

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Electrical model of artificial intelligence infrastructure power supply unit

TESI DI LAUREA MAGISTRALE IN
ELECTRICAL ENGINEERING

Author: **Ioana-Andreea Parvu**

Student ID: 11075262

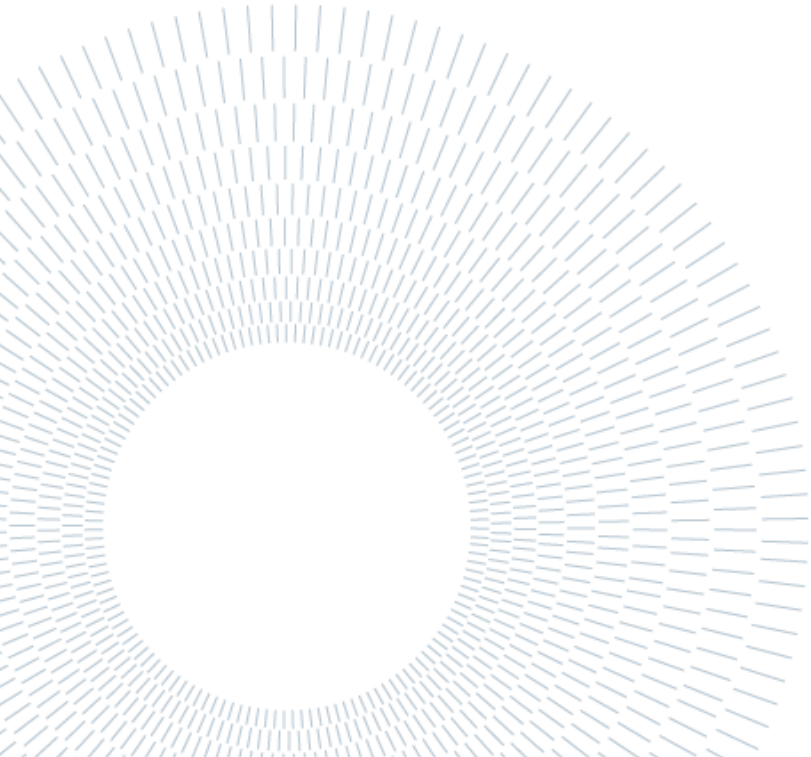
Advisor: Roberto Sebastiano Faranda

Academic Year: 2025-26

Abstract

Power supply units (PSUs) in AI data centre racks must satisfy demanding dynamic current requirements, as the GPU loads they power transition between idle and peak states at sub-millisecond timescales. The OCP Diablo 400 specification defines eight dynamic pulse load profiles for PSU qualification testing in 800 V DC AI data centre racks, but provides no simulation tool for generating them nor is there a clearly established circuit model for these devices. This thesis designs and compares two Simulink implementations: an active model, based on a signal-flow with an integrator FCN architecture commanding a controlled current source, and a passive model, that uses a switched-resistance RL topology to generate its current profile. After a structural comparison, both models were quantitatively validated against three reference grids. While both models perform the same for slow profiles, the active model is clearly superior for fast, slew-rate-limited transients and is the recommendation for PSU qualification testing. The passive model is still essential for studies where bus dynamics and inductive transients are relevant.

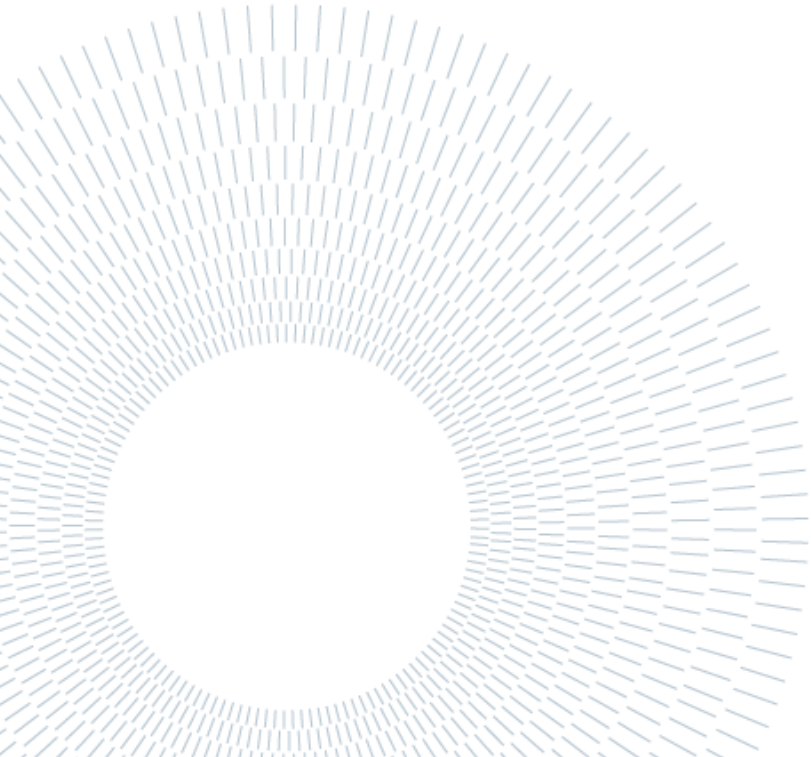
Key-words: 800V DC, load profile modelling, Simulink, OCP Diablo 400, PSU qualification, switched-resistance load.



Sommario

Le unità di alimentazione (power supply units) nei rack dei data center per l'IA devono soddisfare requisiti di variazioni dinamiche della corrente estremamente esigenti, poiché i carichi delle GPU che alimentano passano dallo stato di riposo (idle) a quello di picco in scale temporali inferiori al millisecondo. La specifica OCP Diablo 400 definisce otto profili di carico dinamico impulsivo per i test di qualificazione delle PSU nei rack AI a 800 V DC, ma non fornisce alcuno strumento di simulazione per generarli, né esiste un modello circuitale chiaramente stabilito per questi dispositivi. L'obiettivo di questa tesi è quindi stato quello di presentare e confrontare due differenti implementazioni in Simulink: un modello attivo, basato su un flusso di segnale con architettura a integratore (FCN) che comanda una sorgente di corrente controllata, e un modello passivo, che utilizza una topologia RL a resistenza commutata per generare il proprio profilo di corrente. Dopo un confronto strutturale, entrambi i modelli sono stati validati quantitativamente rispetto a casi di riferimento. Sebbene entrambi i modelli offrano prestazioni analoghe per i profili lenti, il modello attivo è nettamente superiore per i transitori veloci limitati dallo slew-rate ed è la soluzione raccomandata per i test di qualificazione delle PSU. Il modello passivo rimane tuttavia essenziale per gli studi in cui la dinamica del bus e i transitori induttivi risultano rilevanti.

Key-words: 800 V DC, modellazione di profili di carico, Simulink, OCP Diablo 400, qualifica PSU, carico a resistenza commutata.



Contents

Abstract **3**

Sommario **5**

Contents **7**

Introduction **11**

1 Theoretical Background **13**

1.1. AI Data Centres as an emerging electrical load 13

1.2. Distribution architecture: from AC to DC 13

1.3. Load modelling: from grid-level to rack-level DC 15

1.4. GPU load dynamics 16

1.5. PSU slew-rate requirements 18

1.6. Pulse load profiles 18

2 Current source model **19**

2.1. Model architecture 19

2.2. The MATLAB FCN 20

2.2.1. Initialization 20

2.2.2. Threshold computation and the Fraction/Ampere switch 21

2.2.3. Profile change detection - the initial_sign logic 21

2.2.4. The timer and state machine 22

2.2.5. The threshold clamping - preventing overshoot 22

2.2.6. Full threshold analysis for all 8 profiles 23

2.2.7. Profiles that do not reach their lower thresholds (profiles 1, 2, 4) 24

2.3. Scope outputs 25

2.4. Control parameter summary 26

3 The Controlled Current Source **27**

3.1. Connection to current source model 27

3.2. Circuit components 27

4 Passive Load Model **29**

4.1. Circuit description 29

4.2. FCN logic 30

4.3. Waveform analysis 32
4.4. Identified limitations 35
5 Comparison of Modelling Approaches 37
5.1. Fundamental difference in methodology of models 37
5.2. Circuit topology comparison 37
5.3. Waveform behaviour by profile 38
6 Reference Grid Models for Validation 41
6.1. Tuning grid parameters from measurement data 41
6.2. Grid topologies in Simulink 43
6.2.1. Grid 1: resistive source (V + R) 44
6.2.2. Grid 2: resistive source with diode (V + R + D) 45
6.2.3. Grid 3: controlled voltage source 46
7 Validation Methodology 49
7.1. Mapping load percentages to the nominal current 50
7.2. Nominal current choice for OCP compliance testing 50
7.3. Simulation setup 52
7.4. Profile classification and special cases 52
7.5. Data extraction and comparison metrics 54
7.5.1. Extraction method 54
7.5.2. R^2 computation 54
7.5.3. R^2_B load range metric 55
7.6. Results 55
7.6.1. Slow profiles R^2 56
7.6.2. Fast profiles R^2 56
7.7. Load-Range robustness R^2_B 59
7.8. Voltage behaviour 59
7.9. Summary and recommendation 60
8 Conclusion and Future Developments 63
8.1. Main conclusions 63
8.2. Future developments 64
Bibliography 65
A Appendix A 67
A.1. OCP Compliance Analysis Script 67
A.2. Parameters tuning for grid variations 75
A.3. Extract values per model, per grid, per load 76

List of Figures **79**
List of Tables **81**
List of symbols **83**

Introduction

The rapid growth of artificial intelligence infrastructure has fundamentally changed the electrical demands of data centres. GPU rack power has grown from around 10 kW per rack to over 120 kW today, with no signs of slowing in the future [1]. This has made traditional 48V distribution architectures physically impractical and driven the industry toward 800V DC distribution, where the same power can be delivered at a fraction of the conductor cross-section [1] [2]. The current these loads draw does not behave like any conventional load as it switches between idle and peak states at sub-millisecond timescales across distinct pulse profiles defined by the OCP Diablo 400 specification, and every PSU deployed in this architecture must be qualified against them.

While the industry specifications outline what those current profiles look like (their current levels, timing and slew-rates), they do not provide a simulation model for generating them in a controlled test environment. Most load modelling research centres on big-picture facility energy use [3] and AC-side grid interactions [4], while rack-level DC dynamics at the sub-millisecond, multi-profile transients remain mainly unaddressed.

The absence of a standard simulation tool for generating these profiles has a direct practical consequence PSU designers currently have no open reference implementation against which to verify dynamic compliance before hardware testing. This thesis addresses that gap with the following contributions. First, it designs and documents two Simulink [5] load models, an active controlled current source model and a passive switched-resistance model, as open reference implementations for PSU qualification. Second, it establishes a structured validation methodology using three reference supply grids of increasing physical detail, quantifying model accuracy against OCP specification targets using R^2 metrics. Third, it provides a clear, evidence-based recommendation on which modelling approach is more appropriate for qualification testing and under what conditions. The central research question is: which of the two topologies more faithfully reproduces the OCP pulse load profiles, and where does each approach fall short?

The thesis is structured as follows. Chapter 1 provides theoretical background on AI data centre loads, their impact on DC distribution networks, and the load modelling context. Chapters 2 and 3 document the signal-flow model and the controlled current

source circuit that form the active model. Chapter 4 presents the passive load model. Chapter 5 compares the two approaches qualitatively. Chapter 6 introduces three reference supply grids used to create a controlled validation environment. Chapter 7 presents the quantitative comparison against OCP targets using R^2 metrics and load-range robustness testing. Chapter 8 draws conclusions and identifies directions for future work.

1 Theoretical Background

1.1. AI Data Centres as an emerging electrical load

The shift to AI has pushed rack densities and power demand to levels that traditional data centre infrastructure was never built to handle. Traditional 48V distribution architectures become physically impractical as AI rack power demand increases: delivering 100 kW at 48V already requires conductors carrying over 2,000 A, a problem especially for next-generation liquid-cooled racks, which are projected to push current requirements even further [1]. This has driven the industry toward 800V DC distribution, where the same power requires a fraction of the conductor cross-section, reducing I^2R losses and enabling denser, more efficient rack designs [1] [2]. The OCP Diablo 400 specification, co-developed by Microsoft, Meta, and Google, formalises this architecture and defines the electrical requirements that every PSU in this type of rack must satisfy [2].

According to the International Energy Agency, global data centre electricity consumption reached approximately 415 TWh in 2024, with AI designed hardware projected to account for up to half of total data centre electricty demand by 2030 [6]. This growth rate is more than four times faster than that of all other electricity-consuming sectors combined [6].

In the past, IT equipment had a steady, predictable power draw. Grid operators and power supply engineers could treat these as stable loads that didn't change much over time. AI GPU workloads, by contrast, exhibit power demand that varies at multiple simultaneous timescales: sub-millisecond transients at the Electrical Design Point (EDP), second-scale checkpointing events during which the GPU briefly reduces power to save model state, and minute-scale ramp-up and ramp-down phases as entire training jobs start and stop [1]. This current load profile doesn't behave like a traditional server or a slow motor, but as a dynamic transient, representing something that must be modelled explicitly for a correctly designed PSU.

1.2. Distribution architecture: from AC to DC

Data center IT equipment, like servers, GPUs and network switches, has always consumed DC power internally, while microprocessors, memory, and storage all

operate on DC supply rails, typically at 12 V or below [1] [7]. What has evolved is the distribution architecture that delivers power to those loads from the utility grid.

Traditional data centres are supplied through AC distribution chains. The architecture path of a medium-voltage AC utility first steppes down to 480 V or 400 V AC at the building entrance, then it passes through a double-conversion UPS, following through PDU isolation transformers, which stepped down again the voltage, and arrives finally where the AC rectifies to DC inside each server PSU [7]. This chain involves at least four conversion stages before power reaches the load. In a typical data centre less than half the energy consumed at the utility meter is actually delivered to the processors, with the remainder lost in conversion, distribution, and cooling, this value worsens further at partial load, where converters operate well below their rated efficiency [7].

The advantages of AC distribution are very clear: a large ecosystem of certified components, mature international protection standards, and natural fault-current zero crossings every half-cycle that allow conventional circuit breakers to interrupt faults without specialised devices [8] [9]. For legacy data centres built around CPU workloads, where rack power was measured in kilowatts and load behaviour was relatively stable, these properties made AC distribution the practical and economical choice [9].

However, as AI rack power density has risen from kilowatt-scale to megawatt-scale, the physical limits of AC distribution have become apparent [1] [9] [10]. In order to maintain AC distribution to the rack at these power levels requires extreme conductor cross-sections: going from 415 VAC (4-wire, three-phase) to 800 VDC (3-wire) allows 157% more power to be transmitted through the same conductor cross-section [1], since power scales with voltage at fixed current and the DC system requires one fewer conductor. To illustrate the opposite extreme, distributing 1 MW at the traditional in-rack server voltage of 54 VDC would require currents exceeding 18,000 A (since $I = P/V$) and industry estimates put the resulting copper busbar requirement at up to 200 kg per rack [10] [11]. This shows that the distribution voltage must be high enough to keep currents manageable, and 800 VDC has emerged as the practical solution at megawatt rack densities [1] [9].

The multiple conversion stages in AC chains introduce cumulative losses: eliminating the UPS inverter stage and the PDU step-down transformer through facility-level DC distribution has been shown to meaningfully improve overall efficiency [7]. The transition to 800 V DC moves the single AC/DC conversion step upstream to the facility level and delivers DC directly to the rack, reducing both the number of conversion stages and conductor requirements [1] [9]. Industry projections suggest that 800 VDC distribution could reduce copper requirements by 45% and improve efficiency by approximately 5% relative to AC, with an estimated 30% reduction in total cost of ownership at gigawatt scale [10] (these figures reflect a single industry analysis and should be understood as indicative rather than independently verified).

However, DC distribution shifts the stability and protection problem from the AC side to the DC bus. On a DC bus there is no zero crossing: without natural interruption, fault currents can be sustained indefinitely and are accompanied by high-frequency oscillations which are driven by the discharge of bus capacitors through the distributed inductance of the cabling, with oscillation periods as short as a single millisecond and amplitudes reaching tens of thousands of amperes [8]. The discharge of PSU input filters can also trigger a reverse-polarity pulse visible by all protection devices on the bus and creates the risk of unnecessary tripping of breakers not on the fault current path [8]. Protection therefore requires fast, selective devices capable of distinguishing between faults and normal high-current transients, a much harder problem than in AC systems [8] [9] [12]. Furthermore, since there is no AC inertia to buffer sudden load changes, bus voltage regulation during fast transient must be handled through capacitive buffering and converter control [1]. Managing these transients at the rack level requires energy storage sized to cover both sub-millisecond capacitive spikes and second-scale idle-to-peak transitions generated by GPU workloads [1] [13].

The OCP Diablo 400 specification addresses this by imposing strict transient limits: ±3% voltage deviation and 3 ms settling time following step load changes up to 140% of rated current [2]. These limits can only be verified through simulation against realistic dynamic load profiles, which is precisely the function of the simulation models developed in this work. Currently, the absence of a standard DC load model for OCP pulse profiles means designers currently have no reference tool for simulation-based PSU qualification. This thesis directly addresses this issue, by proposing two types of Simulink implementations.

1.3. Load modelling: from grid-level to rack-level DC

Load modelling in power systems originated in the AC grid context, where utilities and system operators needed aggregate representations of demand to run stability and load-flow studies. The ZIP model, developed for this purpose, represents an aggregate load as a mix of constant impedance (Z), constant current (I), and constant power (P) components, a framework that remains the dominant international industry practice for utility stability studies [14] [15]. The existing data centre simulation models operate in a similar way at this aggregate level, producing hourly or component-level consumption profiles for facility planning [3], rather than the sub-millisecond current waveforms at a single rack that PSU qualification requires. Applying it directly to DC rack-level dynamics, where current switches between states at sub-millisecond timescales driven by GPU workload transitions, requires significant adaptation [4].

