figure 5.12: One-second window of the FCR-D Up activation on the activating edge. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

Fig. 5.13 shows the battery state of charge over the event. Since FCR-D up discharges, the state of charge falls in steps, one for each activation burst and flat in between, from about 22.0% to 21.7%, a net drop of roughly 0.3 percentage points.

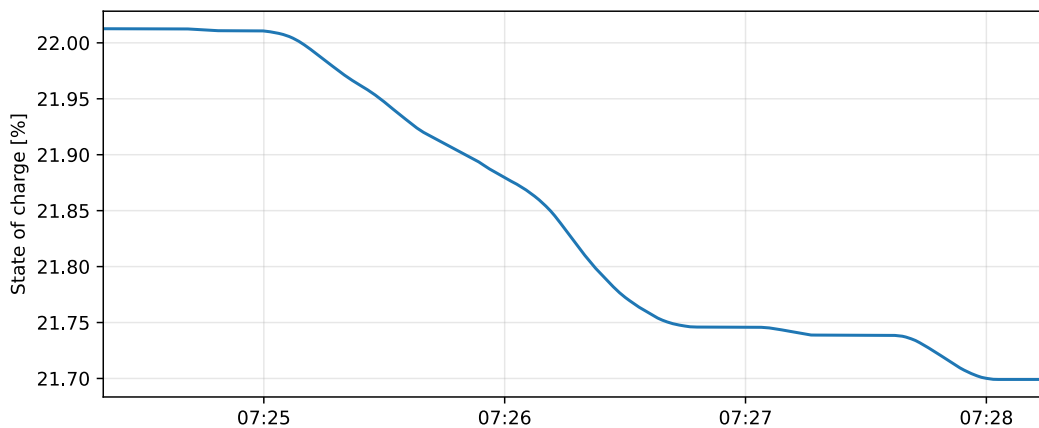


Figure 5.13: Battery state of charge over the FCR-D Up event.

5.4.3 FCR-D Down

Fig. 5.14 shows the corresponding FCR-D down events, which are triggered by over-frequency above 50.1 Hz and are likewise brief and infrequent. Here a single event occurs just after 08:00, and the figure is zoomed to the few minutes around it, roughly 07:59 to 08:04.

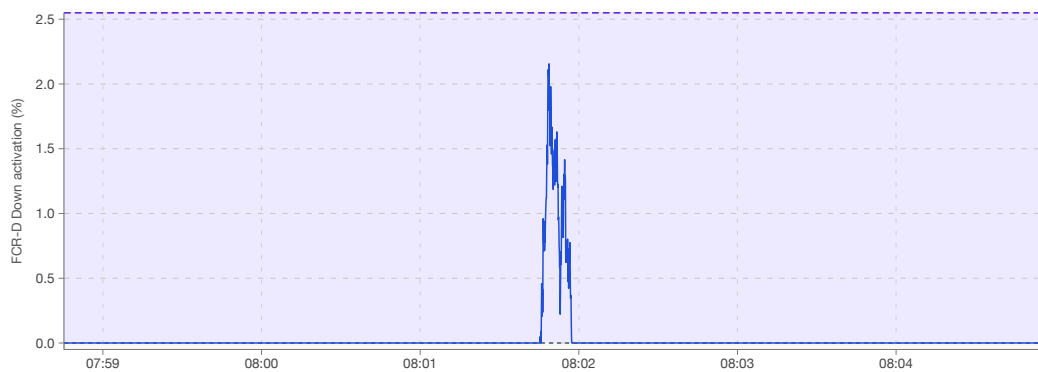


Figure 5.14: FCR-D Down activation on 2026-02-09 as a percentage of rated power derived from the frequency, shown over the few minutes around its single event. Frequency data from [28].

The single event, around 08:02, was replayed onto the dynamic power request at the rated power of 100 kW with the same raised EV limits, and the measured EVSE power recorded, shown in Fig. 5.15. FCR-D down is an over-frequency reserve, so the request is positive and asks the unit to charge, absorbing power from the grid. The request follows the frequency-derived profile and reaches about 2.1 kW, and the measured power follows it closely over the whole event, up to about 2.1 kW, with only a small deviation. The activation stays well within the unit’s capability, so the reserve is delivered without saturation, and the positive measured power confirms that the unit charges to provide down-regulation.

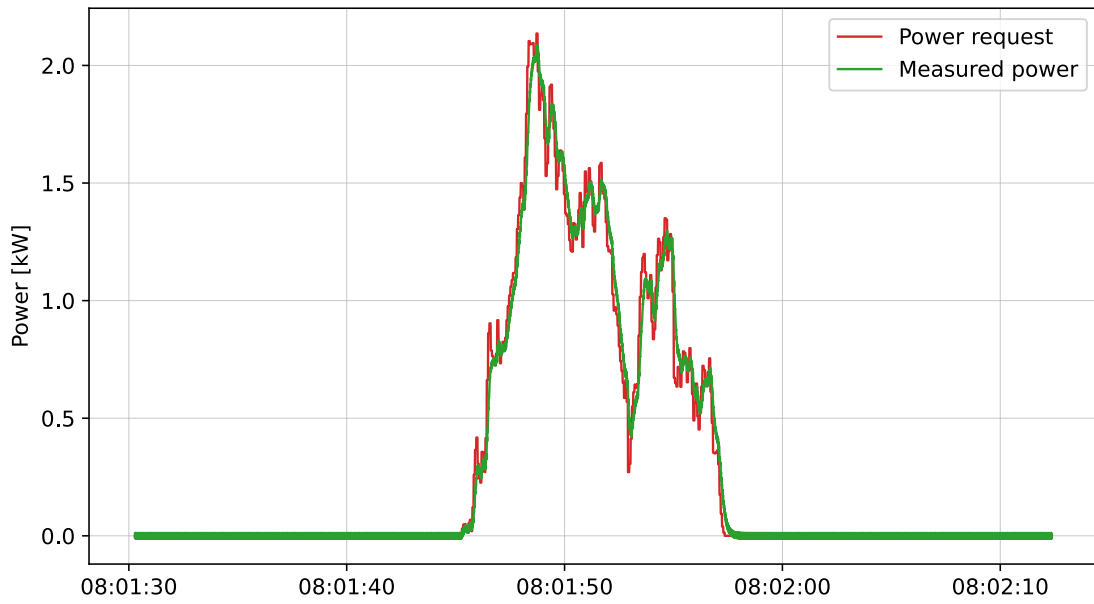


Figure 5.15: Measured FCR-D Down response. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

Fig. 5.16 shows the tracking error over this event, which likewise stays within about ± 0.5 kW despite the much smaller activation.

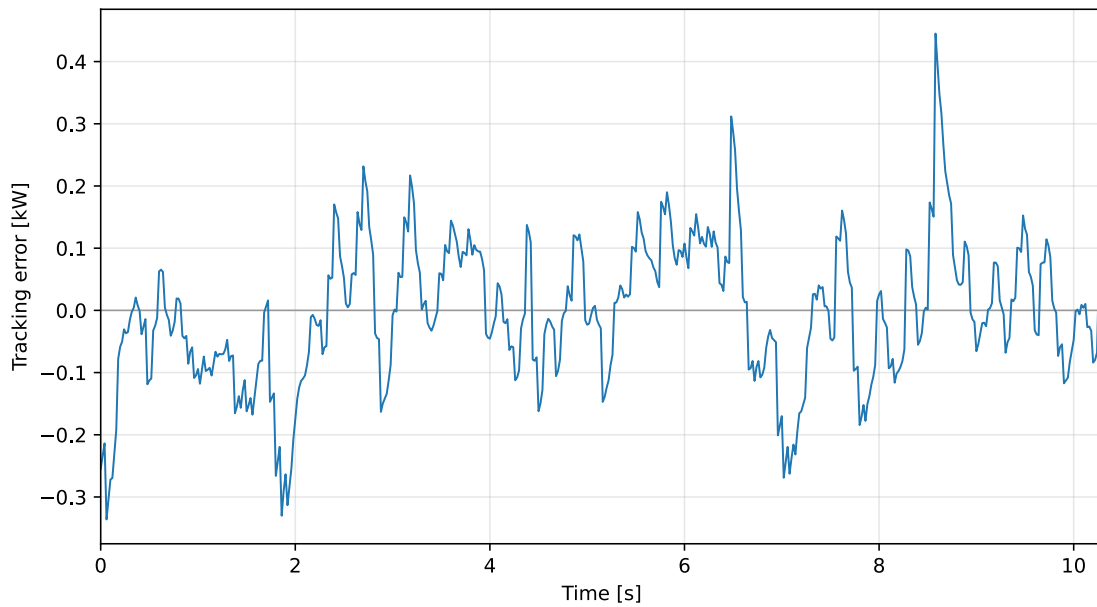


Figure 5.16: FCR-D Down tracking error over the event, taken as the measured EVSE power minus the dynamic power request.

Fig. 5.17 shows a single second of the activating edge of the event. As for the up reserve, the request advances in discrete steps and the measured power follows them

with a slight lag and a small ripple from the power stage, staying within a fraction of a kilowatt of the request, here in the charging direction.

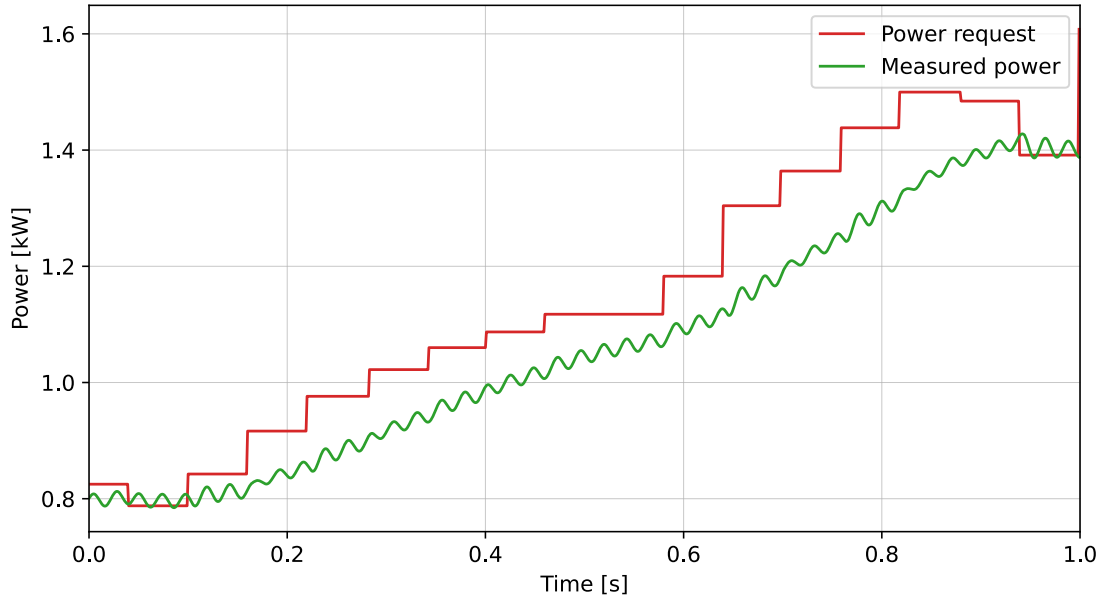


Figure 5.17: One-second window of the FCR-D Down activation on the activating edge. The red line is the dynamic power request by the ancillary service provider to the EVSE and green the power delivered by the EVSE.

Fig. 5.18 shows the battery state of charge over the down event. The single brief activation charges the battery, so the state of charge rises, but by only about 0.004 percentage points, since the activation is small and short.

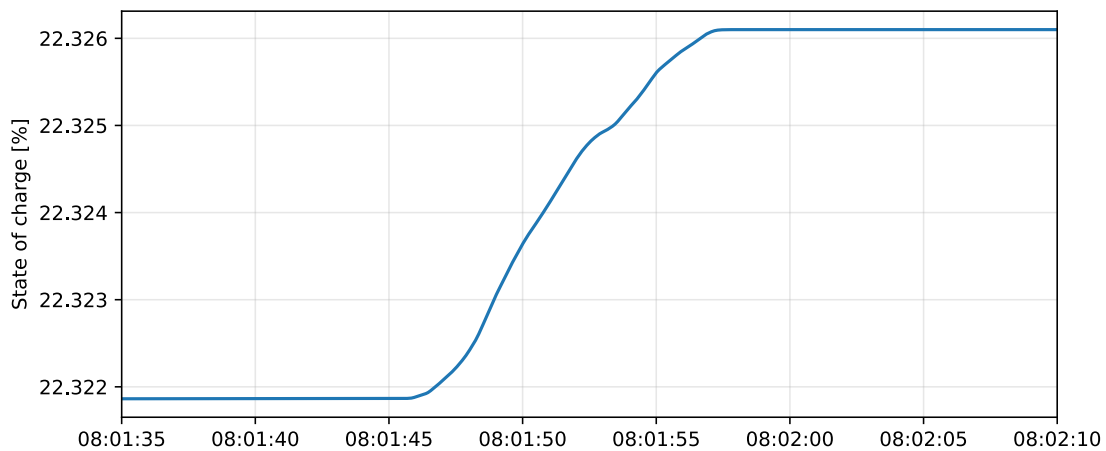


Figure 5.18: Battery state of charge over the FCR-D Down event.

5.5 Summary

Table 5.1 summarizes the tracking accuracy of the ancillary-service tests. For each reserve the measured EVSE power is compared against the dynamic power request over the active part of the event, using the mean absolute error (MAE), the root-mean-square error (RMSE), the largest absolute deviation, and the ratio of delivered to requested energy ($E_{\text{del}}/E_{\text{req}}$). For FCR-N the opening step of the window is excluded, since it is an artifact of cutting the window at a non-zero activation rather than a property of the reserve.

Table 5.1: Tracking accuracy of the measured EVSE power against the dynamic power request. The FCR-N figures exclude the opening step of the window.

Reserve	MAE [kW]	RMSE [kW]	Max. dev. [kW]	$E_{\text{del}}/E_{\text{req}}$
FCR-N	0.742	0.983	7.02	1.003
FCR-D Up	0.085	0.109	0.49	0.999
FCR-D Down	0.091	0.116	0.49	0.997

The two FCR-D reserves both track to within about 0.1 kW in mean absolute error and RMSE, and the energy delivered is within 0.3% of the energy requested. Relative to the peak of each event the up reserve tracks to about 0.8% RMSE, while the down reserve shows a larger relative error only because its activation is much smaller at about 2.1 kW and not because the absolute tracking is worse. FCR-N is the most demanding of the three, since it alternates between charging and discharging and reaches a much larger amplitude, and its deviations are correspondingly larger at 1.7% and 2.3% of the peak, with the worst case falling at a reversal. Its energy is still delivered to within 0.3%. Across all three reserves the unit follows the frequency-derived request closely and delivers almost exactly the requested energy.

These results can be viewed against the requirements listed in Table 2.5. The activation of FCR-N and FCR-D is specified as a linear function of frequency rather than as a fixed deadline, and the measurements show that the unit reproduces that characteristic accurately in both directions. The speed of the setup is not a limitation for these reserves either. The step response of Section 5.2 reaches the full 100 kW in about 2.3 s and settles within 2% after about 3.4 s, and the lag measured on the ramp is about 130 ms, which is short compared with the time over which the grid frequency traverses the containment bands. The minimum bid of 0.1 MW is also met, since the rated power used here is exactly 100 kW, although a single vehicle of this rating would sit precisely at the entry threshold and a real bid would be aggregated from a fleet.