At the rack level, the two models developed in this thesis correspond physically to the constant-current and constant-impedance archetypes of the classical ZIP framework

[4], implemented at the sub-millisecond timescales relevant to GPU load dynamics rather than slow averages. Recent research categorizes data centres as Large Electronic Loads (LELs), a load class that demands a hybrid modelling that combines circuit physiscs with calibration to real operational data [16]. While data-driven methods map parameters to waveforms, they can produce unrealistic dynamic behaviour and overfitting when the training data is limited [16]. This thesis follows a physics-based model instead, constructing an equivalent circuit whose component values replicate the measured electrical behaviour of the load. For PSU design and qualification, the physics-based approach is preferred because it provides clear, interpretable results that can be linked directly to circuit parameters and OCP specification requirements.

The work of Xu et al. on solid-state transformer-driven 800VDC data centre simulation demonstrates that physics-based high-fidelity RTDS models evaluated against real operating profiles are both feasible and necessary for design validation of next-generation distribution architectures [13]. This thesis follows that methodology, but in Simulink, at the load side: rather than modelling the entire distribution chain, the focus is on accurately replicating the current profiles that the load presents to the bus, using two complementary topologies whose properties are mapped in detail in the following chapters.

1.4. GPU load dynamics

The load profiles seen in modern AI data centres differ from conventional IT loads. The primary challenge is the wide range of the current drawn. In this case, it is observed how GPUs operate across multiple timescales simultaneously, moving from high-speed transients to long-duration cycles. To design a reliable PSU, simulation models that can replicate these exact behaviours are needed, since standard load models often fail to capture these dynamics.

From the document “800 VDC Architecture for AI Infrastructure” its Figure 4-top (the screenshot noted as Figure 1-1 in this report) presents the actual power envelope of a GPU rack during a large language model (LLM) run [1]. It can be seen that the workload is more complex than the ideal approximation of a square wave. In real world, the workloads exhibit events at multiple timescales simultaneously:

- The first essential part is the Electrical Design Point (EDP). It presumes the very small (milliseconds) slew rate transitions of how GPU goes between idle and peak states.
- Other elements are memory operations, GPU peak, idle, and power-limit-throttled states happen in milliseconds cycles, while the checkpointing events (the GPU briefly reduces power to save model state) have a duration of 1-5 seconds, and lastly, the ramp-up and ramp-down phases happen in minutes.

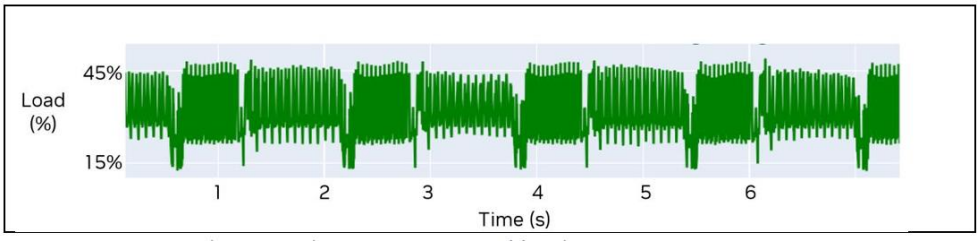


Figure 1-1 - Actual power envelope of a GPU rack during a Large Language Model (LLM) run [1]

However, these events happening at different timescales (ms or even min) create issues which are conflicting with each other: the power grid demands a stable, predictable load, yet the GPU demands a highly dynamic power source. The solution consists in energy storage at the rack level, which makes a more accurate characterisation of the load dynamics during the design stage.

In more detail, in the same document, Figure 4-bottom (screenshot seen in Figure 1-2) shows the power envelope over a full cycle, which basically contains a breakdown of the above mentioned LLM workload run from Workload Start to De-loading [1]. Here are the important sections seen clearly: Workload Ramp, EDPP (Electrical Design Point Peak) spikes, Idle periods, and Workload Running, which are plotted against three reference levels: P_Peak, P_Avg, and P_Idle. The shaded regions indicate which portion of the power swing must be covered by bulk capacitance (dark green) versus battery or power smoothing storage (light green), with the desired AC envelope shown as a dashed red line. Together, these two figures establish that energy storage must be designed to cover the transients from under millisecond spikes down to seconds-long idle-to-peak transitions.

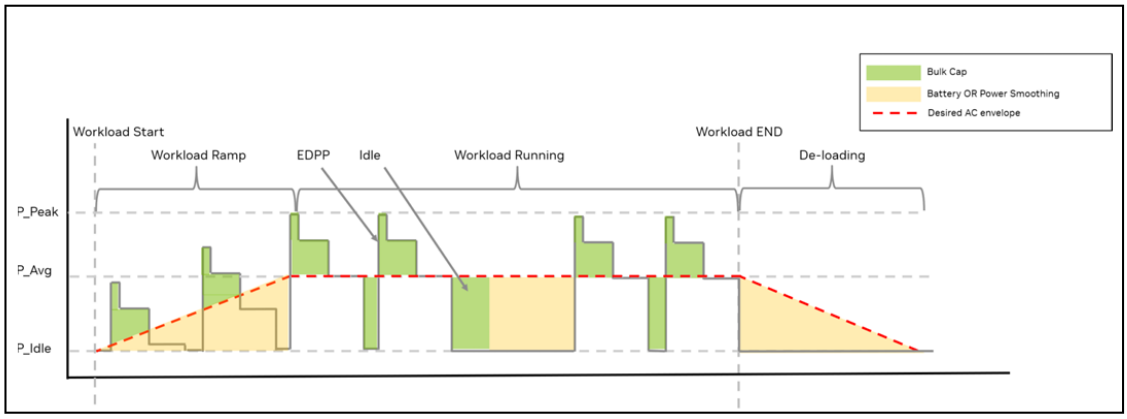


Figure 1-2 - Power envelope over a full cycle [1]

1.5. PSU slew-rate requirements

In the "OCP Diablo 400" documentation [2], Table 6 defines the dynamic load requirements that every PSU must satisfy. The key constraints relevant to this model are seen below (Table 1-1):

Table 1-1 - Table 6 from "OCP Diablo 400" documentation [2]

Step load increase or decrease	Frequency	Slew rate per PSU	Max undershoot and overshoot
50% step load (10% min. load)	Up to 10 kHz	1A/us	±3%
90% step load (10% min. load)	Up to 10 kHz	1A/us	
140% step load (10% min. load)	Up to 10 kHz	1A/us	

The slew rate requirement of 1 A/µs is found in the simulation model under the label sr, being a parameter of the MATLAB FCN block. Its purpose is to physically limit how fast the current source can transition between its low and high states. This constraint will later (Chapter 2.2.6.) show how it can determine whether the current waveform achieves its target thresholds.

1.6. Pulse load profiles

Table 8 (reproduced in Table 1-2) of the "OCP Diablo 400" document defines 8 specific pulse load profiles that PSUs must be tested against [2]. Each profile specifies a low and a high current level, phase durations, and the required PSU slew rate. These 8 profiles are the direct source of the load_params lookup table in the simulation FCN. Every array element has a one-to-one mapping with one row of Table 8. Therefore, with the help of all this information, the simulation is supposed to generate current waveforms that replicate what each of these 8 profiles demands from the bus, first using a controlled current source to isolate the current behaviour from voltage effects and then using only passive components to finally compare these two topologies.

Table 1-2 - Table 8 from "OCP Diablo 400" document [2]

No.	Low load	High load	T_low	T_high	Average load	Frequency	PSU slew rate
1	70%	130%	0.05 ms	0.05 ms	100%	10 kHz	1 A/us
2	20%	140%	0.1 ms	0.1 ms	80%	5 kHz	1 A/us
3	40%	175%	2.5 ms	2 ms	100%	222 Hz	1 A/us
4	0%	80%	1 ms	1 ms	40%	500 Hz	0.1 A/us
5	20%	140%	50 ms	50 ms	80%	10 Hz	1 A/us
6	60%	140%	50 ms	50 ms	100%	10 Hz	1 A/us
7	30%	140%	28.6 ms	50 ms	100%	12.7 Hz	1 A/us
8	30%	175%	0.43 ms	0.4 ms	100%	1.2 kHz	1 A/us

2 Current source model

The signal-flow model documented in this chapter, which includes the FCN, integrator, and profile subsystem, was provided as a reference implementation for this project. This chapter documents its architecture in full, as understanding its logic is necessary both for the circuit implementation in Chapter 3 and for the final comparison of the active vs passive models in Chapter 7.

2.1. Model architecture

The active model is implemented across two layers. The first is a Simulink signal-flow layer, which handles profile selection, parameter passing, and current integration. The second is a MATLAB algorithm layer, implemented as an FCN block, which contains all the profile logic, state machine, and threshold clamping.

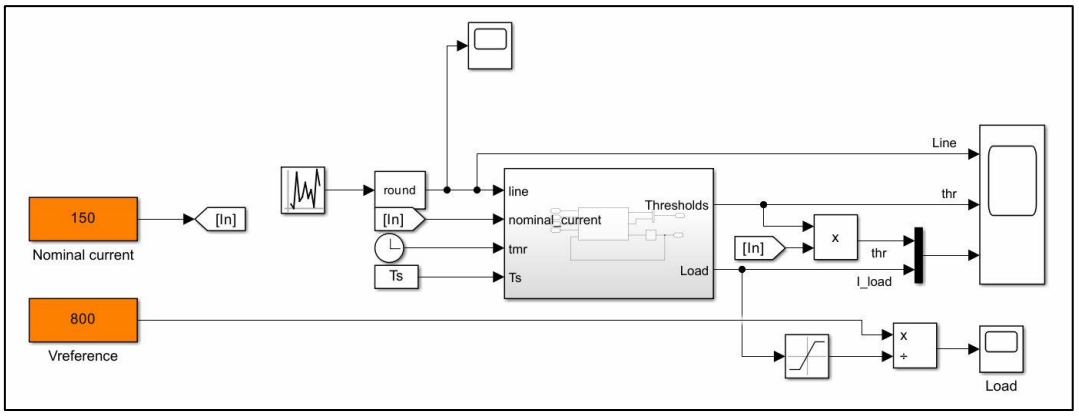


Figure 2-1 - Top-level Simulink model

The Simulink layer (Figure 2-1) contains the following key blocks:

- Band-limited noise to round: This is the profile selector. It generates a varying random signal, which approximately converts to an integer 1-8. This selects which of the 8 Table 8 profiles is currently active and it cycles through all profiles during simulation.
- Nominal current (150 A): a constant block. Thresholds are calculated inside FCN with the help of this value, therefore making the model reusable for any nominal current.
- Vreference (800 V): the bus voltage reference.

- Ts constant: the simulation timestep which also plays a role in the slew rate clamping calculation inside the FCN.
- The FCN block: takes profile, nominal current, tmr, Ts and outputs_thresholds (a 2-element vector of [high, low] as fractions) and load (normalized current value, 0 to ~1.75).

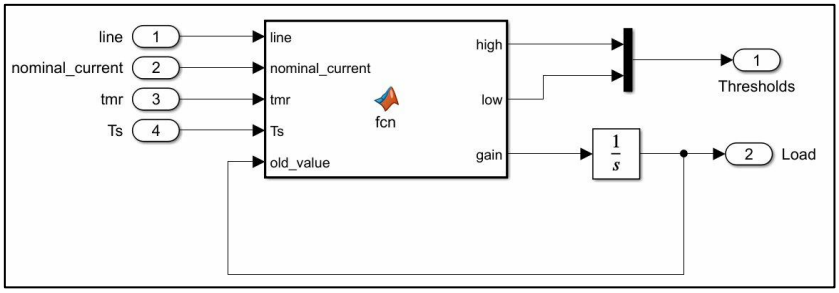


Figure 2-2 - Internal signal flow showing FCN, integrator, and feedback path

The Figure 2-2 reveals how slew rate is physically implemented in simulation. The FCN block does not output a current value directly, it outputs a rate of change (gain, in A/s). The integrator 1/s converts this rate into an actual current value each timestep:

$$\text{Load}(t) = \text{Load}(t-1) + \text{gain}(t-1) \times \text{Ts}$$

A gain of 1×10^6 A/s with Ts = 1×10^{-5} s adds 10 A per timestep, which is exactly a 1 A/µs ramp rate. The feedback wire old_value returns the integrator output back into the FCN so the function can monitor the current value and apply threshold clamping before overshoot occurs.

2.2. The MATLAB FCN

2.2.1. Initialization

The initialization part is an encoding of Table 8 profiles (the one-to-one mapping) from the OCP Diablo 400 specification. Each index 1-8 corresponds to one row:

- low: minimum current as fraction of nominal (e.g., 0.70 = 70%)
- high: maximum current as fraction of nominal (e.g., 1.30 = 130%)
- t_low: duration of the falling phase in seconds
- t_high: duration of the rising phase in seconds
- sr: the PSU slew rate in A/s (1×10^6 = 1 A/µs, while line 4 uses 0.1×10^6 = 0.1 A/µs)

An important observation is that the FCN works with normalized fractions. Actual Ampere scaling is applied outside the subsystem via the × nominal_current multiplier, making the FCN reusable for any system nominal current.

2.2.2. Threshold computation and the Fraction/Ampere switch

At the start of each call, high and low are computed in Amperes for use in the clamping logic:

Algorithm 1

```
high = load_params.high(line) * nominal_current;
low  = load_params.low(line)  * nominal_current;
```

At the very end of the function, these are overwritten back to fractions before being output to the thresholds port:

Algorithm 2

```
low  = load_params.low(line);   % ← overwrites to fraction
high = load_params.high(line);  % ← overwrites to fraction
```

This is an intentional design choice: the Ampere values are needed for the clamping math (gain correction) inside the function, but the output Thresholds port must carry normalised fractions so the scope can overlay them with the normalised Load signal correctly.

2.2.3. Profile change detection - the initial_sign logic

This block only executes when the line input has just changed value (jump to a new profile index). It performs three actions:

- Resets tmr0 = tmr: restarts the internal period clock from zero for the new profile
- Out-of-range guard: if line > 8, returns zero gain and zero thresholds safely
- Sets initial_sign: a critical edge-case handler

When switching profiles mid-simulation, old_value (the current integrator output) may be above the new profile's high threshold. Without correction, a positive gain would drive the current further away from the target rather than toward it. Setting initial_sign = -1 flips the ramp direction so the current descends toward high first. Once old_value <= high, initial_sign resets to +1 and normal operation resumes.

This produces the characteristic staircase descent visible in Figure 2-3 when transitioning from Profile 2 to Profile 1. At each 0.1 ms step (10 kHz) both half-phases push current downward until it falls below the new high threshold:

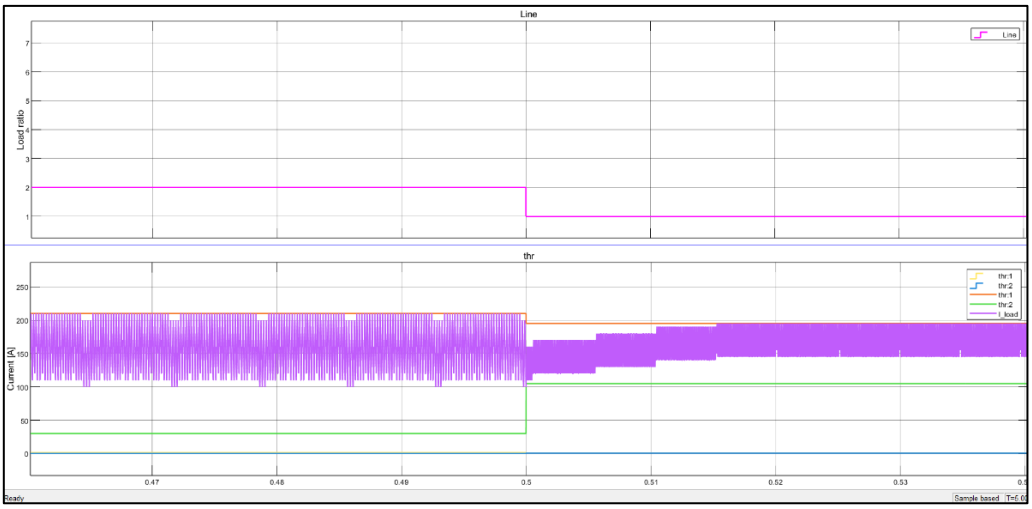


Figure 2-3 - Zooming into 0.46–0.56 s shows the staircase descent of Profile 2 to 1 transition

2.2.4. The timer and state machine

delta = tmr − tmr0 is simply elapsed time since the start of the current pulse period. This is used to determine which phase of the pulse we are in (detailing in the Table 2-1 below):

Table 2-1 - Phase State Machine summary

Phase	Condition	gain value	Effect on current
Rising	0 < delta ≤ t_high	+sr	Ramps up toward high at slew rate
Falling	t_high < delta ≤ t_high + t_low	−sr	Ramps down toward low at slew rate
Idle/Restart	delta > t_high + t_low	0 (tmr0 resets)	Holds; new period begins next step

An important note is that the idle phase (gain = 0) is present structurally but inactive for all 8 profiles, since each profile is defined with t_high + t_low exactly equal to one full period. For example, Row 7 has t_high = 50 ms and t_low = 28.6 ms giving a 78.6 ms period with no idle gap.