Endurance is the requirement that these experiments do not demonstrate. As shown in Table 2.5, FCR-D has to be sustained for at least twenty minutes and FCR-N for an hour, while the tests reported here last minutes at most. The state of charge recorded during the FCR-N window shows the mechanism that would decide it. The reserve is symmetric in how it responds to frequency but it is not neutral in energy over a short window, so the state of charge drifts. Whether that drift stays within

the usable range of the battery over a full hour of FCR-N, or twenty minutes of FCR-D, is therefore one of the question left open by this work. On the evidence collected here the tested setup is capable of providing all three reserves in terms of tracking, speed and power, and the remaining uncertainty is one of energy rather than of control.

6

Conclusion

This thesis has presented the design and commissioning of a real-time DC V2G HIL testbed based on the dSPACE MicroLabBox II platform, the DS5366, and the SL1550A. The testbed establishes bidirectional DC charging over ISO 15118-20 communication between the emulated EVSE and the SL1550A acting as the vehicle, without a physical car, and it supports both manual and automated experiments. This meets the two objectives of commissioning functioning real-time high-level communication and of using it to run bidirectional charging experiments.

The scope of the emulation should be stated clearly. The SL1550A provides the vehicle side of the ISO 15118-20 communication and the physical layer for the PLC, while the electrical behavior of both the vehicle and the charger is simulated inside the MicroLabBox II. The SL1550A does not emulate a battery beyond a fixed present voltage that it reports during the precharge, and it sends no battery voltage during the charge loop. The battery is therefore simulated in the same model as the power supply, and the power controller reads its state directly inside the model rather than from the messages received over the PLC. In a full HIL split the battery would instead be emulated on the vehicle side and its state communicated to the EVSE power controller over the charge loop, which would place the PLC path inside the power-control loop. Because that path is absent here, the close tracking reported above does not include the latency and update rate that the real charge loop messaging would impose, and the tracking figures should be read with this in mind.

The commissioned setup was characterized across the communication sequence and the power response. The control pilot states and the full ISO 15118-20 session, from connection through cable check and precharge to the charge loop, were reproduced and observed end to end. The power stage was then exercised with step, ramp and frequency-derived profile tests. The step response is slew limited and settles within a few seconds, the ramp is tracked with a response lag of about 130 ms, and the three containment reserves FCR-N, FCR-D up and FCR-D down are all followed closely, with the delivered energy within about 0.3% of the requested energy in each case.

Measured against the Nordic reserve requirements, the setup satisfies the linear activation characteristic, the response speed and the rated power for all three reserves, so the open question is one of energy rather than of control. Endurance was not demonstrated, since the tests last minutes against requirements of twenty minutes for FCR-D and one hour for FCR-N, and the state of charge recorded during FCR-N

shows the drift that would decide it. The FCR-N measurements further show that the reserve inherently demands continuous charge and discharge reversals, which the power stage tracks with brief transient errors at each change of direction. These errors reflect the control and converter response rather than the battery itself, whose behavior is represented only by the simplified plant model in the simulation and not by a detailed electrical or ageing model. The testbed can therefore establish that FCR-N requires far more frequent reversals than the smoother FCR-D activations, but not the additional wear that such cycling would impose on a real battery. On that basis FCR-D still appears the more practical reserve for V2G, though a firm statement on battery stress would require a dedicated battery model on the vehicle side.

Overall the work confirms that HIL validation using standardized high-level communication is a practical approach for evaluating V2G systems under realistic real-time constraints. Future work should extend the tests to the full endurance windows required for prequalification against the Nordic FCR requirements defined by Svenska Kraftnät. A more faithful split would also emulate the battery on the EVCC side so that the EVSE power controller regulates against the battery state received over the PLC rather than read from the same model, which would place the communication inside the power-control loop and expose its latency and its effect on performance. As of writing, such vehicle-side battery simulation was not yet available on the SL1550A, although the corresponding parameters are already present in the SCE software, which suggests that the capability is in development.

Bibliography

- [1] J. Hellgren, J. Hagman, A. Mohammadi, K. Boman, and C. Jacobsson, “Analys av dubbelriktad laddning för en flotta av godsfraktande elfordon,” RISE Research Institutes of Sweden, Sweden, Project Plan Version 2, 2025.
- [2] J. Ru, M. Gillott, and R. Shipman, “Vehicle-to-grid (V2G) research: A decade of progress, achievements, and future directions,” *Energies*, vol. 18, no. 23, 2025. [Online]. Available: <https://www.mdpi.com/1996-1073/18/23/6148>
- [3] I. S. Bayram, “Quantifying the Effects of Communication Network Performance in Vehicle-to-Grid Frequency Regulation Services,” in *2020 International Conference on UK-China Emerging Technologies (UCET)*, Aug. 2020, pp. 1–4. [Online]. Available: <https://ieeexplore.ieee.org/document/9205457>
- [4] dSPACE GmbH. “MicroLabBox II: Compact Real-Time System for Development and Testing.” Accessed: Jan. 2026. [Online]. Available: <https://www.dspace.com/en/pub/home/products/hw/microlabbox/microlabbox-ii.cfm>
- [5] ——. “Smart Charging Solution: Testing Smart Charging Interfaces and Communication.” Accessed: Jan. 2026. [Online]. Available: https://www.dspace.com/en/inc/home/products/special_solutions/smart-charging-solution.cfm
- [6] Keysight. “SL1550A EV - EVSE Charging Communication Interface Tester.” Accessed: Mar. 2026. [Online]. Available: <https://www.keysight.com/ie/en/assets/3123-1157/data-sheets/SL1550A-EV-EVSE-Charging-Communication-Interface-Tester.pdf>
- [7] S. Yumiki, Y. Susuki, Y. Oshikubo, Y. Ota, R. Masegi, A. Kawashima, A. Ishigame, S. Inagaki, and T. Suzuki, “Autonomous vehicle-to-grid design for provision of frequency control ancillary service and distribution voltage regulation,” *Sustainable Energy, Grids and Networks*, vol. 30, p. 100664, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352467722000339>
- [8] J. J. Makrygiorgou and A. T. Alexandridis, “Unified modeling, control, and stability for a vehicle-to-grid and plug-in EV system,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 3, pp. 3284–3301, 2022.
- [9] E. García-Martínez, J. Muñoz-Cruzado-Alba, J. F. Sanz-Osorio, and J. M. Perié, “Design and experimental validation of power electric vehicle emulator

- for testing electric vehicle supply equipment (EVSE) with vehicle-to-grid (V2G) capability,” *Applied Sciences*, vol. 11, no. 23, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/23/11496>
- [10] M. Simpson, “SPIDERS bi-directional charging station interconnection testing,” National Renewable Energy Laboratory (NREL), Golden, CO, Technical Report NREL/TP-5400-60222, Oct. 2013. [Online]. Available: <https://docs.nrel.gov/docs/fy13osti/60222.pdf>
 - [11] A. Bahrami, *EV Charging Definitions, Modes, Levels, Communication Protocols and Applied Standards*, Jan. 2020.
 - [12] *Road Vehicles — Vehicle to Grid Communication Interface*, ISO 15 118-1:2019, 2019.
 - [13] *Electric Vehicle Conductive Charging System — Part 1: General Requirements*, IEC 61 851-1:2017, 2017.
 - [14] *Road Vehicles — Vehicle-to-Grid Communication Interface — Part 3: Physical and Data Link Layer Requirements*, ISO 15 118-3:2015, 2015.
 - [15] *Road Vehicles — Vehicle-to-Grid Communication Interface — Part 20: 2nd Generation Network and Application Protocol Requirements*, ISO 15 118-20:2022, 2022.
 - [16] Wikimedia Commons. “File:iec 62196 type 2 (m, dc, ccs combo 2).svg — wikimedia commons, the free media repository.” Accessed: Apr. 2026. [Online]. Available: [https://commons.wikimedia.org/w/index.php?title=File:IEC_62196_Type_2_\(M,_DC,_CCS_Combo_2\).svg&oldid=842793645](https://commons.wikimedia.org/w/index.php?title=File:IEC_62196_Type_2_(M,_DC,_CCS_Combo_2).svg&oldid=842793645)
 - [17] HomePlug Powerline Alliance, “HomePlug Green PHY Specification, Release Version 1.00,” HomePlug Powerline Alliance, Tech. Rep., Jun. 2010.
 - [18] Typhoon HIL. “ISO 15118-20 Protocol.” Accessed: Apr. 2026. [Online]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iso_15118_20_protocol.html
 - [19] Svenska kraftnät, “Technical requirements for frequency containment reserve provision in the nordic synchronous area,” Svenska kraftnät, Tech. Rep. Version 1.1, 2025. [Online]. Available: https://www.svk.se/495d28/siteassets/aktorsportalen/bidra-med-reserver/om-olika-reserver/fcr/fcr-technical-requirements_v1.1_28_march_2025.pdf
 - [20] —. “Snabb frekvensreserv (FFR).” Accessed: Apr. 2026. [Online]. Available: <https://www.svk.se/aktorsportalen/bidra-med-reserver/om-olika-reserver/ffr/>
 - [21] —. “Frekvensstabilitet.” Accessed: Mar. 2026. [Online]. Available: <https://www.svk.se/om-kraftsystemet/om-systemansvaret/kraftsystemstabilitet/frekvensstabilitet/>
 - [22] —. “Information on different ancillary services.” Accessed: Apr. 2026. [Online]. Available: <https://www.svk.se/en/stakeholders-portal/electricity-market/provision-of-ancillary-services/information-on-different-ancillary-services/>