2.2.5. The threshold clamping - preventing overshoot

Without clamping, the integrator would overshoot the threshold on the final timestep of each phase, since gain × Ts may be larger than the remaining gap. The way to do the clamping check on the rising phase is:

Algorithm 3

```
1:    if initial_sign*(old_value) >= (high - gain*Ts)*initial_sign
2:        gain = (high - old_value) / Ts;
3:    end
```

This is like anticipating the next step: if the following step would overshoot, the gain is recomputed to land exactly on the threshold. The same logic applies symmetrically for the falling phase toward low. This means the waveform ramps linearly at the slew rate, then snaps cleanly onto the threshold at the last step, regardless of how large Ts is.

Here is a worked example for profile number 3, rising phase (Ts = 1×10^{-5} s):

Normal gain step: 1×10^{6} A/s × 1×10^{-5} s = 10 A per timestep

Suppose old_value = 255 A, high = 262.5 A (the peak seen in the simulation graph)

Remaining gap = 262.5 - 255 = 7.5 A

Since 7.5 A < 10 A, clamping activates: gain = (262.5 - 255) / 1×10^{-5} = 7.5×10^{5} A/s

Next step: current = 255 + 7.5×10^{5} × 1×10^{-5} = 262.5 A exactly

Without clamping the current would reach 265 A, a 2.5 A overshoot. Clamping eliminates this regardless of timestep size.

Another example is line 1 (with 10kHz, Ts = 1e-5) the current would oscillate around the thresholds. With clamping, it locks cleanly to exactly 195A at the high threshold (for Line 1). The low threshold of 105A is never reached, as the available fall of 50A leaves the current oscillating around ~145A.

2.2.6. Full threshold analysis for all 8 profiles

Whether the current reaches its target threshold in each phase depends entirely on whether the available current swing (slew rate multiplied by phase duration) meets or exceeds the required swing (high minus low, in Amperes):

Available rise = sr × t_high [A]

Available fall = sr × t_low [A]

Required ΔI = (high% - low%) × nominal_current [A]

Reaches threshold if: Available >= Required

A full threshold analysis for all 8 OCP profiles at 150 A nominal current can be obseved in Table 2-2 below (*top-anchored).

An example for profile 1 (at 150A nominal) presumes:

sr = 1×10^{6} A/s, t_high = 0.05 ms = 5×10^{-5} s

Max achievable rise = 1×10^{6} × 5×10^{-5} = 50 A

But: high − low = (130% − 70%) × 150A = 90 A needed

Table 2-2 - Full Threshold Analysis- complete table - All 8 Lines @ 150A Nominal

Line	Low %	High %	Low [A]	High [A]	t_low	t_high	SR	Req. ΔI [A]	Avail Rise [A]	Avail Fall [A]	Reach High?	Reach Low?
1	70%	130%	105	195	50 µs	50 µs	1 A/µs	90	50	50	Yes *	No
2	20%	140%	30	210	100 µs	100 µs	1 A/µs	180	100	100	No	No
3	40%	175%	60	262.5	2.5 ms	2 ms	1 A/µs	202.5	2000	2500	Yes	Yes
4	0%	80%	0	120	1 ms	1 ms	0.1 A/µs	120	100	100	Yes*	No
5	20%	140%	30	210	50 ms	50 ms	1 A/µs	180	50000	50000	Yes	Yes
6	60%	140%	90	210	50 ms	50 ms	1 A/µs	120	50000	50000	Yes	Yes
7	30%	140%	45	210	28.6 ms	50 ms	1 A/µs	165	50000	28600	Yes	Yes
8	30%	175%	45	262.5	0.43 ms	0.4 ms	1 A/µs	217	400	430	Yes	Yes

Here moving from profile 2 to 1 (Figure 2-3) is done when it's left at high, not low. The process is:

Rise: starts at ~150A, ramps +50A → lands at 200A ≈ high (clamping activates)

Fall: starts at 200A, ramps −50A → lands at ~150A (105A never reached)

Repeat

The triangle has settled into a band that is anchored at the top (high), not at the bottom, each rise ends exactly at high, and this is where clamping is involved. So, it fires on the last step to land precisely on high, but the fall only covers 50A downward, leaving it well above low. The reason it reaches high but not low is that in steady state the oscillation self-anchors to the top, each cycle the rise starts exactly 50A below high, travels exactly 50A, and clamping lands it precisely there. The bottom just floats wherever the fall runs out.

2.2.7. Profiles that do not reach their lower thresholds (profiles 1, 2, 4)

For Profiles 1, 2, and 4, the slew rate is too slow relative to the pulse duration to complete the required current swing. The phase clock (delta) advances past t_high (or t_low) before the current gets close to the threshold, and the FCN immediately reverses the ramp direction. The result is a triangular waveform that never flattens at the low limit. For profile 1, the high threshold is still reached via the top-anchored clamping mechanism described in Chapter 2.2.6 while only the low threshold is missed.

Profile 4 is the most extreme case: it uses 0.1 A/µs (ten times slower than the others), and even with a relatively long 1 ms window, the available swing (100 A) is still less than the required 120 A.

Profiles 3, 5, 6, 7, 8 reach both thresholds. The current ramps up quickly, holds flat at high for the remainder of t_high, then ramps down and holds flat at low, having a classic trapezoidal/square shape.

2.3. Scope outputs

The top graph in Figure 2-4 scope represents the "Line": integer 1 to 8, with its random pick of steps. Here, for the simulation duration of T=5s, it cycles 2, 1, 5, 7, 3, 6 and so on. The bottom graph consists in two "thr" lines:

- Orange line representing the high threshold:

  normalized fraction × 150A = Amps;

- Green line representing the low threshold;
- Purple is the I_load = Load × 150A.

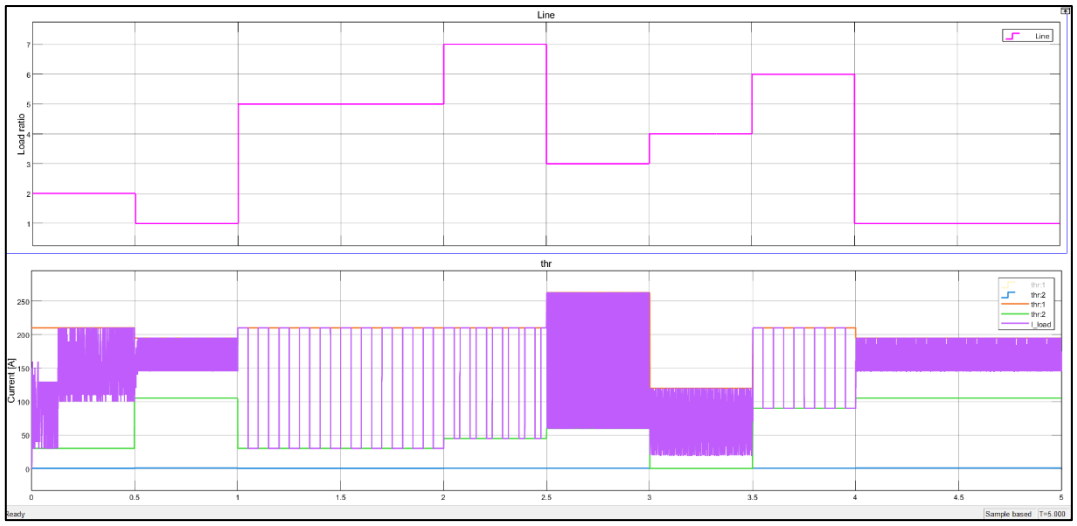


Figure 2-4 - Simulation scope output for T=5s

Key observations from the scope, which can be helpful in the design of the circuits:

- Profile 2 (0-0.5 s): 5 kHz, large amplitude. It is a high-density triangular waveform, having neither threshold reached.
- Profile 1 (0.5-1 s): 10 kHz, moderate amplitude. Current bounces between ~150 A and ~200 A. It has the high threshold (orange) reached, while the low threshold (green, ~105 A) is not.
- Profile 5 (1-2 s): 10 Hz, 50 ms pulses. It is a clean trapezoidal shape, with both thresholds reached, while the slow period allows the full 180 A swing at 1 A/µs (180 µs ramp out of 50 ms available).
- Profile 7 (2-2.5 s): 12.7 Hz, asymmetric t_high/t_low. Both thresholds are reached, being a flat-top, flat-bottom trapezoid.
- Profile 3 (2.5-3 s): 222 Hz, 2/2.5 ms phases. Both thresholds are reached, and it has a large amplitude swing (262.5 A peak at 175% of 150 A).

- Profile 4 (3-3.5 s): 500 Hz, 0.1 A/μs slew rate. Triangular waveform with the high threshold reached via top-anchored clamping, but the low threshold not reached, due to the reduced slew rate.

2.4. Control parameter summary

The summary of the main parameters used and how they affect the current waveform is presented in Table 2-3 below.

Table 2-3 - Summary of parameters and their effects on the current waveform

Parameter	Controls	Impact
sr	Ramp slope (A/s)	Faster sr = steeper transitions
t_high	Duration of rising phase	If sr × t_high < req. ΔI, high not reached
t_low	Duration of falling phase	If sr × t_low < req. ΔI, low not reached
high, low × nominal	Target thresholds	Defines amplitude of current swing
Ts	Simulation timestep	Only affects clamping precision, not waveform shape
delta = tmr - tmr0	Phase clock within pulse period	Determines which phase (rising/falling/idle) is active

As shown in the table, the waveform shape depends on the interaction between sr, t_high, and t_low, with Ts only affecting the precision of the threshold clamping. This means that it can be determined whether both thresholds will be reached by evaluating the inequality sr × t_phase ≥ ΔI_required, which can be evaluated analytically before running any simulation. Profiles where this condition holds in both phases produce trapezoidal waveforms with flat plateaus at both limits, while profiles where it fails in one or both phases produce triangular waveforms that never fully settle. The active model's architecture therefore makes its behaviour predictable from the parameter values alone, with no dependence on circuit physics or supply conditions, and so representing the starting point for the passive model comparison in Chapter 5.

3 The Controlled Current Source

3.1. Connection to current source model

The signal model described in Chapter 2, the FCN, integrator, and profile selector, produces a current command: a time-varying value that says, "draw these many amperes right now." To turn that command into an actual electrical simulation, it must drive a circuit connected to the 800 V DC bus. The first and simplest circuit for this purpose is the controlled current source, which takes the FCN output and imposes it directly on the bus with no additional filtering or impedance, making it a direct electrical implementation to the signal model: the current waveform on the bus is exactly what the FCN computes. This circuit was designed as part of this work and is referred to throughout this thesis as the active model.

3.2. Circuit components

The circuit is implemented using Simscape Electrical [17]. The first type of circuit that can be implemented is the one with the controlled current source (Figure 3-1), which validates the shape and amplitude of the current waveform simulated against the OCP Table 8 specifications. It contains three essential elements:

- DC_+ and DC_- terminals. They are connection points to the 800 V DC bus;
- C1 (capacitor): represents the capacitance at the load, which partially buffers fast current transients and smooths voltage;
- Controlled current source (arrow symbol): this is the active element. It draws exactly the current commanded by the incoming profile signal, regardless of what the bus voltage is doing.

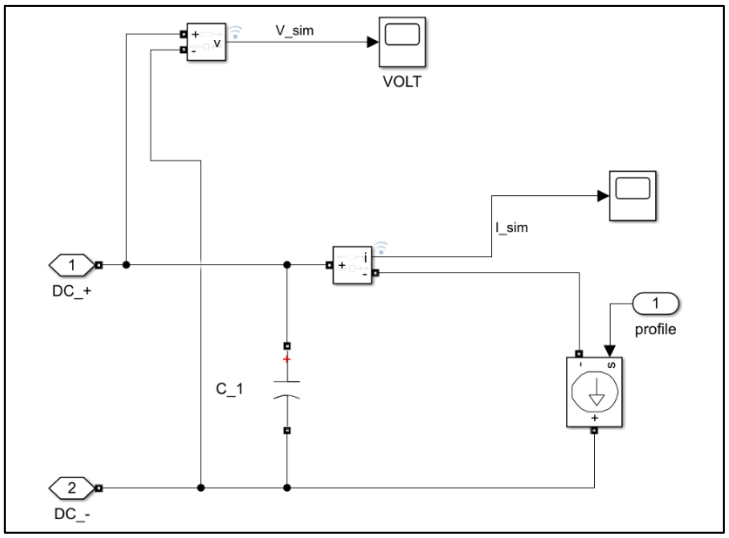


Figure 3-1 - Controlled current source circuit (Simulink)

There are however some issues that arise with this model even though it replicates very well the current waveform shape specified by OCP Table 8. The problems are related to the absence of physical phenomena:

- Voltage independence: the controlled current source manipulates the current waveform, without taking into consideration the bus voltage. Real GPU converters do not have a constant current output (regardless of impedance), the voltage drop can affect their current input to some degree.
- No bus impedance: voltage drop, overshoot, ringing, and settling transients are not captured and they sometimes are needed for detail for some users. These effects only appear in the passive load model, which includes R_grid, C_DC, and L_slewRate.
- The profile selector is ideal: the random profile selector is a test convenience, not a representation of actual GPU workload sequencing. Real workloads have statistically structured transitions, not uniformly random ones.

These limitations motivate the development of the passive load model in Chapter 4, where current is governed by Ohm's law (I = V/R), addresses the voltage domain and is the correct tool for observing bus voltage transients under the same profiles. Therefore, with adding missing physical elements, the two topologies can be compared directly.

4 Passive Load Model

The passive load model takes a different approach from the controlled current source examined in the chapter before, constructing a resistive circuit whose physical behaviour generates the current naturally. The consequence of this is seen in the real circuit effects: inductive transients, settling time, which is limited to the time constant, as well as current changes that depend on voltage, all appear in the output. This makes the passive model much harder to tune, but a closer representative of what a real PSU will encounter.

4.1. Circuit description

In Figure 4-1 below, the Simulink circuit topology for the passive model can be seen. The electrical circuit has five distinct elements, each representing a real component in a data centre power delivery path.

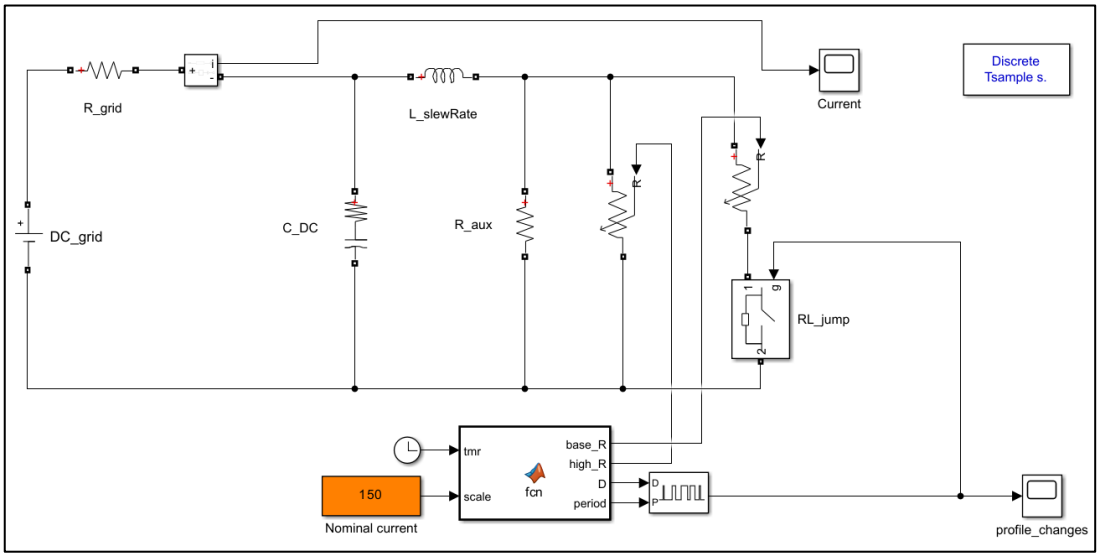


Figure 4-1 - Passive load model circuit in Simulink

The elements scope and meaning are explained as follow:

- The DC_grid is basically the power supply, and so, it represents the 800V DC bus coming from the AC/DC power shelf as defined in the OCP Diablo 400 rack architecture.

- R_grid is a small series resistance modelling the cable resistance of every cable that connects the PSU to the server rack. When current flows, this causes a voltage drop (I^2R) which is direct proportional to the current.
- C_DC is a bulk capacitor placed across the bus at the rack input, acting like a local energy buffer. When there is a sudden current overshoot, the capacitor supplies the extra charge instantly, stopping the bus voltage from dipping too hard, giving it time to catch up. The sizing of this capacitor against dI/dt is critical as the bus voltage must stay within the ±3% limit (seen in the OCP Table 6).
- L_slewRate is a series inductor between C_DC and the load resistors. This is the element that physically enforces the slew rate and resists sudden changes in current: the voltage across it is $V = L \times dI/dt$ [18], which rearranges to $dI/dt = V/L$. So, the slew rate can be calculated:

  the bus voltage=800 V meaning that L = 0.8 mH.

  Using: L = V/slew_rate

  the slew rate which the circuit can produce is:

  $800 / 0.8\times10^{-3} = 1\times10^{6}$ A/s = 1 A/μs

  This value matches the OCP specification from Table 8 (Table 1-2).

- The load itself (the GPU server rack) consists of two resistors:
  - R_aux (labelled high_R in the FCN output), which is always connected.
  - RL_jump (labelled base_R in the FCN output), which is switched in or out by a relay block driven by a pulse generator [17], whose period and duty cycle comes from the FCN.

  When RL_jump is disconnected, only R_aux carries current, giving the low idle current. When RL_jump connects in parallel with R_aux, the combined resistance drops and the current rises toward the high level (active).

4.2. FCN logic

This FCN of the passive load model is written with the help of the active model's current scope as a calibration reference, whose FCN had to find the gain (rate of change of current). Therefore, the passive load FCN purpose is to compute resistance values and timing parameters. So, its outputs are base_R, high_R, duty cycle D, and pulse period, all being fed to the relay and pulse generator that drive RL_jump.