- [23] The MathWorks, Inc. “Simulink Coder.” Accessed: Sep. 2026. [Online]. Available: <https://www.mathworks.com/products/simulink-coder.html>
- [24] dSPACE GmbH. “Milestones.” Accessed: Mar. 2026. [Online]. Available: <https://www.dspace.com/en/inc/home/company/companyhistory.cfm>
- [25] ——. “HIL Testing for Electronic Control Units (ECU).” Accessed: Mar. 2026. [Online]. Available: <https://www.dspace.com/en/inc/home/applicationfields/foo/hil-testing.cfm>
- [26] Research and Markets. “Hardware in the Loop Company Evaluation Report 2025 | dSPACE, Emerson Electric, and Vector Informatik Dominate with Robust ECU Testing, Automation, and Automotive Electronics Solutions.” Accessed: Mar. 2026. [Online]. Available: <https://www.globenewswire.com/news-release/2025/08/28/3140953/28124/en/Hardware-in-the-Loop-Company-Evaluation-Report-2025-dSPACE-Emerson-Electric-and-Vector-Informatik-Dominate-with-Robust-ECU-Testing-Automation-and-Automotive-Electronics-Solutions.html>
- [27] Keysight. “EV Charging and V2G Test.” Accessed: Aug. 2026. [Online]. Available: <https://www.keysight.com/us/en/industries/automotive/ev-charging-and-v2g.html>
- [28] Fingrid. “Frequency – historical data.” Dataset 339, licensed under CC BY 4.0. Accessed: Sep. 2026. [Online]. Available: <https://data.fingrid.fi/en/datasets/339>
- [29] H. Lee and M. Shin, “TestShark: A Passive Conformance Testing System for ISO 15118 Using Wireshark,” *Energies*, vol. 17, no. 23, Nov. 2024.

A

ISO 15118-20 Flowchart

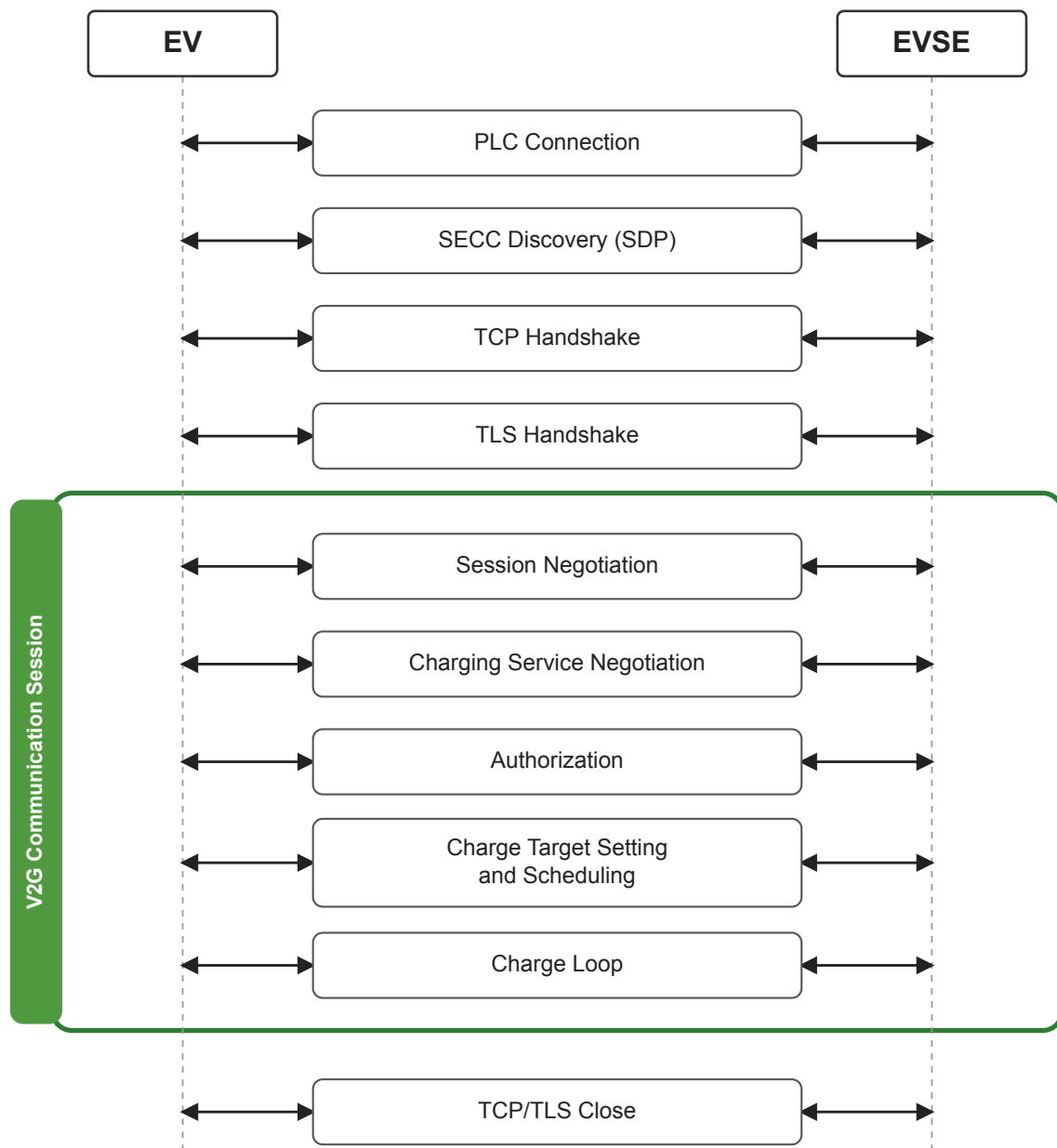


Figure A.1: ISO 15118-20 charge flowchart. Sourced from [29].

Fig. A.1 gives a high-level overview of a complete ISO 15118-20 charging session, from establishing the physical link to tearing it down. Before any V2G application messages are exchanged, the session is built up as a stack of successively higher-level connections, each depending on the one below it.

The individual phases of Fig. A.1 are as follows.

PLC Connection A logical PLC network is first established over the CP line using the SLAC procedure described in Appendix B.

SECC Discovery (SDP) The EV then locates the SECC on that network by issuing an SDP request, learning the SECC's IP address and the transport protocol it requires.

TCP and TLS Handshakes With the address known, a TCP connection is opened for reliable transport, followed by a mandatory TLS handshake that authenticates the SECC and encrypts all subsequent traffic.

V2G Communication Session The application-layer exchange then takes place. This is where session and service negotiation, authorization, charge scheduling, and the cyclic charge loop occur. The individual request/response messages that make up this phase are summarized in Appendix D.