The FCN has an array of frames which are determined by boundary times: [0.5, 1, 2, 2.5, 3, 3.5, 4, 5] seconds. A counter, which represents the current frame index,

increments every time the tmr value passes one of those array values. So, this process selects the next row of the profile tables, playing out in a fixed order.

Resistance values are derived directly from Ohm's Law (V = IR), so, for each frame the FCN reads the target currents I_high and I_low, then computes:

Algorithm 4

```
R_I_high = V / I_th_high(frame)    % resistance to deliver I_high
R_I_low  = V / I_th_low(frame)     % resistance to deliver I_low
```

These are the ideal resistances for a 800 V bus. However, the actual circuit outputs are derived using the parallel resistor relationship. Since R_aux (high_R) always remains in circuit and must produce alone I_low during idle, it is set equal to R_I_low. The switched resistor RL_jump (base_R) is then chosen so that its parallel combination with high_R equals R_I_high:

Algorithm 5

```
high_R = R_I_low
base_R = (R_I_high × R_I_low) / (R_I_low - R_I_high)
```

The name of high_R refers to its high value, basically a small current flows through a big resistor. The name of base_R comes from it being the base of the parallel network and it is used during the active pulse, bringing the total resistance down to R_I_high and the current up to I_high.

To make it clearer, here is the Table 4-1 that summarizes connections between the 4 resistors functions during the idle and active:

Table 4-1 - All the resistances explained

State	Simulink name	FCN output	Connection	How to calculate	Current result
Idle	R_aux always connected	high_R = R_I_low	Only R_aux in circuit. RL_jump open.	R_I_low = V / I_low= =800 / I_lowhigh_R = R_I_low × scale/150	I = V / R_aux= V / high_R ≈ I_low
Active	R_aux always connected	high_R = R_I_low	R_aux stays in circuit. RL_jump also closes, both in parallel.	Same as idle: R_aux value does not change between states	I = V / (R_aux ‖ RL_jump)= V / R_I_high ≈ I_high Parallel combination is designed to equal R_I_high exactly
Active	RL_jump switched in by relay	base_R =parallel complement	Switched in parallel with R_aux when pulse generator fires.	R_I_high = V / I_high= =800 / I_highbase_R = =(R_I_high × R_I_low) / (R_I_low − R_I_high) × scale/150	

The FCN also contains two conditional corrections intended to account for cases where the $\tau = L/R$ [19] time constant is longer than the pulse duration:

Algorithm 6
1: if L/R_I_low >T1 (the idle phase duration), R_I_low is multiplied by 1.1.
2: if L/R_I_high > (T2 - T1) (the active pulse duration), R_I_high is divided by 10

The intent of these adjustments is to push the resistance toward a value whose time constant is shorter than the pulse, so the current has a better chance of settling. In practice, covered in Chapter 5.3, these corrections introduce more problems than they solve.

4.3. Waveform analysis

The passive model scope (Figure 4-2) shows eight distinct segments corresponding to the eight frames, running for T=5s. Understanding why each segment looks the way it does requires applying the RL circuit settling equation to the specific L/R ratio (time constant) and pulse duration of each frame.

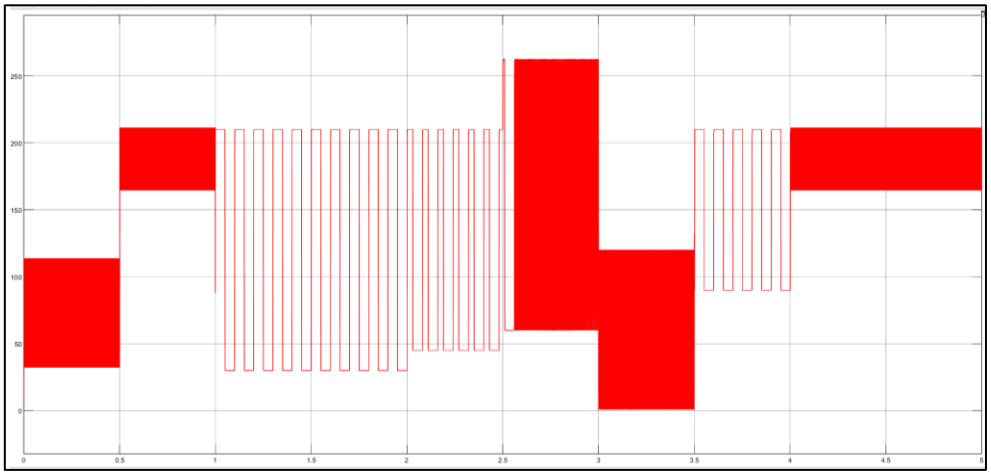


Figure 4-2 - Passive load model current scope output

When resistance switches to R_high (active pulse), current does not jump instantly depending on τ = L / R_total: $I(t) = I_target \times (1 - e^{-t/\tau})$ [19].

How much of the target it reaches in the available pulse time T1 is: $(1 - e^{-T1/\tau})$.

The conclusions are:

- when T1 >> τ the exponential decays to zero and the current reaches I_target, settling fully, and giving a square waveform.
- when T1 <= τ, the current only reaches partway to the target current before the pulse reverses, producing a triangular waveform.

Therefore, there are two types of frames based on the condition above:

Slow frames like, 3, 4, 5 and 7, which are equivalent to profiles 5,7,3,6, operate at 10-222 Hz with pulse durations of 2.5-50 ms (slow). Their time constants (τ) are around

0.2-0.9 ms, giving T1/τ ratios between 8 and 238. These frames settle completely within the first fraction of a millisecond of each pulse and hold flat at their target levels for the rest resulting so in a clean square wave waveform.

Fast frames like 1, 2 and 8, which are equivalent to profiles 2 and 1, operate at 5-9 kHz, with time constant τ ≈ 0.2 ms, the T1/τ ratio drops to 0.3–0.4, meaning the current reaches only 27–35% of the target value before ramping back down. This results in a dense triangular waveform that never flattens at either up or down limits. However, this is a correct physical behaviour of an RL circuit which operates faster than its own time constant allows and has its constraint with L not R, so, no resistance value can change this.

A **special mention** is frame 6 (profile 4), set in the t = 3 to 3.5 s and has OCP I_low = 0%, but the passive FCN uses 1 A as a workaround to avoid the singularity in the parallel resistance formula (R_I_low → ∞ as I_low → 0). It is the most numerically problematic frame: R_I_low = 800/1 = 800 Ω, which when substituted into the parallel formula gives a base_R of approximately 6.7 Ω, and when RL_jump disconnects, only R_aux = 800 Ω carries the idle current. The 120 A active pulse settles well, but the near-zero idle level the condition doesn't work anymore when R_I_low is far larger than R_I_high, so, the idle current seen on the scope is unreliable.

Now, looking at **transitions between the frames**, it can be observed that not all of them are clean. The most significant is at t = 2.5 s, the transition from frame 4 to frame 5 (7 to 3) produces a visible current spike (Figure 4-3 and zoomed in Figure 4-4) that overshoots the frame 5 target before settling. This is due to the inductor L_slewRate, which was carrying 45 A of stored energy from frame 4's idle current. When frame 5's resistance connects, the full bus voltage minus the small resistive drop appears across the inductor:

$$V_L(t = 2.5s^{+}) \approx 800 - (45 \times 3.05) \approx 663\ V$$

$$dI/dt = V_L / L = 663 / 0.8\times10^{-3} \approx 829{,}000\ A/s \approx 0.83\ A/\mu s$$

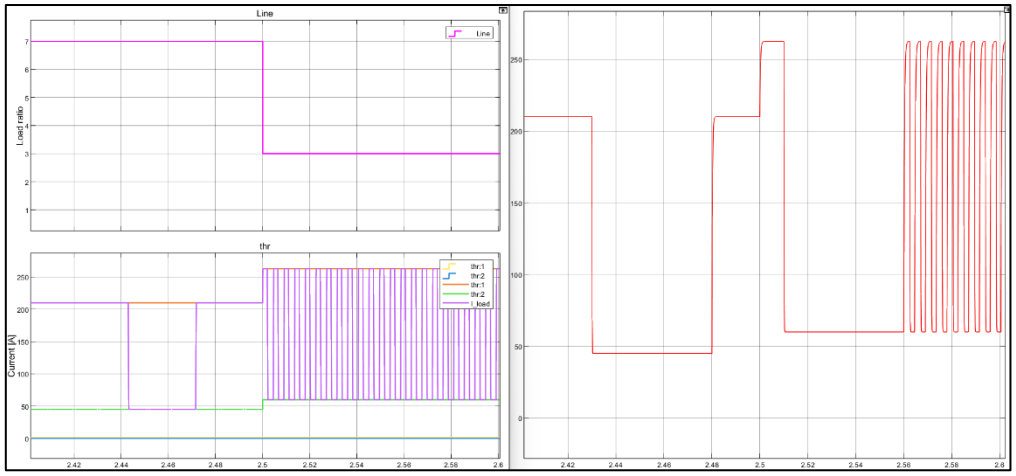


Figure 4-3 - Profile 7 to 3 transitions. Left-current model. Right-passive model

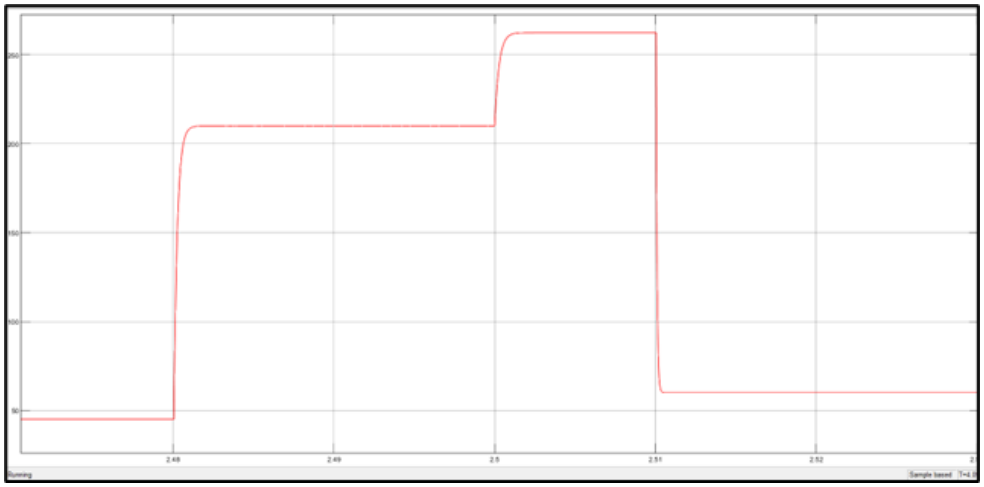


Figure 4-4 - Passive load during the 7 to 3 frame transition: exponential transitions due to RL time constant τ = L/R" and "inductive overshoot at t ≈ 2.5 s

Transitions between slow frames at t = 1 s, 2 s, and 3.5 s are relatively smooth since the inductor currents are in the accepted intervals and the resistance changes are smaller. Transitions into fast frames at t = 0.5 s (Figure 4-5) and 4 s show an adjustment period (seemingly exponential) as the circuit moves from a settled level into the fast oscillation waveform.

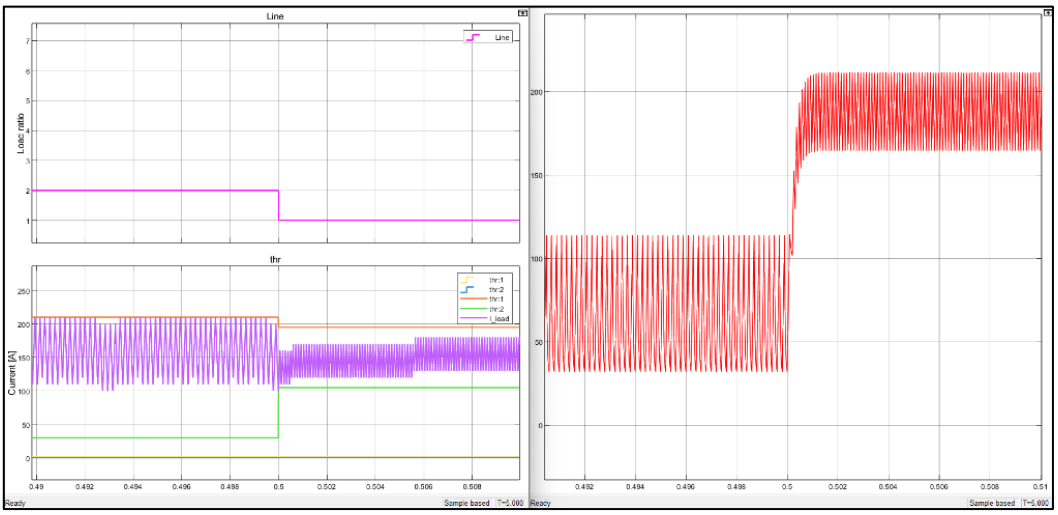


Figure 4-5 - t=0.5 s transition between 1 and 2 frames (profiles 2 to 1). Left-active model. Right-passive model

All transition spikes are physically correct because they represent how the inductance affects the current before settling when circuit topology changes. That is why the OCP Diablo 400 Table 6 is essential when designing the load with the help of the ±3% voltage overshoot requirement. This inductor issue represents exactly the kind of transient that a real PSU must be able to withstand during workload transitions, and

its presence in the passive model but not in the current source model is one of the key differentiators between the two approaches. Here is a clearly explained in Table 4-2 for all the frames in the passive load model (Note: ”frame” is different than “profile”):

Table 4-2 - Passive load model values

Frame	Profile	Time (s)	I_high / I_low (A)	Period T2 / Pulse T1	Freq	τ = L/R_high (ms)	T1 / τ	Settles	Waveform shape
1	2	0 – 0.5	210 / 30	0.20 ms / 0.09 ms	5 kHz	0.210	0.43	Partial	Triangular
2	1	0.5 – 1	195 / 105	0.11 ms / 0.06 ms	9 kHz	0.195	0.31	Partial	Triangular
3	5	1 – 2	210 / 30	100 ms / 50 ms	10 Hz	0.210	238	Full	Square wave
4	7	2 – 2.5	210 / 45	80 ms / 30 ms	~12 Hz	0.210	143	Full	Square wave
5	3	2.5 – 3	262.5 / 60	4.5 ms / 2.5 ms	222 Hz	0.261	9.6	Full	Square wave + spike
6	4	3 – 3.5	120 / 1	2 ms / 1 ms	500 Hz	0.120	8.3	High only	Near-zero drop
7	6	3.5 – 4	210 / 90	100 ms / 50 ms	10 Hz	0.210	238	Full	Square wave
8	1	4 – 5	195 / 105	0.11 ms / 0.06 ms	9 kHz	0.195	0.31	Partial	Triangular

Note: the passive FCN uses T2 = 0.11 ms for profile 1 (≈ 9 kHz), a slight deviation from the OCP-specified 10 kHz (T2 = 0.10 ms), introduced during tuning.

4.4. Identified limitations

Two numerical issues arise in the FCN. The first concerns the "/10" correction applied when the inductor time constant L/R_I_high exceeds the active pulse duration. The intention is to reduce the resistance so that the time constant decreases and the current settles within the available pulse time. However, R_I_high appears in the denominator of the parallel resistance formula:

$$\text{base_R} = (\text{R_I_high} \times \text{R_I_low}) / (\text{R_I_low} - \text{R_I_high})$$

Therefore, when R_I_high is divided by 10, the denominator (R_I_low − R_I_high) grows, and base_R approaches zero or becomes numerically unstable. This type of base_R, which also means near-zero total resistance when RL_jump connects, allowing an unbounded current demand ($I = V/R \rightarrow \infty$). In this condition, L_slewRate physically limits the peak current, but the inductance no longer governs the steady-state resistance target, as L_slewRate dominates the transient response across all affected frames. Removing the "/10" correction produces a shift in the waveform amplitude of the triangular waveform in frames 1, 2 and 8 without improving settling. A rate limiter was also applied to the FCN resistance outputs to attenuate frame transitions: this reduced the spike at t = 2.5 s but suppressed the pulse amplitude in the fast frames.

As an illustration, consider frame 5, which has I_high = 262.5 A, so R_I_high = 800/262.5 = 3.05 Ω. The time constant τ = L/R = 0.8 mH / 3.05 Ω = 0.26 ms. The active pulse duration (T2−T1) = 4.5 ms − 2.5 ms = 2ms. Since 0.26 ms < 2 ms, it works well. On the opposite end there is the first frame: R_I_high = 800/210 = 3.81 Ω, so τ = 0.8 mH/3.81 = 0.21 ms and the active pulse duration = 0.09 ms. Since 0.21 ms > 0.09 ms, the condition is met and R_I_high becomes 0.381Ω, base_R collapses toward zero, producing the instability described above.

The second issue is the near-zero I_low value in frame 6: when I_low = 1 A, R_I_low = 800 Ω. The parallel formula then produces base_R (6.73 Ω) that is almost identical to R_I_high (6.67 Ω), but when RL_jump disconnects during the idle phase, only R_aux remains, and R_aux = high_R = R_I_low = 800 Ω. However, since resistance calculations are based on an ideal version Ohm's Law (V= 800V), when considering approximations or voltage drops and parasitic impedances, any small deviation in actual bus voltage due to parasitic impedances or resistive drops produces a proportionally large error in the idle current at R_I_low = 800 Ω. Consequently, the idle current at the output is unreliable and does not reproduce the intended 1 A idle level.