TCP/TLS Close When charging is complete, the session is terminated and the TLS and TCP connections are closed, releasing the link.

B

SLAC

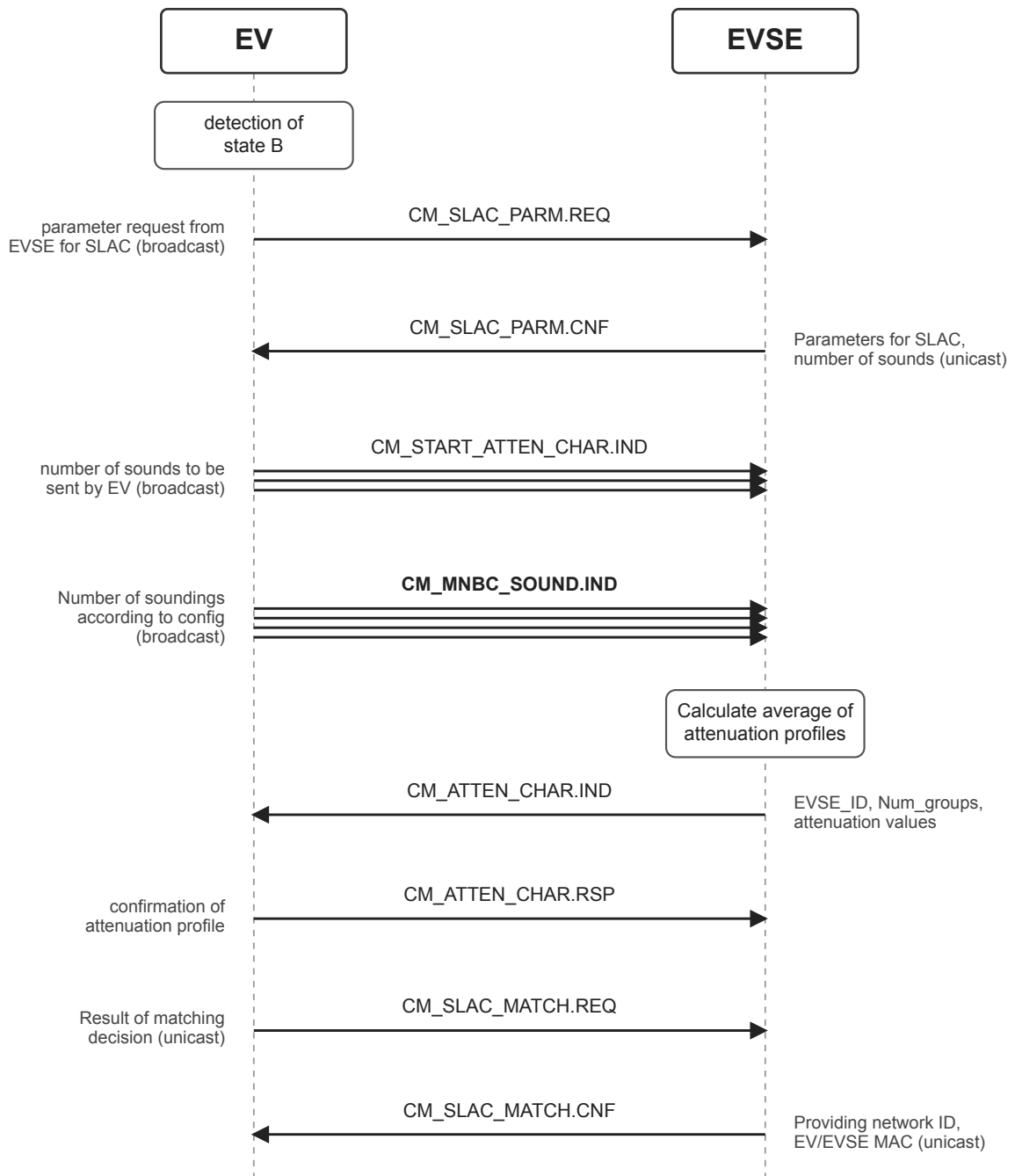


Figure B.1: SLAC message sequence between the EV and EVSE. Sourced from [11].

SLAC is the procedure defined in the HomePlug Green PHY standard that establishes a reliable point-to-point PLC link between the EV and the physically connected EVSE. Because the CP line is a shared medium and several chargers may be energized in close proximity, the EV must determine which EVSE it is actually plugged into rather than a neighboring one. SLAC resolves this by measuring the attenuation of test signals exchanged over the link. The strongest (least attenuated) connection identifies the correct physical pairing. The complete message exchange is shown in Fig. B.1.

The procedure begins once the EV detects state B on the CP line (see Table 2.3), indicating that a cable has been connected.

CM_SLAC_PARM.REQ The EV broadcasts this message to request the SLAC parameters from the EVSEs in range.

CM_SLAC_PARM.CNF Each EVSE replies with a unicast confirmation stating the number of sounding signals it expects.

CM_START_ATTEN_CHAR.IND The EV announces the upcoming soundings it is about to transmit.

CM_MNBC_SOUND.IND The EV transmits the configured number of sounding signals, whose received strength each EVSE measures.

CM_ATTEN_CHAR.IND Each EVSE averages the measured attenuation profiles and reports the result back to the EV.

CM_ATTEN_CHAR.RSP The EV acknowledges the received attenuation profile.

CM_SLAC_MATCH.REQ The EV compares the reported attenuations, selects the EVSE with the lowest one as its physical partner, and confirms this choice.

CM_SLAC_MATCH.CNF The chosen EVSE responds with the network identifier and the EV/EVSE MAC addresses needed to form the logical network.

Once matched, the two devices share a common logical PLC network on which the higher-level ISO 15118-20 communication of Fig. A.1 can proceed.

C

SECC Discovery Protocol

After the SLAC procedure has paired the EV with the physically connected EVSE and both endpoints have configured link-local Internet Protocol version 6 (IPv6) addresses, the EVCC still needs to learn how to reach the SECC at the network layer. The SDP provides this discovery step. It lets the EVCC obtain the SECC's IP address, port number, and the transport and security it expects, before any TCP or TLS connection is opened.

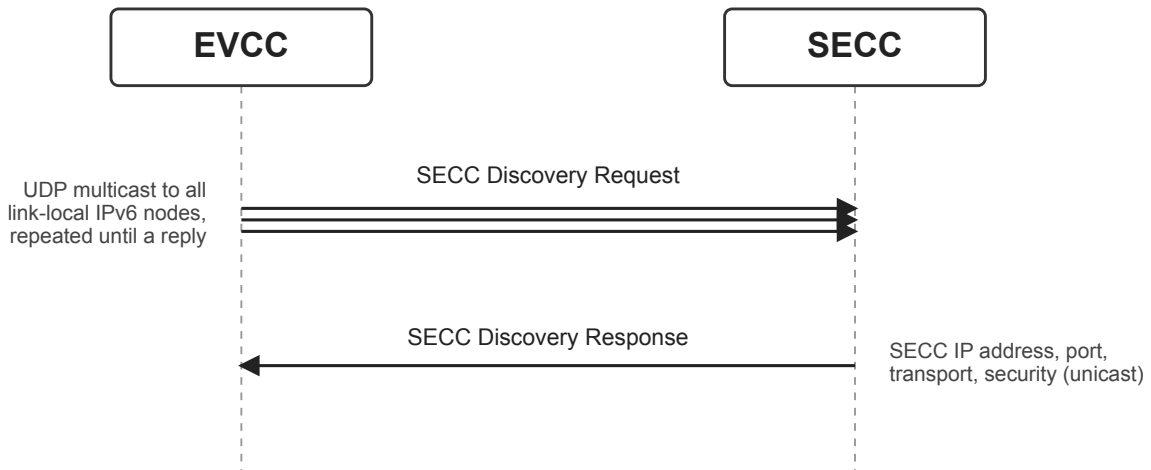


Figure C.1: SECC Discovery Protocol message exchange between the EVCC and SECC.