The transition spike at t = 2.5 s could be reduced by increasing C_DC, by a larger capacitor absorbing some of the inductive energy, reducing the peak overshoot. The 800 VDC pdf recommends this approach, noting that capacitors placed close to the rack are effective for transients as fast as 400 μs, but there is a sizing conflict. The capacitance that would be used to reduce the spike, of value 2.8 mF, can affect and slow the RC response of the fast frames (1, 2 and 8), distorting the fast-frame waveform shape entirely. Using a much smaller capacitor, around 10–50 μF, limits the distortion to the fast frames but also provides only partial spike suppression. This trade-off reflects a real design constraint: the stored inductive energy in L_slewRate drives the spike, and the only way to eliminate it entirely would be to reduce L, which would violate the 1 A/μs slew rate. The spike is therefore considered as a physically correct observation of what the inductance of a real power distribution path does during a profile transition.

In conclusion, the decision is to leave the FCN as it is and document these as limitations rather than apply corrections that introduce larger distortions elsewhere in the waveform. Moreover, the physical constraint of the inductor is that, when modelling the circuit, it cannot simultaneously represent the correct slew rate for all eight profiles (profile 4 is the only one with 0.1). Before connecting either model to a supply grid, Chapter 5 compares their architectures and waveform behaviour directly, establishing the qualitative basis for the quantitative comparison in Chapter 7.

5 Comparison of Modelling Approaches

This chapter establishes the qualitative basis for the formal comparison in Chapter 7. The active model is voltage-independent by construction, as it draws exactly the commanded current regardless of bus voltage, making it the most idealised representation of a tightly regulated downstream converter. The passive switched-resistance model, by contrast, represents the physical impedance-switching behaviour that actually occurs in the power delivery path. The two modelling approaches configured to represent the same OCP Table 8 profiles, are examined on why the two models behave differently, by tracing the differences back to their FCN architectures, circuit topologies, and modelling philosophy.

5.1. Fundamental difference in methodology of models

The two models differ in one fundamental way: the current source model has the FCN directly commanding what current must flow, while the passive model uses the FCN to set a resistance target and the way the current that flows is determined by the circuit physics. This is why voltage-domain effects such as voltage drop, inductive transitions and capacitor dynamics are absent from the current source model but fully present in the passive one.

This distinction is also reflected in their FCN architectures. The current source FCN outputs a gain in A/s, which an integrator turns into current with a feedback path so it can clamp against overshoot, controlling the waveform. The passive FCN outputs resistance values and timing signals, setting up the conditions for the circuit to produce a current but has no certainty of what that current is for sure.

5.2. Circuit topology comparison

It is also worth looking at what components are found in the two circuits, since the circuit topology determines what appears in the simulation output.

The fundamental point this Table 5-1 makes is that the passive circuit models every stage of the power delivery path, while the current source model has all of that into a single ideal element. The inductive spike, the voltage droop, the RC settling in fast frames emerge from the passive circuit having interactions between real components. None of them can appear in the current source model because there is nothing in that circuit for them to come from.

Table 5-1 - Summary of the 2 models

Element	Current source model (active)	Passive load model
Power source	DC_grid (800V)	DC_grid (800V)
Line impedance	Not modelled	R_grid= cable resistance
Capacitance	C1 — local buffer only	C_DC — sized to limit voltage transients
Slew rate	Software: gain parameter in FCN	Hardware: L_slewRate inductor (L = V/sr)
Load element	Ideal current source, so, voltage independent	R_aux always connected, RL_jump switched
Load control	FCN outputs gain (A/s)	FCN outputs resistance values and timing
Energy storage	None, due to ideal source has no memory	L_slewRate stores energy between frames

5.3. Waveform behaviour by profile

For the slow profiles (frames 3, 4, 5, and 7, equivalent to profiles 5,7,3,6), the two models produce the same kind of results: both show square-wave or trapezoidal waveforms that reach their target current levels and hold flat. The only difference is that the current source reaches the targets exactly every time, since clamping prevents overshoot, while the passive model lands close to the targets within the targeted intervals given by resistance values and the bus voltage drop.

For the fast profiles frames (1, 2, and 8, equivalent to profiles 1 and 2) is where the models start to present differences. They both produce triangular waveforms since the pulse is too short for the current to complete the full swing, but for different reasons. For the current source model, the peak is set by how far the slew rate gets in the available time: sr × t_high. In the passive model the peak is set by the RL time constant $\tau = L/R$, which is a physical property of the circuit. The two give different amplitudes and the passive model's peak shifts depending on the resistance of each frame, while the current source model is more predictable.

When it comes to transitions between frames, the clearest difference between the two models is the spike at t = 2.5 s. The current source model transitions between frames smoothly: when the line input changes, the FCN detects the change, resets its timer

and uses initial_sign if needed steer the current in the correct direction toward the settling in frame 5 within a few cycles. The passive model produces a sharp current overshoot at the same timestamps because the inductor was carrying 45 A from frame 4 and the sudden drop in resistance forces that stored energy out very fast. An ideal current source carries no stored energy, so this phenomenon does not happen there.

Lastly, frame 6 (profile 4) highlights a specific limitation of the passive approach, which cannot do this because there is no way to build a purely resistive load that delivers exactly 1 A at 800 V without the large resistance mismatch breaking the parallel formula, as discussed in Chapter 4.2. The current source model handles I_low = 0% correctly in terms of FCN logic (the clamping will snap to 0A if reached), but in practice profile 4 is slew-rate limited and the current only falls to ~0.127 pu (~19 A) before the phase reverses, as confirmed in Table 7-4.

The passive model's real components mean that the scope shows things the current source model simply cannot produce, like the interactions between L_slewRate, C_DC and the switching resistors create visible events. For example, the spike at t = 2.5 s reveals how the circuit behaves during a profile transition. In the current source model that same transition is invisible: the FCN resets its timer, and the waveform moves cleanly to the new profile.

In conclusion, both are correct representations of what their respective circuits do. The passive model gives more physical detail and is the more useful tool when the question is about what happens at the boundaries between profiles, not just within them: the inductive spike, the exponential settling, and the near-zero idle current failure of frame 6 are all physical phenomena that only the passive circuit can produce. The active model, by contrast, gives a current waveform that complies with the OCP Table 8 specifications: the FCN-integrator loop commands exactly the right ramp rate, clamps precisely at the thresholds, and handles edge cases like the 0% idle state cleanly. The following chapters connect both models to three reference supply grids and use that controlled environment to formally quantify which approach more faithfully reproduces the OCP pulse profiles.

6 Reference Grid Models for Validation

In the previous chapters the active and passive load models are analyzed separately, each being connected to an ideal 800 V DC bus. However, that does not test how the two models behave when they share the same type of supply path, which can "mimic" what it is like in a real data center rack. Before any comparison between them is made, both models must be connected to the same grid representation, so that any difference in their outputs comes from the load model itself rather than from an inconsistent supply.

The aim of this chapter is therefore to introduce three reference grid models that progressively add physical detail to the supply side, starting from a purely resistive cable, then adding a diode and ending with a regulated voltage source. Each grid is applied identically to both the passive and active load models, providing a controlled environment in which the two modelling approaches can be evaluated on equal terms.

The supply chain topology used as reference for this work, an AC/DC active front-end converter feeding a DC bus through cable resistance and an uncontrolled rectifier diode, reflects the power delivery path captured in the simulation data files made available for this project. The parameters for these grid variants are identified from those files, capturing the electrical behaviour of this supply chain at six load levels between 25% and 200% of nominal current. These files provide the measured V-I operating points used to tune R_grid and the diode parameters in Chapters 6.1 and 6.2.

6.1. Tuning grid parameters from measurement data

Before the three grids can be built in Simulink, their parameters must be identified from the measurement data. Six files are provided, each captured at a different fraction of a real system's nominal current: 25%, 50%, 75%, 100%, 150%, and 200%. Each file contains four columns: the time vector, the input current measured before the diode and bus capacitor, the terminal voltage measured before the diode (it will become important later), and the computed reference current profile. Something to take into consideration is that the relationship between current demand and voltage drop is not strictly linear, which is expected due to the active front-end converter and a diode used. The files are large, each containing roughly 10.7 million samples at an

average sample period of about 3.75×10^{-7} s over 5 seconds of simulation. Due to this much data, the information cannot be loaded and displayed through standard MATLAB tools without down sampling.

For parameter identification, only the mean current and mean voltage in the steady-state window of each file is needed. The middle 30% of each simulation is used for this, avoiding the ramps at the beginning and end of transitions between profiles, providing a single (I, V) point per load percentage. In Figure 6-1 , the six operating points are plotted, and it can be seen that the voltage drop ($\Delta V = 800 - V_mean$) against the mean current gives the static V-I characteristic of the simulated supply chain, which becomes the tuning target for Grids 1 and 2. First, a linear fit of voltage drops versus current, forced through the origin to represent a purely resistive line, gives:

$$R_grid = 0.003967\ \Omega \approx 0.004\ \Omega, \quad R^2 = 0.9759$$

When allowing a constant offset to represent a diode forward voltage in series with the resistor ($dV = V_d + R \cdot I$) it gives:

$$Vd = -0.828\ V, \quad R = 0.004550\ \Omega, \quad R^2 = 0.9981$$

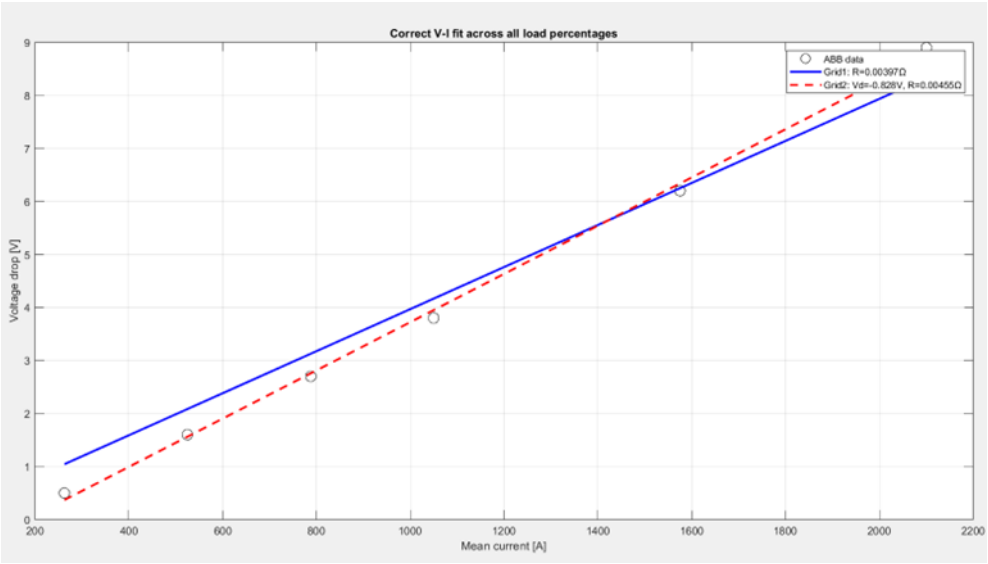


Figure 6-1 - V-I characteristic of the supply extracted from all six load percentages

The negative diode offset is physically impossible and meaningless: a forward-biased diode cannot produce a negative voltage drop. However, it is important to note that the terminal voltage in the measurement data provided is recorded before the diode, as stated in the data description. This explains why the offset regression returns to a negative value: there is no real diode drop in the measured signal, and the small negative number appears due to the fitting of a two-parameter model to purely resistive data. The cable resistance $R \approx 0.004\ \Omega$ identified by the single parameter fit is therefore the physically meaningful result, and the $R^2 = 0.976$ confirms a good fit. Even though the steady-state operating point matches Grid 1, the diode is still included in the Grid 2 model using default Simscape parameters ($V_f = 0.8$ V, $R_on = 0.001\ \Omega$). The

diode is still topologically relevant to the system, so keeping it in the model is necessary to capture some behaviors like reverse-blocking and transient switching during fast load changes.

Finally, the R^2 values validate that the models closely align with the provided data. The slightly higher R^2 of the Grid 2 fit (0.9981 vs 0.9759) reflects the slight curvature of low current data, consistent with the non-linear observation mentioned previously. However, this curvature is small enough that a negative offset is just a mathematical value of regression rather than a physical property.

There is a methodological choice here: the grid parameters were extracted from all six load percentages together rather than only from the percentage closest to the nominal operating point. The reason is that the cable resistance is a linear time-invariant parameter, with the relationship $dV = R \cdot I$ holding at every operating point with the same R. All six measurements are therefore observations of the same parameter under different excitation levels, producing a statistically robust value of R. If it were to restrict the fit to one percentage, it would have given a noisier estimate from a single operating point. Once R has been determined this way, it is fixed for all the following simulations without needing to be adjusted. The same logic applies to the diode parameters in Grid 2 and to the regulator setpoint in Grid 3.

6.2. Grid topologies in Simulink

Before describing the three variants, it is worth clarifying what the grid models are replacing and where they connect and how to keep the comparison between the models fair. In the passive model, the supply side already exists as a DC voltage source (DC_grid) in series with a resistor (R_grid), feeding the rest of the load circuit. The three grids replace this DC_grid and R_grid pair with progressively more detailed equivalents while the load circuit (C_DC, L_slewRate, R_aux, RL_jump) is never touched. In the active model (seen as a subsystem in Figure 6-3 b.), exposes two electrical ports, DC_+ and DC_−, which represent the connection point to whatever supply is driving the load. The grid model connects directly to these ports, with no other modification to the subsystem. In both cases, the grid sits in exactly the position occupied by the supply chain between the regulated AC/DC converter output and the load-side capacitor. The three variants represent three different levels of approximation of that chain: a purely resistive cable, a cable with a rectifier diode, and a fully regulated voltage source.

A key motivation for testing across three grids rather than one is to verify that the passive-vs-active comparison conclusion is not affected if tested under a different grid.

6.2.1. Grid 1: resistive source (V + R)

Grid 1 is the simplest Thevenin equivalent of a model DC bus [18]: an ideal 800 V voltage source in series with R_grid = 0.004 Ω identified in the previous chapter. It represents the cable resistance between the converter and the rack and ignores both the diode and any active regulation. In the passive model it replaces the original DC_grid block while in the active model it connects to the DC_+ and DC_− ports of the Circuit Load subsystem. Figure 6-2 and Figure 6-3 show the two implementations.

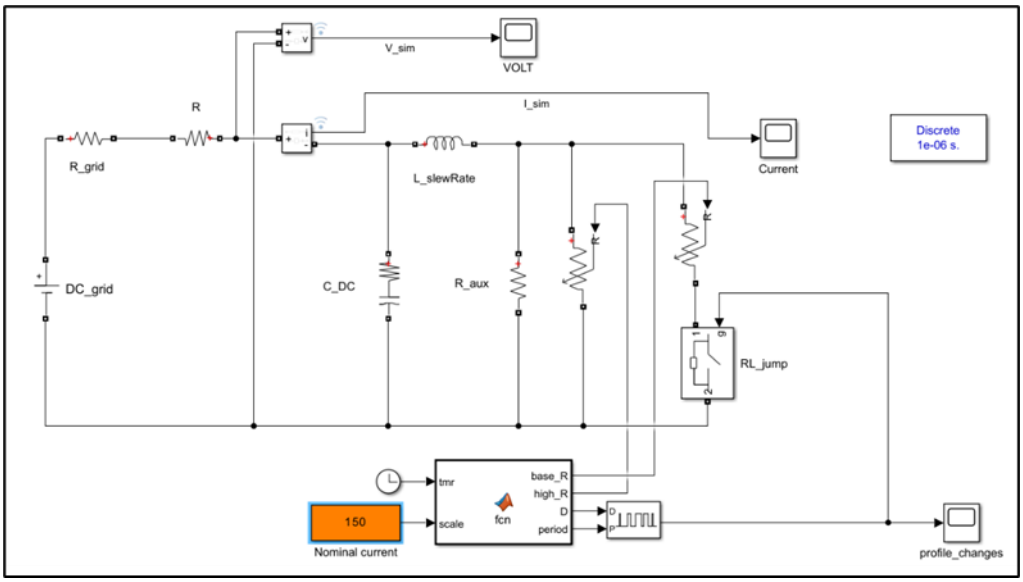
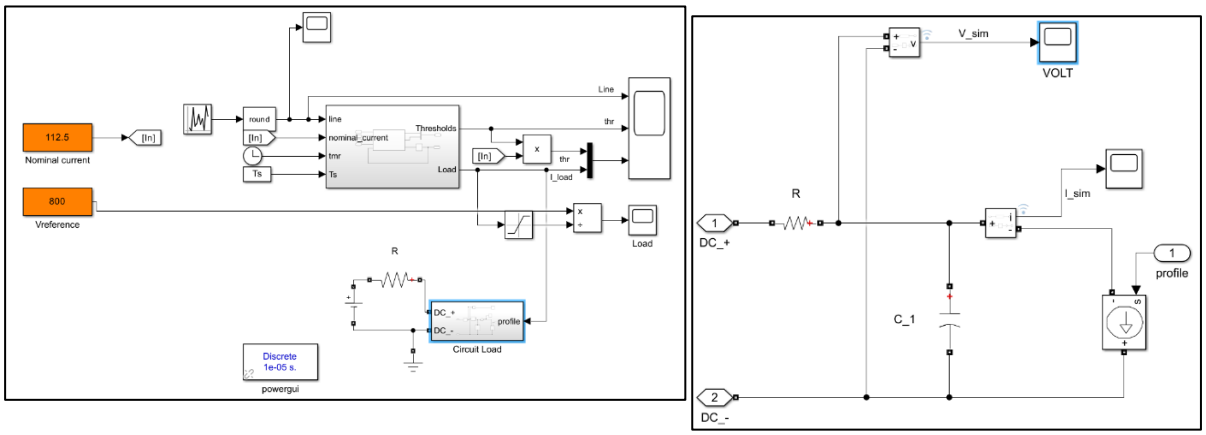


Figure 6-2 - Passive load model with Grid 1 (V + R) at the supply side

As seen during the original design, the passive model uses an ideal 800 V voltage source with no series impedance, which does not represent the cable drop present in a supply path. Grid 1 reintroduces the cable resistance identified in the 6.1 chapter, making the supply based on the measured data and not ideal.