The exchange consists of a single request/response pair carried over User Datagram Protocol (UDP), as shown in Fig. C.1.

SECC Discovery Request The EVCC sends this as a UDP datagram to a well-known link-local multicast address and port, stating which transport protocol (TCP) and which security (TLS or none) it wishes to use. It is retransmitted at fixed intervals until a response is received.

SECC Discovery Response The SECC replies with a unicast message containing its IPv6 address, the port on which it is listening, and the transport and security it will actually use.

Once the EVCC has this information it opens the TCP connection and, where required, performs the TLS handshake, after which the ISO 15118-20 application session of Fig. A.1 can begin.

D

V2G messages

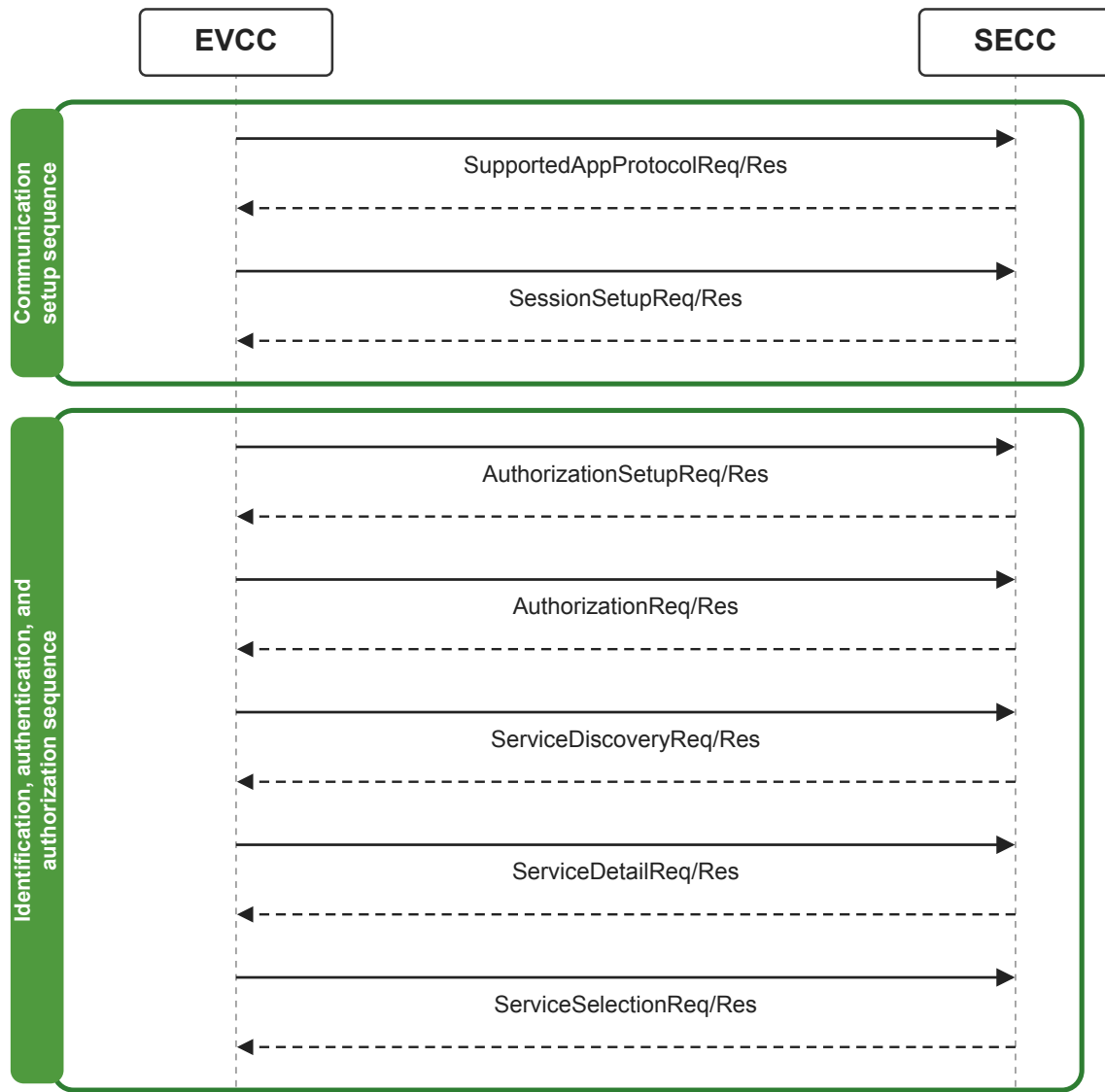


Figure D.1: ISO 15118-20 DC charging message sequence between EVCC and SECC: communication setup and authorization (1 of 2).

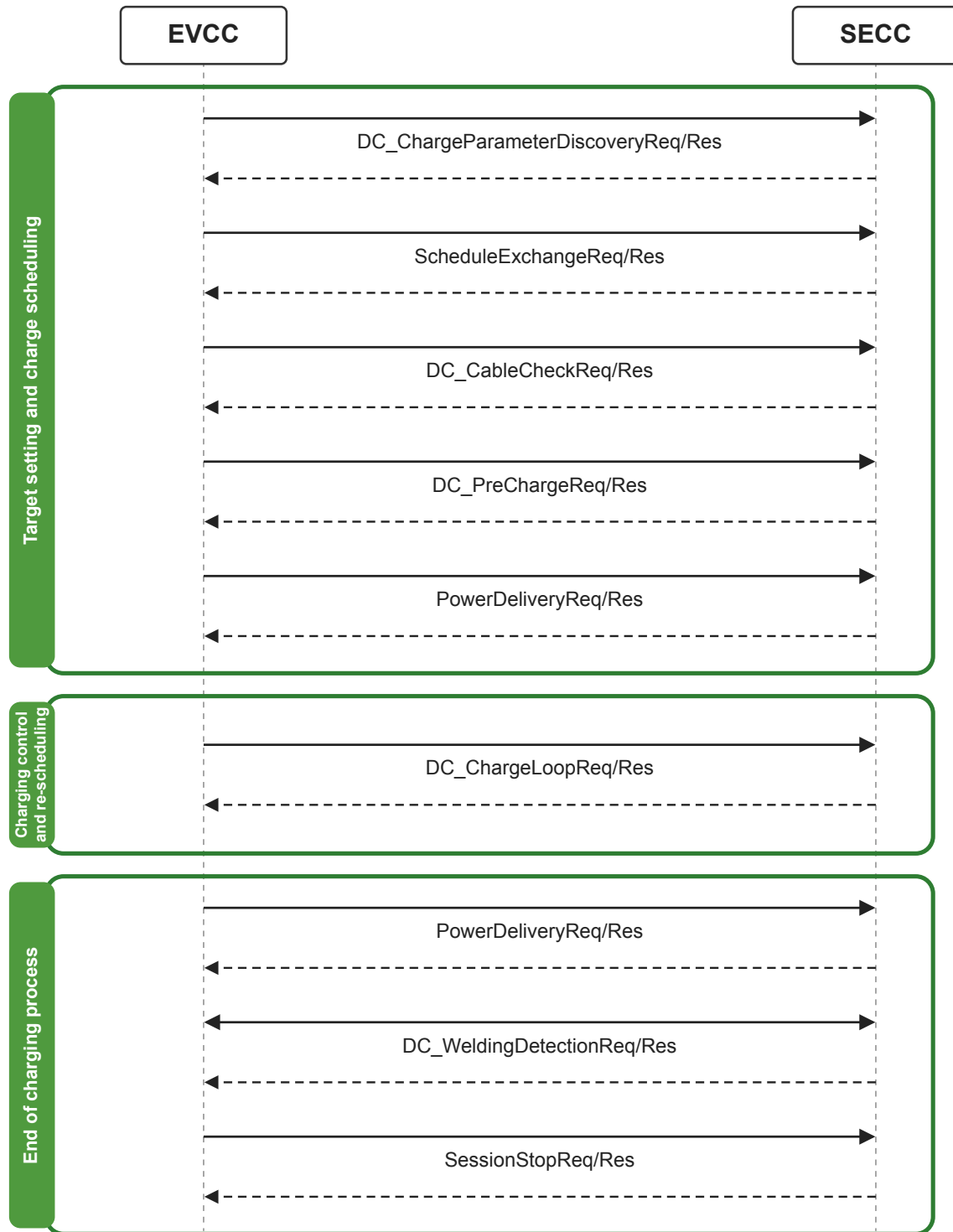


Figure D.2: ISO 15118-20 DC charging message sequence between EVCC and SECC: charging and session teardown (2 of 2).