(a) Active load model full (b) Active load subsystem

Figure 6-3 - Active load model with Grid 1 (V+R)

6.2.2. Grid 2: resistive source with diode (V + R + D)

Grid 2 adds an uncontrolled diode (set to V_f = 0.8 V and R_on = 0.001 Ω) in series after the resistor, immediately before the bus capacitor and the load. As discussed in Chapter 6.1, the V-I regression gave no measurable value for this diode, but it is included for topological correctness as it enforces a single direction of current flow and contributes to transients during fast load changes. The voltage measurement point in both models is kept before the diode, so that the measured voltage includes the diode drop. The active model, for this grid variant, has the same full model as Grid 1 (Figure 6-3 a.). Figure 6-4 and Figure 6-5 show the Grid 2 circuits.

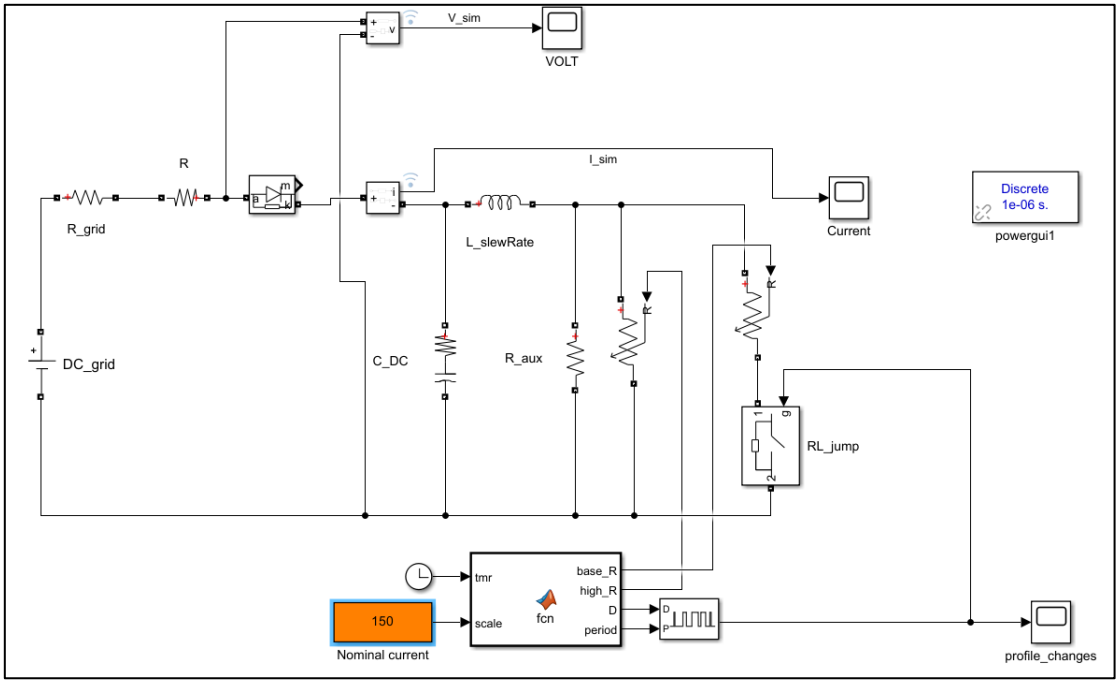


Figure 6-4 - Passive load model with Grid 2 (V + R + Diode)

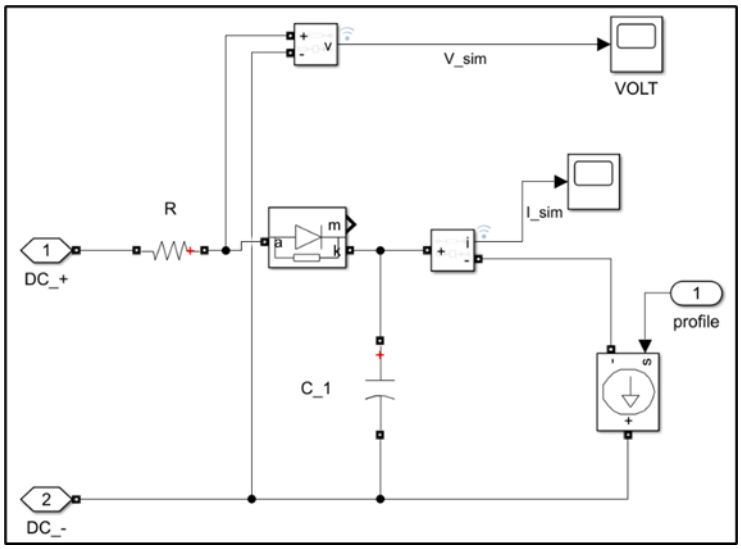


Figure 6-5 - Active load model (subsystem) with Grid 2 (V + R + Diode)

6.2.3. Grid 3: controlled voltage source

Grid 3 uses a controlled voltage source held at a constant 800 V reference. This imitates the regulated output of an AC/DC active front-end converter: instead of letting the bus voltage drop with current, the source maintains its terminal voltage regardless of how much current is drawn. This is the most accurate representation of the converter behaviour among the three grids, as it captures the voltage regulation that an active front-end converter provides in a real data centre supply path.

The practical consequence is that Grid 3 produces no static voltage drop and no dynamic droop under transient loading. While Grids 1 and 2 allow the bus voltage to sag by up to a few volts during high-current pulses, Grid 3 holds the voltage at exactly 800 V. This means the passive model, whose current is governed by I = V/R, will draw slightly different current under Grid 3 than under Grids 1 and 2. For the active model this distinction is irrelevant since the controlled current source is voltage-independent by design. This asymmetry is one of the reasons for testing all three grids rather than relying on a single supply model. Figure 6-6 and Figure 6-7 show the Grid 3 implementations.

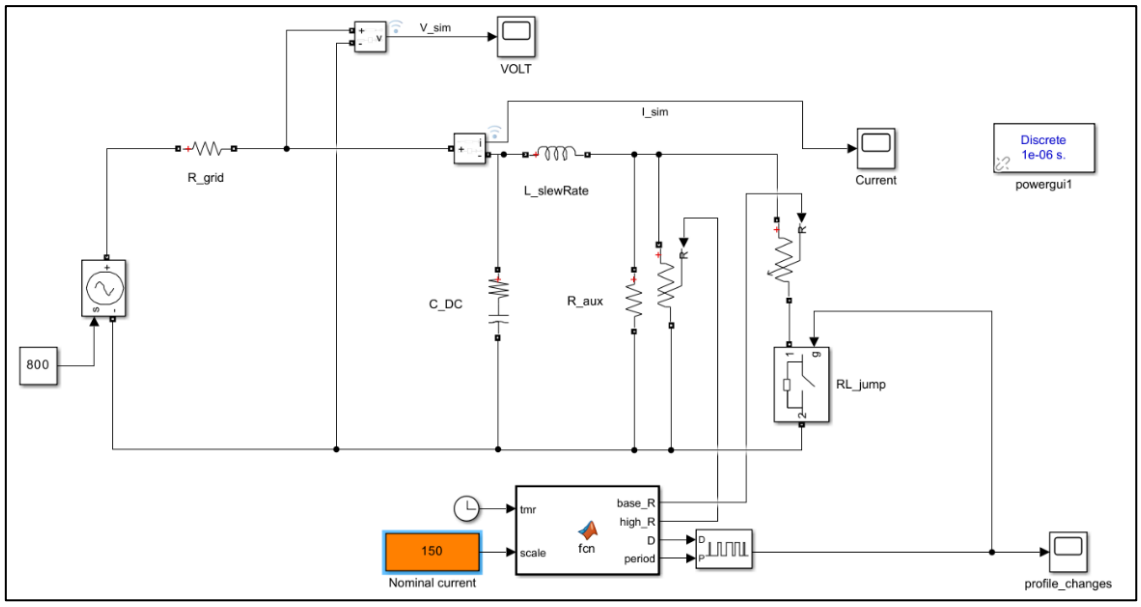


Figure 6-6 - Passive load model with Grid 3 (controlled voltage source)

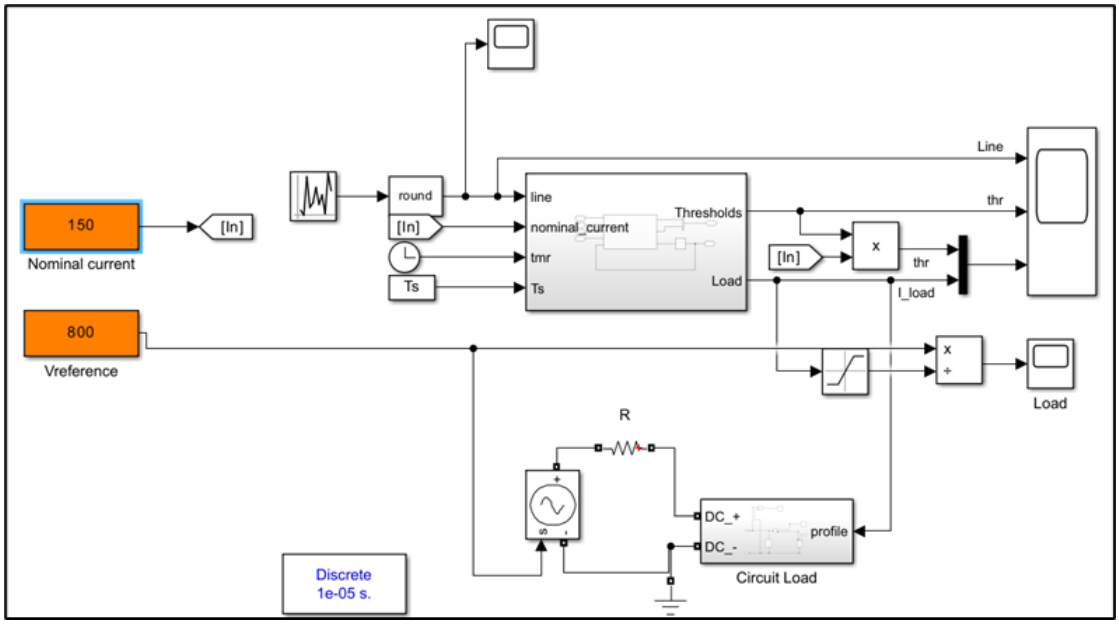


Figure 6-7 - Active load model with Grid 3, full system view including the controlled voltage source and the load profile selector

7 Validation Methodology

The previous chapters presented the passive and active models individually and introduced three refrence grid topologies tuned with the help of simulation data files provided for this project. This cahpter brings together these components in order to structure an accurate comparison. The main ideea is: given the same grid, the same nominal current and the same set of OCP Table 8 pulse load profiles, which model type, passive or active, reproduces the specification most faithfully. Therefore, this chapter describes how the simulations were set up, why specific load scenarios were chosen, and why a metric-based comparison strategy was adopted instead of a direct time-domain R^2. The coefficient of determination R^2 used throughout this chapter is defined as:

$$\boldsymbol{R^2 = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(y_i - \bar{y})^2}} \qquad (7.1)$$

where y_i are the reference values (OCP targets), $\hat{y}_i$ are the simulated values, and $\bar{y}$ is the mean of the reference vector [20].

Just as mentioned during the design chapters of the models, the way they are built differs. The active model, by construction, drives exactly the current commanded by the FCN block regardless of what the bus voltage is doing. The passive model, also by construction, draws a current that is determined by the instantaneous ratio of bus voltage to circuit resistance, filtered through the RL time constant of L_slewRate. These are fundamentally different physical mechanisms, and their performance depends on which aspect of the OCP specification is being evaluated: amplitude accuracy, waveform shape, or bus voltage behaviour.

The comparison is divided into three parts, each answering a different but complementary question. The first part checks OCP profile amplitude compliance at the rated operating point. The second part tests whether passive accuracy holds across a broader range of operating points, from 25% to 200% of rated load. The last part examines bus voltage behaviour, which the current-level metrics alone cannot capture. Together, these provide a complete picture of where each model excels and where it falls short.

7.1. Mapping load percentages to the nominal current

The six measurement files provided represent the full data centre system running at 25%, 50%, 75%, 100%, 150%, and 200% of a rack nominal current. From inspection of the Load100 file, the rack nominal is approximately 1050 A, which is consistent with a rack of 800 V GPU server modules at the scale described in the OCP Diablo 400 specification. The designed passive and active models operate at a configurable nominal_current parameter set to 150 A, giving a ratio of seven between the two systems.

This scale difference is handled through per-unit normalisation: all current values are divided by their respective nominal before any comparison. The 800 V bus reference is shared by both systems, so voltage can be compared directly. Under per-unit normalisation, the waveform shapes are identical at matching load fractions regardless of the absolute scale. Table 7-1 shows the simulated system data from the files: the correspondence between the load percentages, local nominal_current settings, and the measured terminal voltage variation at each operating point.

Table 7-1 - load percentages, nominal_current values and measured bus voltage variation in each file

Measurements files	Load fraction	nominal_current (local)	REAL V_var (measured)
Load25.mat	25%	37.5 A	≈ 2.9%
Load50.mat	50%	75 A	≈ 5.7%
Load75.mat	75%	112.5 A	≈ 10.2%
Load100.mat	100%	150 A	≈ 11.4%
Load150.mat	150%	225 A	≈ 27.7%
Load200.mat	200%	300 A	≈ 39.9%

In the simulated supply chain, the voltage variation is non-linear due to AC/DC converter dynamics. However, in the simplified grids that are build for the comparison of the passive and active models there is nothing like that. The measurement files are only used for grid parameter tuning and not for the actual verification of the models so there is no use for these files furtherly.

7.2. Nominal current choice for OCP compliance testing

OCP Table 8 defines all eight pulse load profiles as percentages of the PSU's rated nominal current. The profiles themselves do not specify an absolute current level, but they specify fractions. For example, profile 5 operates between 20% (low state) and 140% (high state) of whatever nominal the PSU is rated for. From these load % files, Load75 (112.5 A nominal) and Load100 (150 A nominal) sit cleanly within the OCP

operating band. The principal OCP comparison is chosen to be conducted at 150 A nominal (Load100).

Testing at a different nominal, like 112.5 A (the 75% load point), would produce waveforms with different absolute amplitudes but identical per-unit shapes. Both models scale their current tresholds and resistance proportionally to the nominal current through the FCN scale parameter: high_frac × nominal_current and low_frac × nominal_current regardless of which profile is active, whereas passive scales its resistances as 150/scale. The RL time constant τ = L_slewRate / R_load is also invariant, since R_load is proportional to V_bus / (high_frac × nominal), which scales with nominal in the denominator and cancels it. The resulting per-unit waveform shape is therefore identical at any nominal current: for example, profile 5 produces 1.40 pu high and 0.20 pu low whether the nominal is 75 A, 150 A, or 300 A, producing identical percentage error tables and R^2 values, adding no new information. Testing all eight profiles at a single nominal current of 150 A is therefore not a simplification, but it is the interpretation of the OCP specification, which defines profiles as fractions of a configurable rated current rather than as fixed absolute current levels.

It should be noted that this per-unit invariance holds exactly for the active model, where the FCN directly commands the current fractions, and approximately for the passive model, where the achieved current is slightly reduced by the R_grid voltage drop at the load terminals.The load-range sensitivity, which is the one quantity that does change with nominal current, due to the voltage drop across R_grid, is addressed separately in Chapter 7.7. through a dedicated R^2_B analysis across all six load levels.

A related methodological choice is the use of per-unit current normalisation rather than absolute amperes for all comparisons. If absolute currents were used, the R^2 computed across the slow profiles would be dominated by the large spread in absolute I_high values (from 0.2 × 150 = 30 A for I_low of profile 5 to 1.75 × 150 = 262.5 A for I_high of profile 3), and a model that simply scales linearly with nominal would appear to have $R^2 = 1$ even if it systematically over- or undershoots the target fractions. In per-unit, the reference vector contains the actual OCP fractions [1.75, 1.40, 1.40, 1.40, 0.40, 0.20, 0.60, 0.30] and the simulated values are divided by the same nominal current, so R^2 measures whether the model hits the correct fraction at each profile, not just whether the values are large where the target is large. What isconsidered is that the OCP Table 8 specifies fractions, not absolute currents, and the per-unit comparison respects that specification directly.

A related question is whether different OCP profiles should be tested at different nominal currents, guided by the "average load" column in OCP Table 8: for example, testing profile 5 at 80% nominal and profile 1 at 100% nominal. This is not necessary. The "average load" percentage is a duty-cycle-weighted descriptor of each profile's mean power consumption relative to nominal. It is computed as (I_high × t_high +

I_low × t_low) / (I_nom × T2), it being not a specification of the nominal current at which the profile should be exercised.

7.3. Simulation setup

Each passive model simulation runs for 5 seconds, during which the FCN cycles through eight frames following a specific time sequence: profile 2 from t = 0 to 0.5 s, profile 1 from 0.5 to 1 s, profile 5 from 1 to 2 s, profile 7 from 2 to 2.5 s, profile 3 from 2.5 to 3 s, profile 4 from 3 to 3.5 s, profile 6 from 3.5 to 4 s, and profile 1 again from 4 to 5 s. This gives seven unique profiles within the window, while profile 8 is never visited in this 5-second sequence because of how the frame timer was structured. It is classified as SLOW (ΔI_achievable = 400 A >> ΔI_required = 217.5 A), meaning it would reach its thresholds if given a frame, but it is absent from the sequence and therefore excluded from both model comparisons. More context is provided in Chapter 7.4.

For the active model, the Band-Limited White Noise block driving the random profile selector uses a fixed input, making its sequence repeatable across runs. Giving the design of the passive model's frame boundaries, the active and passive models visit the same OCP profile in the same time window. So, they allign: both models are always running profile 1 during the frame [0.60, 0.95] s, profile 5 during [1.20, 1.85] s, and so on, making the selection for comparisons much clearer.

Therefore, every model, passive and active, is run at every nominal current for each of the three grid models, resulting into 3x6=18 files for each passive and active. These are saved as active_gridx_Ly and passive_gridx_Ly, where x is the grid variation 1, 2 or 3 and y is the load value 25 to 200, as in the files given. As mentioned before, only L100 is used in comparison, therefore active_gridx_L100 and passive_gridx_L100 are the 6 selected files.

7.4. Profile classification and special cases

Before computing any accuracy metrics, the eight OCP profiles must be classified according to whether their current swing is physically reachable within the available pulse time at the specified slew rate. Just like in Table 2-2, an updated version, with p.u. values is seen in Table 7-2 considering the following: a profile is classified as slow if the achievable current change at the PSU slew rate covers the full swing within t_high, otherwise it is fast (slew-rate-limited).

$$\Delta I_achievable = sr \times t_high \quad vs \quad \Delta I_required = (I_high - I_low) \times I_nom$$

Table 7-2 - Profile classification at 150 A nominal

Profile	I_high (pu)	I_low (pu)	t_high (ms)	sr (A/μs)	ΔI_req (A)	ΔI_ach (A)	Class
1	1.30	0.70	0.05	1	90	50	FAST
2	1.40	0.20	0.10	1	180	100	FAST
3	1.75	0.40	2.00	1	202.5	2000	SLOW
4	0.80	0.00	1.00	0.1	120	100	FAST
5	1.40	0.20	50.0	1	180	50000	SLOW
6	1.40	0.60	50.0	1	120	50000	SLOW
7	1.40	0.30	50.0	1	165	50000	SLOW
8	1.75	0.30	0.40	1	217.5	400	SLOW

Three profiles require special handling beyond the slow/fast classification.

The passive model follows a fixed frame sequence in which profile 2 occupies frame 1 (t = 0–0.5 s). During this same window, the active model's Band-Limited White Noise block, which is driving the random profile selector, had not yet settled at the beginning of the simulation onto a stable profile, producing an unsettled transitional waveform from which no reliable I_high or I_low values can be extracted. Active profile 2 values are therefore noted as NaN and excluded from the fast-profile R^2 computation

Profile 4 has I_low = 0 A, which is fundamentally incompatible with the passive model topology. The FCN base resistance formula $R_{base} = (R_{I_high} \times R_{I_low})/(R_{I_low} - R_{I_high})$ diverges because $R_{I_low} \to \infty$ as $I_{low} \to 0$, producing a circuit that cannot achieve 0 A in the low state. This is a known limitation of the switched-resistance passive topology and is documented in Chapter 4. The active model handles it correctly because the FCN simply sets the output gain to zero.

Profile 8 is classified as SLOW ($\Delta I_{achievable} = 400$ A >> $\Delta I_{required} = 217.5$ A at 1 A/μs) but is never visited in the passive model's 5-second frame sequence while also the active model's random selector never visited it either. It is therefore absent from both passive and active extraction and marked as NaN in the results. Profile 8 is excluded from both model's R^2 computations, so neither distorts the comparison. However, based on the slow-profile classification ($\Delta I_{achievable} = 400$ A far exceeds $\Delta I_{required} = 217.5$ A, giving a $T1/\tau$ ratio well above 1) the RL circuit is expected to settle fully within each pulse and reproduce both thresholds accurately, consistent with the behaviour observed for profiles 3, 5, 6, and 7. This physical reasoning gives confidence that both models would pass the OCP amplitude test for profile 8, deven though the limited 5-seconds run did not provide this profile.

7.5. Data extraction and comparison metrics

7.5.1. Extraction method

Current levels are extracted from time windows within each frame, specifically the intervals [t_0 + 0.05 s, t_1 − 0.05 s], in order to exclude transient settling at frame boundary. For slow profiles, where the current reaches and holds a flat plateau (a trapezodial waveform), the maximum of the extracted current segment is used for I_high and the minimum for I_low. For fast profiles, where the current rises and falls continuously without settling (a triangular waveform), the 90th and 10th percentiles are used instead.

The passive model's inductance L_slewRate produces a settling transient at the start of each frame as the circuit adjusts to the new resistance. For most frame transitions this is brief and the current reaches its target level quickly. The most significant exception is the transition at t = 2.5 s, where the inductor carries stored energy from frame 4's idle current and produces a visible overshoot spike before settling, as documented in Chapter 4.3 The 90th percentile extraction represents the current level sustained for 90% of the extraction window, corresponding to the settled pulse amplitude rather than the initial transient, and is the chosen quantity to compare against the OCP target. The same extraction is applied to the active model for consistency, even though the active model produces no settling transients.

7.5.2. R^2 computation

The coefficient of determination R^2 (Equation (7.1)) is computed separately for slow and fast profile groups.

For slow profiles the reference is the OCP Table 8 target, by using the I_high and I_low values that the specification requires the PSU to encounter. For fast profiles, the reference is also the OCP Table 8 target. As established in Chapter 2.2.6, fast-profile frames enter from higher-current previous profiles, so the oscillation anchors at the top: each rise is clamped or arrested at I_high and settles at I_high minus the available fall swing. Both models can't meet the OCP I_low target, just like it is detailed in Chapters 2 and 4, they have an available half-cycle swing is sr × t_low / I_nom = 0.333 pu for profile 1, for example, which is less than the required OCP swing of 0.600 pu, with the current never reaching I_low = 0.700 from the top-anchored position. Therefore, neither the passive nor the active model can reach the OCP I_low target within the available fall time, so R^2 will not reach 1.0000 for fast profiles, but comparing against OCP directly shows answers whether the simulation reproduces the current stress that OCP Table 8 specifies the PSU must encounter.

A direct point-by-point time-domain R^2 between the passive and active waveforms was investigated and discarded. Both models visit the same profile in the same frame window, however, the passive model settles over a timescale governed by its RL time constant, while the active model transitions at the FCN's commanded slew rate. Every frame boundary therefore creates a transient in the passive waveform that has no counterpart at the same time in the active waveform. When R^2 is computed across thousands of time samples that include these boundary transients, the comparisons are dominated by the mismatched transitions rather than by the settled current levels. This produces strongly negative R^2 values: the results showed means of "−0.03" for slow frames and "−3.27" for fast frames, which are a consequence of the timing mismatch, not of any model inaccuracy, as it is known that slow profiles hit their targets. The scalar extraction of I_high and I_low is made through getting a single representative value for each window, ignoring transition dynamics.

7.5.3. R^2_B load range metric

The OCP compliance analysis in Chapter 7.5.2 was conducted at 150 A nominal, which is the rated operating point. In practice, a PSU may be tested at partial load during commissioning or overloaded during qualification stress testing. A complementary metric, R^2_B, is defined to test whether passive model accuracy holds across the full operating range from 25% to 200% of rated nominal. For each load level, the passive I_high values across all slow profiles are compared against the theoretical active target (high_frac × nominal), and R^2 is computed across this set. The theoretical active target is used because the controlled current source reaches it exactly by construction for slow profiles. Six separate passive simulations are run per grid at nominal currents of 37.5 A, 75 A, 112.5 A, 150 A, 225 A, and 300 A.

R^2_B therefore answers a different question from the primary OCP comparison: not whether both models hit the OCP target at rated conditions, but whether the passive model's accuracy degrades as operating conditions move away from the design point.

7.6. Results

This chapter reports the actual numerical results and draws conclusions about which model type is more appropriate and in what context. The results address both verification and validation: for the active model, confirming that it correctly implements the OCP specification as intended, while for the passive model, confirming whether a physical RL circuit can reproduce those same specifications empirically.

7.6.1. Slow profiles R^2

Table 7-3 shows the extracted current levels in per unit for slow profiles at 150 A nominal across all three grids, together with the OCP Table 8 targets. For the active model, the slow-profile R^2 = 1.0000 is a verification result, confirming correct implementation, whereas for the passive model, the same result is a validation finding.

Both models achieve R^2 = 1.0000 against OCP targets for the slow profiles across all three grids. The active model hits the target to four decimal places for every profile, which is expected, because the controlled current source ramps at 1 A/µs, reaches the FCN threshold, and holds it. The passive model deviates by at most 0.3% from the OCP target, with Grid 2 showing the largest deviation due to the additional diode forward voltage reducing the available bus voltage at the load terminals. Grid 3 shows zero deviation because its controlled source holds the terminal voltage constant and no resistive drop appears at the measurement point.

Table 7-3 - Extracted current levels vs OCP targets for slow profiles at 150 A nominal

	OCP Target		***Active (all grids identical)***						***Passive***					
Profile	**Ref H**	**Ref L**	**G1 H**	**G1 L**	**G2 H**	**G2 L**	**G3 H**	**G3 L**	**G1 H**	**G1 L**	**G2 H**	**G2 L**	**G3 H**	**G3 L**
P3 222 Hz	1.750	0.400	1.750	0.400	1.750	0.400	1.750	0.400	1.748	0.400	1.745	0.399	1.750	0.400
P5 10 Hz	1.400	0.200	1.400	0.200	1.400	0.200	1.400	0.200	1.399	0.200	1.397	0.200	1.400	0.200
P6 10 Hz	1.400	0.600	1.400	0.600	1.400	0.600	1.400	0.600	1.399	0.600	1.397	0.599	1.400	0.600
P7 12.7 Hz	1.400	0.300	1.400	0.300	1.400	0.300	1.400	0.300	1.399	0.300	1.397	0.300	1.400	0.300
P8 1.2 kHz	1.750	0.300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
R^2 vs OCP target			**1.0000 (all grids)**						**1.0000 (G1) / 1.0000 (G2) / 1.0000 (G3)**					

The small undershoot in the passive model is physically correct. At I_high = 262.5 A (profile 3 at 150 A nominal), the drop across R_grid = 0.004 Ω is 1.05 V, reducing the effective bus voltage from 800 V to 798.95 V and proportionally reducing the passive model's steady-state current by 0.13%. This is not a modelling error: it is exactly the static droop a real resistive cable would impose. However, the passive model I_low values match the OCP targets exactly across all slow profiles and all grids. This is because the R_grid voltage drop is proportional to load current, which is smaller during the low phase than the high phase: at profile 5 with I_low = 0.20 × 150 = 30 A, the R_grid drop is only 0.004 × 30 = 0.12 V, producing a 0.015% undershoot. The R^2 = 1.000 confirms that these deviations are small enough not to register at four decimal places of precision.

So, the comparison between passive and active on slow profiles is, in practical terms, a draw on OCP amplitude compliance.

7.6.2. Fast profiles R^2

Table 7-4 shows the extracted I_high and I_low values for all three fast profiles across both models and all three grids, compared against the OCP Table 8 targets. Just as

mention in Chapter 7.4, profile 2 for the active model is not included, seeing NaN instead of values. The analysis is detailed in terms of each profile.

Table 7-4 - Extracted fast-profile vs OCP target current levels at 150 A nominal

	OCP Target		*Active*						*Passive*					
Profile	**OCP H**	**OCP L**	**G1 H**	**G1 L**	**G2 H**	**G2 L**	**G3 H**	**G3 L**	**G1 H**	**G1 L**	**G2 H**	**G2 L**	**G3 H**	**G3 L**
P1 10 kHz	1.300	0.700	1.300	0.967	1.300	0.967	1.300	0.967	1.375	1.122	1.372	1.120	1.376	1.122
P2 5 kHz	1.400	0.200	NaN	NaN	NaN	NaN	NaN	NaN	0.674	0.226	0.674	0.226	0.674	0.226
P4 500 Hz (0.1A/µs)	0.800	0.000	0.800	0.127	0.800	0.127	0.800	0.127	0.799	0.007	0.798	0.007	0.799	0.007
R^2 vs achievable			**0.8987 (all grids)**						**0.7864 (G1) / 0.7886 (G2) / 0.7858 (G3)**					

For profile 1 (10 kHz), the active model achieves I_high = 1.300 I_low = 0.967 and across all three grids. As established in Chapter, profile 1 is slew-limited: within t_low = 0.05 ms at 1 A/µs, the available swing per half-cycle is sr x t_high =50 A=0.333 pu, less than the required 0.600 pu, so the waveform is a triangle that never reaches the OCP I_low target. In steady state, the oscillation is top anchored, due to FCN threshold clamping architecture that allows the integrator to reach and hold the programmed at high threshold of I_high=1.300, while the fall of 0.33 pu lands at 0.967, above the OCP I_low target of 0.700. The OCP I_low target is therefore not met by the active model on this profile, given the slew rate and pulse duartion.

The passive model gives I_high = 1.372-1.376 across grids (slightly above the active value due to inductive overshoot at the transition) and I_low = 1.120-1.122, which is significantly above the OCP target, confirming that the RL circuit does not return fully to I_low within the short t_low = 0.05 ms before the next pulse begins. The RL circuit is also top-anchored it enters from a higher-current profile, and inductive inertia carries the current above the OCP I_high target before it decays (+0.075 pu overshoot on I_high, while I_low settles even further above the OCP target than the active model (+0.422 pu vs OCP I_low).

For profile 4 (500 Hz, sr = 0.1 A/µs), both models achieve I_high values that meet the OCP target to within 0.1%: active gives 0.800 (exact), passive gives 0.799. Profile 4 uses a ten-times-slower slew rate (0.1 A/µs), and with t_high = 1 ms the available rise is 0.667 pu from a bottom-anchored starting point, but the top-anchored steady state pins the ceiling at OCP I_high = 0.800 regardless. The active I_low = 0.127 falls short of the OCP target of 0.000: fall = 0.1 A/µs × 1 ms / 150 = 0.667 pu from 0.800, landing at 0.133, so the small difference between 0.133 analytical and 0.127 extracted is attributed to the R_grid voltage drop at peak current. The passive I_low ≈ 0.007 confirms the known topology limitation documented in Chapter 7.4: the passive model cannot achieve I_low = 0% (0 A) because the FCN base resistance formula produces a finite parallel resistance even at the 0 A specification, resulting in a small residual current.

Profile 2 (5 kHz) has no active values (NaN) because the active model's random selector had not yet settled onto a stable profile during frame 1, producing a transitional waveform from which no reliable values can be extracted. The passive values (I_high = 0.674, I_low = 0.226) are consistent across all three grids, confirming

that the passive model's behaviour at 5 kHz is grid-independent: the RL time constant dominates and the small differences in grid impedance are negligible at this frequency.

Table 7-5 - R^2 of active and passive models against OCP targets for fast

Model (fast, P1+P4)	Grid 1 R^2	Grid 2 R^2	Grid 3 R^2	Grid sensitivity
Active vs OCP targets	0.8987	0.8987	0.8987	None, it is grid-independent
Passive vs OCP targets	0.7864	0.7886	0.7858	Small due to RL time constant varying

Table 7-5 shows the extracted current levels, per grid, per model, as a means of the profile comparison values. The reported fast-profile R^2 values (Active: 0.899, Passive: 0.786-0.789) are computed from profiles 1 and 4 only, with profile 2 at 5 kHz, expecting a Passive (fast) vs OCP targets of a lower R^2=0.55 value had it been included. The active model would still govern over passive though.

The active model achieves R^2 = 0.899 against the OCP target, which is identical across all three grids. The controlled current source in the Simulink implementation is voltage-independent by construction (Chapter 5.1): adding a series diode (Grid 2) or switching to a regulated AC source (Grid 3) has no effect on the commanded current ramp. The “grid independence” term needs explanation in detail, as across the three representative supply topologies tested, the active model showed no grid sensitivity. More complex supply models, like those including bus capacitance dynamics or converter control loops are beyond the scope of this work..

The passive model achieves R^2 = 0.786-0.789, which is lower than the active model and shows small variation across grids (±0.003). The lower R^2 reflects a clear difference in the physical mechanism: the passive model's current follows an exponential rise governed by the RL time constant τ = L_slewRate / R_total, where R_total is the sum of R_grid, R_aux, and the switched load resistance set by the FCN (Chapter 4.2). For fast profiles where $\tau \geq$ t_high, the exponential has not decayed to its steady-state value by the end of the high phase, and the inductor carries current above the OCP I_high target before settling, producing the +0.075 pu overshoot on profile 1 visible in Table 7-4. On the low side, the same time constant limits the fall, leaving I_low well above the OCP target. The variation with each grid of R^2 happens because R_grid in series with the switched load resistance makes the effective time constant slightly different between Grid 1 (pure R_grid) and Grid 2 (R_grid plus diode), shifting the exponential rise by a small amount and producing the ±0.003 spread in R^2 values.

The active model achieves a higher R^2 because I_high values match the OCP target exactly for both profiles, due to the FCN clamping designed for this, while I_low deviations are driven only by the slew-rate-limited fall (Chapter 2.2.7). The passive model has a lower R^2 for two separate reasons: the RL inductor causes I_high to overshoot the OCP target slightly (rather than landing on it), and the RL fall is slower than the FCN ramp, leaving I_low further above the OCP target than the active model (Chapter 4). Both R^2 values are below 1.000 because neither model can reach the OCP I_low specification within the available pulse duration, a physical constraint

established analytically previously, not a model failure. Therefore, the fast-profile R^2 values confirm that the active model more consistently reaches the achievable target than the passive model against the same reference.

7.7. Load-Range robustness R^2_B

The primary OCP comparison was conducted at 150 A nominal. To confirm that the passive model's accuracy does not degrade meaningfully across a broader operating range, a supplementary R^2_B analysis was performed at six nominal current levels from 37.5 A (25% load) to 300 A (200% overload), using the eighteen passive simulation files described in Chapter 7.3. For each load level, R^2_B compares the passive I_high values across all slow profiles against the theoretical active target (high_frac × nominal), and R^2 is computed across this set. Table 7-6 shows the results.