D.1 Communication setup sequence

SupportedAppProtocolReq/Res The EVCC announces the application-layer protocols and version numbers it supports (e.g. ISO 15118-20, ISO 15118-2, DIN 70121), and the SECC selects the highest commonly supported protocol for the session.

SessionSetupReq/Res The EVCC requests a new communication session. The SECC replies with a unique session identifier and its EVSE ID, establishing the logical session that all subsequent messages reference.

D.2 Identification, authentication, and authorization sequence

AuthorizationSetupReq/Res The EVCC queries which authorization methods the SECC supports, either External Identification Means (EIM) or PnC, and for PnC it receives a generation challenge used to sign the contract credentials.

AuthorizationReq/Res The EVCC submits its selected authorization data (a signed contract certificate for PnC, or a reference to external means for EIM); the SECC returns whether authorization is accepted, still ongoing, or rejected.

ServiceDiscoveryReq/Res The EVCC requests the list of energy-transfer services (e.g. DC or bidirectional DC) and value-added services the SECC offers, each identified by a service ID.

ServiceDetailReq/Res For a chosen service ID, the EVCC requests the detailed parameter sets describing how that service can be configured, and the SECC returns the available options.

ServiceSelectionReq/Res The EVCC selects the energy-transfer service and any value-added services it intends to use, and the SECC confirms the selection.

D.3 Target setting and charge scheduling

DC_ChargeParameterDiscoveryReq/Res The EVCC and SECC exchange their respective DC operating limits, namely the minimum and maximum voltage, current, and power (including discharge limits for bidirectional transfer), which establishes the envelope within which charging must remain.

ScheduleExchangeReq/Res The EVCC requests a charging schedule. The SECC returns either a *dynamic* schedule, in which the SECC continuously dictates the setpoints, or a *scheduled* mode containing time, price, and power tuples that the EVCC follows.

DC_CableCheckReq/Res Before any high voltage is applied, the SECC performs an insulation (isolation) test of the cable and vehicle. The EVCC repeats the request until the SECC reports the test as finished.

DC_PreChargeReq/Res The SECC ramps its output voltage to match the EV battery voltage so the contactors can be closed without a large inrush current. The message repeats until the two voltages are sufficiently close.

PowerDeliveryReq/Res The EVCC commands the SECC to begin power delivery (charge progress = start), prompting the contactors to close and the charging loop to start.

D.4 Charging control and re-scheduling

DC_ChargeLoopReq/Res The cyclic message exchanged throughout active charging. The EVCC reports present voltage, current, and SoC together with its target and limit values, while the SECC reports its present output and, in dynamic mode, updated setpoints. This loop carries the actual (bidirectional) power transfer and any in-session re-scheduling.

D.5 End of charging process

PowerDeliveryReq/Res The EVCC commands the SECC to stop power delivery (charge progress = stop); the SECC ends its output and opens the contactors.

DC_WeldingDetectionReq/Res After the contactors open, the voltage is monitored to detect whether the DC contactors have welded shut, a safety check performed before the connector may be removed.

SessionStopReq/Res The EVCC requests termination (or pausing) of the session. The SECC confirms, after which the TCP and TLS connections are closed.

E

Automation Scripts

The automated ancillary-service tests rely on a small set of Python scripts that span the whole pipeline. An offline script converts a day of grid frequency measurements into per-reserve power-request profiles (Section E.1). A ControlDesk script replays a chosen profile onto the EVSE in real time (Section E.2), and a reduced form of it drives the ramp test (Section E.3). A final offline script extracts the recorded signals from the resulting measurement file for post-processing (Section E.4).

E.1 Frequency-to-power profile generation

Listing E.1 shows the script that maps recorded grid frequency to the activation level of each reserve. For every sample it evaluates the FCR-N, FCR-D up, FCR-D down, and FFR characteristics, combines them into a single net power request, and writes the per-reserve and combined percentages back to a CSV file.

```
1 import pandas as pd
2 import numpy as np
3
4 # 1. Load the frequency data
5 df = pd.read_csv('Taajuusdata2026-02-09.csv')
6 f = df['Value'].values
7
8 # 2. Map frequencies to unified and individual reserve activations (%)
9
10 # Symmetrical FCR-N: Continuous band between 49.9 Hz and 50.1 Hz
11 # Positive (+) means FCR-N Up, Negative (-) means FCR-N Down
12 fcr_n = np.select(
13     [
14         f <= 49.9,                # Fully saturated FCR-N Up
15         (f > 49.9) & (f < 50.0), # Linear FCR-N Up range
16         f == 50.0,                # Nominal dead-center
17         (f > 50.0) & (f < 50.1), # Linear FCR-N Down range
18         f >= 50.1                # Fully saturated FCR-N Down
19     ],
20     [
21         100.0,
22         (50.0 - f) / 0.1 * 100.0,
23         0.0,
24         -((f - 50.0) / 0.1 * 100.0),
25         -100.0
26     ],
```

```
27     default=0.0
28 )
29
30 # FCR-D down: active when f > 50.1 Hz, saturates at 100% above 50.5 Hz
31 fcr_d_down = np.select(
32     [f > 50.5, (f > 50.1) & (f <= 50.5)],
33     [100.0, (f - 50.1) / 0.4 * 100.0],
34     default=0.0
35 )
36
37 # FCR-D up: active when f < 49.9 Hz, saturates at 100% below 49.5 Hz
38 fcr_d_up = np.select(
39     [f < 49.5, (f >= 49.5) & (f < 49.9)],
40     [100.0, (49.9 - f) / 0.4 * 100.0],
41     default=0.0
42 )
43
44 # FFR: 100% step activation when f <= 49.7 Hz
45 ffr = np.select(
46     [f <= 49.7],
47     [100.0],
48     default=0.0
49 )
50
51 # 3. Combined net power request percentage
52 # Net Upward Direction (+ reserves) minus Net Downward Direction (- reserves)
53 combined_power = (fcr_n + fcr_d_up + ffr) - fcr_d_down
54
55 # 4. Save results back into the DataFrame
56 df['FCR_N_pct'] = fcr_n
57 df['FCR_D_down_pct'] = fcr_d_down
58 df['FCR_D_up_pct'] = fcr_d_up
59 df['FFR_pct'] = ffr
60 df['Combined_Power_Request_pct'] = combined_power
61
62 # 5. Export to CSV
63 df.to_csv('power_requests_2026-02-09.csv', index=False)
64 print("Data successfully processed with a single, symmetrical FCR-N column.")
```

Listing E.1: Offline script mapping recorded grid frequency to per-reserve power-request profiles.

E.2 ControlDesk playback

Listing E.2 shows the ControlDesk Python script used to replay a recorded power profile onto the EVSE's dynamic external power request. It resolves the model variable by its path, reads the per-sample power-request percentages from a CSV file, detects the active FCR-D up events (padding each with a short margin), and plays one event back in real time by writing the scaled setpoint to the variable at each timestamp.

```
1 variablesPlatform2 = Application.ActiveExperiment.Platforms[0].Platforms[0].
    ↪ ActiveVariableDescription.Variables
```

```

2 myVariable = variablesPlatform2.ItemByPath("urn:dspace:variable?ModelVariable=
    ↳ SCS_SECC_ISO_20/Tunable%20Parameters/
    ↳ SCSE_ISO20_PowerConfig__Dynamic_ext_power_request&ExperimentPlatform=
    ↳ Platform_EVSE&ApplicationPart=SCS_EVSE")
3
4 #start, stop, step, dt = 0.0, 10000, 0.5, 0.25 # 0 -> 10, +0.5 every 0.25 s
5 #v = start
6 #while v <= stop:
7 #     myVariable.ValueConverted = v
8 #     print(f"{time.strftime('%H:%M:%S')} -> {v:.2f}")
9 #     time.sleep(dt)
10 #     v += step
11 #myVariable.ValueConverted = stop # land exactly on target
12
13
14 import pandas as pd, numpy as np
15
16 CSV = r"C:\Users\maronen\Desktop\power_requests_2026-02-09.csv"
17 P_RATED_KW = 10.0
18 SCALE = 1.0 # 1000.0 if the variable is in W
19 SIGN = +1.0 # see caveat below -- flip to -1.0 if needed
20
21 df = pd.read_csv(CSV, parse_dates=["Time"])
22 t = (df["Time"] - df["Time"].iloc[0]).dt.total_seconds().to_numpy()
23 kw = df["FCR_D_up_pct"].to_numpy() / 100.0 * P_RATED_KW * SIGN # <-- FCR-D
    ↳ up only
24
25 MARGIN_S = 5.0
26 dt = np.median(np.diff(t)) # 0.1 s
27 pad = int(MARGIN_S / dt)
28
29 active = df["FCR_D_up_pct"].to_numpy() != 0
30 # indices where activity turns on/off
31 edges = np.diff(active.astype(int))
32 starts = np.where(edges == 1)[0] + 1
33 ends = np.where(edges == -1)[0] + 1
34 if active[0]: starts = np.r_[0, starts]
35 if active[-1]: ends = np.r_[ends, len(active)]
36
37 windows = [(max(0, s - pad), min(len(active), e + pad)) for s, e in zip(starts,
    ↳ ends)]
38 print(f"{len(windows)} FCR-D up events")
39 for i, (a, b) in enumerate(windows):
40     print(f" event {i}: t={t[a]:8.1f}s dur={t[b-1]-t[a]:5.1f}s peak={kw[a:b].
    ↳ max():.2f} kW")
41
42 #assert Application.ActiveExperiment.Platforms[0].Platforms[0].CalibrationState
    ↳ != 0, \
43     "Go online first"
44
45 def play(a, b, gap=0.0):
46     ts = t[a:b] - t[a]
47     t0 = time.monotonic()
48     for due, v in zip(ts, kw[a:b]):
49         slack = due - (time.monotonic() - t0)
50         if slack > 0: time.sleep(slack)

```

```
51         elif slack < -0.5: continue
52         myVariable.ValueConverted = v * SCALE
53         myVariable.ValueConverted = 0.0
54         time.sleep(gap)
55
56 play(*windows[0])                                # one event
57 #for w in windows: play(*w, gap=5.0)             # all events, 5 s idle between
```

Listing E.2: ControlDesk automation script for replaying an FCR-D up power profile onto the EVSE dynamic external power request.

E.3 Ramp test

Listing E.3 shows the ramp test, a reduced form of the playback script that ramps the dynamic power request linearly up to a target instead of replaying a recorded profile. It raises the converted value in fixed steps at a fixed interval and, at each step, logs the measurement signals together with the requested value to a CSV file.

```
1 import csv, time
2
3 evse = Application.ActiveExperiment.Platforms[0].Platforms[0].
4     ↳ ActiveVariableDescription.Variables # Platform_EVSE
5 power = Application.ActiveExperiment.Platforms[0].Platforms[1].
6     ↳ ActiveVariableDescription.Variables # Power_model
7
6 power_req = evse.ItemByPath("urn:dspace:variable?ModelVariable=SCS_SECC_ISO_20/
7     ↳ Tunable%20Parameters/SCSE_ISO20_PowerConfig_Dynamic_ext_power_request&
8     ↳ ExperimentPlatform=Platform_EVSE&ApplicationPart=SCS_EVSE")
9
8 # measurement variables to log
9 signals = {
10     "dc_power": evse.ItemByPath("urn:dspace:variable?ModelVariable=
11     ↳ SCS_SECC_Main/Labels/SCSE_Main_DC_power&ExperimentPlatform=Platform_EVSE&
12     ↳ ApplicationPart=SCS_EVSE"),
13     "dc_current": evse.ItemByPath("urn:dspace:variable?ModelVariable=
14     ↳ SCS_SECC_Main/Labels/SCSE_Main_DC_current&ExperimentPlatform=Platform_EVSE
15     ↳ &ApplicationPart=SCS_EVSE"),
16     "voltage_supply": evse.ItemByPath("urn:dspace:variable?ModelVariable=
17     ↳ SCS_SECC_Main/Labels/SCSE_Main_DC_voltage_at_supply&ExperimentPlatform=
18     ↳ Platform_EVSE&ApplicationPart=SCS_EVSE"),
19     "voltage_plug": evse.ItemByPath("urn:dspace:variable?ModelVariable=
20     ↳ SCS_SECC_Main/Labels/SCSE_Main_DC_voltage_at_plug&ExperimentPlatform=
21     ↳ Platform_EVSE&ApplicationPart=SCS_EVSE"),
22     "soc_pct": power.ItemByPath("urn:dspace:variable?ModelVariable=Model
23     ↳ %20Root/EV/battery_simulation/Battery/EV_SOC%5B%25%5D&ExperimentPlatform=
24     ↳ Power_model&ApplicationPart=SCS_power_simulation"),
25 }
26
17 start, stop, step, dt = 0.0, 10.0, 0.04, 0.02 # ramp 0 -> 10 kW at 2 kW/s,
28     ↳ logged at 50 Hz
29
19 with open(r"C:\Users\maronen\Desktop\ramp-test.csv", "w", newline="") as f:
20     writer = csv.writer(f)
```

```

21 writer.writerow(["time_s", "request_kw"] + list(signals))    # header row
22 t0 = time.monotonic()
23 v = start
24 while v <= stop:
25     power_req.ValueConverted = v
26     t = time.monotonic() - t0
27     writer.writerow([t, v] + [s.ValueConverted for s in signals.values()])
28     time.sleep(dt)
29     v += step
30 power_req.ValueConverted = stop                                # land on the
    ↪ target

```

Listing E.3: Ramp test that ramps the dynamic power request up to a target while logging the measured signals.

E.4 Measurement data extraction

During a test ControlDesk streams the measured signals to an ASAM MDF file. Listing E.4 shows the offline script that reads such a recording, selects the signals of interest, and resamples them onto a single uniform time base so that signals acquired on different rasters share one time axis. Each signal is recorded under its full model path, so the script matches on the last path component with any unit suffix removed, exports the result to a CSV file, and leaves the reduction to metrics and plots to the post-processing step.

```

1 from asammdf import MDF
2
3 # leaf name of the recorded channel -> the output column name
4 CHANNELS = {
5     "Ctrl_ext_power_request":      "request_kw",
6     "SCSE_Main_DC_power":         "dc_power",
7     "SCSE_Main_DC_current":       "dc_current",
8     "SCSE_Main_DC_voltage_at_supply": "voltage_supply",
9     "SCSE_Main_DC_voltage_at_plug": "voltage_plug",
10    "current_SOC":                 "soc_pct",
11 }
12
13 mdf = MDF("ramp_test.mf4")
14
15 def leaf(name):                    # last path component, without its unit
16     return name.split("/")[-1].split("[")[0]
17
18 lookup = {leaf(ch): ch for ch in mdf.channels_db}
19 wanted = {lookup[key]: col for key, col in CHANNELS.items() if key in lookup}
20
21 # resample every signal onto one uniform 50 Hz time base and export
22 df = mdf.to_dataframe(channels=list(wanted), raster=0.02, time_from_zero=True)
23 df.rename(columns=wanted, inplace=True)
24 df.index.name = "time_s"
25 df.to_csv("ramp_test.csv")

```

Listing E.4: Offline script that extracts the recorded signals from a ControlDesk MDF file and writes them to a CSV file on a common time base.