R^2_B remains above 0.999 for all grids from 25% to 100% load and above 0.995 at 150% load, confirming that the per-unit invariance argument holds throughout the OCP operating range. Only at 200% overload is when R^2_B drops to 0.987–0.997, driven by the increased R_grid voltage drop at 300 A nominal (~2.1 V for Grid 1), which reduces the effective bus voltage to 797.9 V and causes the passive model to slightly undershoot the FCN target.

Table 7-6 - R^2_B: passive model slow-profile accuracy vs theoretical active target across six load levels for the 3 grids

Nominal current	Load %	Grid 1 R^2_B	Grid 2 R^2_B	Grid 3 R^2_B
37.5 A	25%	1.0000	0.9998	1.0000
75 A	50%	1.0000	0.9997	1.0000
112.5 A	75%	0.9999	0.9996	1.0000
150 A	100%	0.9999	0.9994	1.0000
225 A	150%	0.9995	0.9987	0.9999
300 A	200%	0.9970	0.9953	0.9985

The active model is immune to this effect by construction. This degradation confirms that the passive model becomes a bit less accurate as operating conditions move significantly beyond the OCP rated range. The effect is small: R^2_B remains above 0.987 even at 200% overload this being caused by a physical phenomenon that can happen in real life: the R_grid voltage drop reduces the effective bus voltage at high current, pulling the passive models achieved current slightly below the FCN target.

7.8. Voltage behaviour

The voltage at the model's terminals determines the supply conditions the PSU experiences during a test, and a model that produces no voltage transients will not

stress the PSU's regulation loop in the same way that a real GPU workload would. Table 7-7 shows the voltage deviation metrics for passive and active at 150 A nominal.

Grid 3 is excluded from the voltage comparison because the voltage sensor in that model is placed at the controlled source terminal rather than at the load terminals, meaning the measurement does not capture the load-side voltage behaviour and would read a constant 800 V regardless of the load current.

Table 7-7 - Voltage deviation from 800 V nominal at 150 A nominal. *Grid 3 voltage sensor is at the AC source terminal

Grid	Passive MaxΔV	Active MaxΔV	Passive V_std	OCP limit ±24 V	OCP status
Grid 1 (R only)	1.041 V	1.306 V	0.343 V	±24 V	PASS (both)
Grid 2 (R + diode)	1.039 V	1.306 V	0.342 V	±24 V	PASS (both)
Grid 3 (AC source)*	0.003 V	excl.	0.000 V	±24 V	EXCL

Both passive grids pass the OCP ±24 V limit by a wide margin. The passive model's maximum deviation of 1.041 V (Grid 1) and 1.039 V (Grid 2) corresponds almost exactly to the R_grid drop at the highest current in the simulation: at I_high of profile 3 (262.5 A), the drop across R_grid = 0.004 Ω is 1.05 V. The active model's maximum deviation of 1.306 V is slightly larger despite the current source being voltage-independent. This is because the active model produces instantaneous current steps at each FCN threshold crossing, whereas the passive model's current ramps through L_slewRate. An instantaneous step across R_grid produces a momentary voltage drop dV = R × dI that is larger than the rate of change dI/dt, even though the settled drop is the same. Both models remain well within the OCP ±24 V limit.

7.9. Summary and recommendation

Table 7-8 brings together the key metrics from all four analysis layers. Across all metrics, the passive-vs-active comparison outcome is consistent across Grid 1, Grid 2, and Grid 3, which are stiff enough at this power level that neither model's behaviour is altered by any grid topology attached.

Table 7-8 - Summary of all comparison metrics

Metric	Active model	Passive model	Winner
R^2 vs OCP — slow profiles (all grids)	1.0000	1.0000	Tie
R^2 vs OCP — fast profiles (Grid 1)	0.8987	0.7864	Active
R^2 vs OCP — fast profiles (Grid 2)	0.8987	0.7886	Active
R^2 vs OCP — fast profiles (Grid 3)	0.8987	0.7858	Active
R^2_B at 25%–150% load	1.000 (th.)	≥0.995	Tie
R^2_B at 200% overload	1.000 (th.)	0.995–0.997	Active
Max voltage deviation (G1)	1.306 V (PASS)	1.041 V (PASS)	Passive*
Real inductive bus transients	None	Present	Passive*

The passive model is good when the goal is to expose the PSU to physically realistic electrical conditions. Its RL-based current response produces realistic voltage transients at profile transitions, that pushes the limits of the PSU's control loop and captures the coupling between bus voltage and load current that exists in real hardware. On slow profiles, which represent the majority of the OCP Table 8 test time, it matches the active model's performance exactly (R^2 = 1.000, <0.3% error). The inability to complete the full OCP current swing on fast profiles is a result of physical RL dynamics rather than a flaw in the model. For PSU qualification testing where the voltage response under dynamic loading must be characterized, the passive model provides an honest representation of real operating conditions.

The active model is the better option for precisely replicating the OCP current levels, as the FCN clamping architecture locks I_high exactly at the OCP threshold regardless of entry conditions, and its I_low deviations reflect only the slew-rate constraint that the OCP specification itself imposes, without the additional RL time constant effect that further limits the passive model. It achieves R^2 = 0.8987 against the OCP fast-profile target, compared to 0.786–0.789 for passive, and handles all eight OCP profiles correctly including profile 4. For OCP certification testing, where reproducing the load that reliably delivers the specified current levels is required, the active model is the natural choice.

8 Conclusion and Future Developments

8.1. Main conclusions

This thesis set out to answer a practical question: how should the OCP Diablo 400 Table 8 pulse profiles be reproduced in simulation for PSU qualification testing on an 800 V DC bus, and which modelling approach is more appropriate? Two implementations are designed and compared: a signal-flow active model and a physical passive model, across three reference supply grids and analysis layers.

The data gives a clear answer: the active model is the recommended tool for OCP Table 8 PSU qualification testing. It reproduces the specified current profiles most accurately at the rated 150 A nominal across all three grids, and its per-unit scaling helps with the OCP targets analysis. The R^2 against OCP targets is 1.000 on slow profiles and 0.8987 on fast profiles, meeting the OCP I_high specification target for profiles 1 and 4 via its threshold-clamping architecture, while I_low is slew-limited. It handles profile 4 correctly, visits seven of the eight OCP profiles reliably within the 5-second simulation window (profile 8 is absent from both models) and is completely insensitive to supply topology, the very properties that a qualification tool requires.

The passive model is not without value. The physically correct RL dynamics produce inductive transients at profile boundaries, voltage-current coupling, and exponential settling behaviour that the active model cannot replicate by construction. These properties make it a useful tool when the question is about physical fidelity, basically understanding how a real circuit behaves under these profiles rather than verifying that the profiles themselves are correctly generated. However, for the primary purpose of this thesis, its lower fast-profile accuracy (R^2=0.786-0.789) and topology-dependent behaviour make it the secondary choice.

Chapter 5 showed qualitatively that the active model commands while the passive model responds, while Chapter 7 confirmed this with numbers, consistently across metrics and all three grids.

8.2. Future developments

For future work, there are some limitations identified in this thesis. First, the passive model's fixed inductance forces a trade-off between fast and slow profile accuracy that could not be resolved within the scope of this work. Improving its controllability, particularly for the fast slew-rate-limited profiles, remains an open design problem. Secondly, the three reference grids used here are simplified Thévenin equivalents, so, more complex supply models including converter control loops could reveal voltage-dependent behaviour not captured in this work. Finally, the random profile selector in the active model cycles through the eight OCP profiles without following any statistically representative sequence. Real GPU workloads transition between states in a structured way, driven by training phases, checkpointing events, and idle periods, rather than randomly [1]. A fixed representative sequence based on a realistic LLM workload run would make the simulation more faithful to actual PSU operating conditions.

Bibliography

[1] J. Huntington and M. Tu, "800 VDC Architecture for Next-Generation AI Infrastructure," NVIDIA Corporation, 2024.

[2] Open Compute Project, "Diablo 400 Project: Rack and Power – Base Specification, Version 0.5.2," Open Compute Project, 2025.

[3] R. Rahmani, I. Moser and M. Seyedmahmoudian, "A Complete Model for Modular Simulation of Data Centre Power Load," *IEEE Transactions on Automation Science and Engineering,* vol. 14, no. 8, 2017.

[4] G. E. Colón Reyes, R. Dye, C. Tomlin and D. Callaway, "Effects of Dynamic Power Electronic Load Models on Power Systems Analysis Using ZIP-E Loads," *arXiv preprint arXiv:2407.12715,* 2024.

[5] The MathWorks, Inc., "MATLAB/Simulink, version R2023b," 2023. [Online]. Available: https://www.mathworks.com/products/simulink.html.

[6] International Energy Agency, "Energy and AI," IEA, Paris, France, 2025.

[7] A. Pratt, P. Kumar and T. V. Aldridge, "Evaluation of 400V DC distribution in telco and data centers to improve energy efficiency," in *IEEE International Telecommunications Energy Conference (INTELEC)*, 2007.

[8] G. Wawrzola, "Challenges of DC data center power distribution protection," in *International Conference on Development in Power System Protection (DPSP)*, Edinburgh, UK, 2016.

[9] ABB Ltd, "Redefining Power Infrastructure for AI: The Role of 800 VDC in Data Centers," ABB Ltd, Zurich, Switzerland, 2025.

[10] D. Robb, "Data Centers Are Transitioning From AC to DC," 24 March 2026. [Online]. Available: https://spectrum.ieee.org/data-center-dc.

[11] M. Blake, M. Hsu, I. Goldwasser, H. Petty and J. Huntington, "NVIDIA 800 VDC Architecture Will Power the Next Generation of AI Factories," 20 May 2025. [Online]. Available: https://developer.nvidia.com/blog/nvidia-800-v-hvdc-architecture-will-power-the-next-generation-of-ai-factories/.

[12] T. Dragičević, X. Lu, J. C. Vasquez and J. M. Guerrero, "DC Microgrids—Part I: A Review of Control Strategies and Stabilization Techniques," *IEEE Transactions on Power Electronics,* vol. 31, no. 7, pp. 4876-4891, July 2016.

[13] J. e. a. Xu, "Sequential Operating Simulation of Solid State Transformer-Driven Next-Generation 800 VDC Data Center," *arXiv preprint arXiv:2601.16502,* January 2025.

[14] J. V. Milanović, K. Yamashita, S. M. Villanueva, S. Z. Djokić and L. M. Korunović, "International Industry Practice on Power System Load Modeling," *IEEE Transactions on Power Systems,* vol. 28, no. 3, pp. 3038-3046, August 2013.

[15] IEEE Task Force on Load Representation for Dynamic Performance, "Load Representation for Dynamic Performance Analysis," *IEEE Transactions on Power Systems,* vol. 8, no. 2, pp. 472-482, May 1993.

[16] S. Lu, C. Xiao and Y. Weng, "Dynamic Load Model for Data Centers with Pattern-Consistent Calibration," *arXiv preprint arXiv:2602.07859,* February 2025.

[17] The MathWorks, Inc., "Simscape Electrical, version R2023b," 2023. [Online]. Available: https://www.mathworks.com/products/simscape-electrical.html.

[18] C. K. Alexander and M. N. O. Sadiku, Fundamentals of Electric Circuits, 7th ed., New York: McGraw-Hill, 2021.

[19] R. W. Erickson and D. Maksimović, Fundamentals of Power Electronics, 3rd ed., Cham: Springer, 2020.

[20] N. R. Draper and H. Smith, Applied Regression Analysis, 3rd ed., New York: Wiley, 1998.

A Appendix A

A.1. OCP Compliance Analysis Script

```
%% OCP Compliance Analysis — per-profile, per-grid, slew-rate-aware
%% Nominal current: 150 A
%% OCP Table 8 targets (per-unit) and FCN timing/slew parameters
ocp_high  = [1.30, 1.40, 1.75, 0.80, 1.40, 1.40, 1.40, 1.75];
ocp_low   = [0.70, 0.20, 0.40, 0.00, 0.20, 0.60, 0.30, 0.30];
t_high_p  = [0.05e-3, 0.1e-3, 2e-3, 1e-3, 50e-3, 50e-3, 50e-3, 0.4e-3];
t_low_p   = [0.05e-3, 0.1e-3, 2.5e-3, 1e-3, 50e-3, 50e-3, 28.6e-3, 0.43e-3];
sr_p      = [1, 1, 1, 0.1, 1, 1, 1, 1] * 1e6;   % A/s
%% Classify profiles by slew limitation
I_nom    = 150;
slew_lim = false(1,8);
for p = 1:8
    dI_req     = (ocp_high(p) - ocp_low(p)) * I_nom;
    dI_achieve = sr_p(p) * t_high_p(p);
    slew_lim(p) = dI_achieve < dI_req;
end
%% Achievable targets — what the model can actually reach given slew rate and pulse time
ach_high = ocp_high;
ach_low  = ocp_low;
for p = 1:8
    if slew_lim(p)
        rise = sr_p(p) * t_high_p(p) / I_nom;
        fall = sr_p(p) * t_low_p(p)  / I_nom;
        ach_high(p) = min(ocp_high(p), ocp_low(p)  + rise);
        ach_low(p)  = max(ocp_low(p),  ocp_high(p) - fall);
    end
end
%% Passive frame windows (deterministic) and profile mapping
frame_windows = [0.10 0.45; 0.60 0.95; 1.20 1.85; 2.10 2.45;
                 2.60 2.95; 3.10 3.45; 3.60 3.95; 4.10 4.90];
p_frame_profile = [2, 1, 5, 7, 3, 4, 6, 1];
%% Input files for the three grid configurations
p_files = {'passive_grid1_L100.mat','passive_grid2_L100.mat','passive_grid3_L100.mat'};
a_files = {'active_grid1_L100.mat', 'active_grid2_L100.mat', 'active_grid3_L100.mat'};
```

```
%% R² helper (NaN-safe)
R2 = @(s,r) ( 1 - sum((s-r).^2, 'omitnan') / sum((r - mean(r,'omitnan')).^2, 'omitnan')
);
%% Storage
results = struct();
for g = 1:3
    sp = load(p_files{g});
    sa = load(a_files{g});
    P_high = NaN(1,8); P_low = NaN(1,8);
    A_high = NaN(1,8); A_low = NaN(1,8);
    A_seen = zeros(1,8);  % how many times each profile was visited
%% Passive — frame windows are fixed, profile mapping is deterministic
    for f = 1:8
        p     = p_frame_profile(f);
        idx   = sp.t_sim >= frame_windows(f,1) & sp.t_sim <= frame_windows(f,2);
        I_seg = sp.I_sim(idx) / I_nom;
        if slew_lim(p)
            P_high(p)  = prctile(I_seg,  90);   % percentile avoids settling-transient
contamination
            P_low(p)  = prctile(I_seg, 10);
        else
            P_high(p) = max(I_seg);
            P_low(p)  = min(I_seg);
        end
    end
 %% Active — match each frame to the closest OCP profile (use achievable targets)
    for f = 1:8
        idx    = sa.t_sim >= frame_windows(f,1) & sa.t_sim <= frame_windows(f,2);
        h_meas = max(sa.I_sim(idx)) / I_nom;
        l_meas = min(sa.I_sim(idx)) / I_nom;
        dist   = abs(ach_high - h_meas) + abs(ach_low - l_meas);
        [~,p]  = min(dist);
        A_seen(p) = A_seen(p) + 1;
        if isnan(A_high(p))   % keep first occurrence only
            A_high(p) = h_meas;
            A_low(p)  = l_meas;
        end
    end
    %% Profile 2 fallback: load from per-profile file if not visited by active
    %% Profile 8: never in passive sequence — keep NaN, excluded like profile 4
    for p = [2]
        if isnan(A_high(p))
            fa = sprintf('active_%s_P%d_L100.mat', g, p);
            if isfile(fa)
                sa_p  = load(fa);
```

```
                n_a   = numel(sa_p.t_sim);
                idx_a = round(n_a*0.30):round(n_a*0.70);
                I_seg = sa_p.I_sim(idx_a) / I_nom;
                A_high(p) = prctile(I_seg, 90);
                A_low(p)  = prctile(I_seg, 10);
            end
        end
    end
 %% Profile 8 — never in passive sequence, exclude consistently
    P_high(8) = NaN; P_low(8) = NaN;
    A_high(8) = NaN; A_low(8) = NaN;

    results(g).P_high = P_high;
    results(g).P_low  = P_low;
    results(g).A_high = A_high;
    results(g).A_low  = A_low;
    results(g).A_seen = A_seen;
end

%% ── Per-profile tables and R² summary, per grid ──────────────────────
labels = {'Grid1','Grid2','Grid3'};
for g = 1:3
    fprintf('\n========== %s ==========\n', labels{g});
    fprintf('\nSLOW PROFILES (compared to OCP target)\n');
    fprintf('Prof | OCP_H  OCP_L | A_H    A_L   | P_H    P_L\n');
    fprintf('%s\n', repmat('-',1,60));
    for p = 1:8
        if ~slew_lim(p)
            fprintf(' %d   | %.3f  %.3f | %.3f  %.3f | %.3f  %.3f\n',...
                p, ocp_high(p), ocp_low(p),...
                results(g).A_high(p), results(g).A_low(p),...
                results(g).P_high(p), results(g).P_low(p));
        end
    end
 %% R² per group
    slow = find(~slew_lim);
    fast = find( slew_lim);
    A_vec_slow = [results(g).A_high(slow), results(g).A_low(slow)];
    P_vec_slow = [results(g).P_high(slow), results(g).P_low(slow)];
    O_vec_slow = [ocp_high(slow),          ocp_low(slow)];
    A_vec_fast = [results(g).A_high(fast), results(g).A_low(fast)];
    P_vec_fast = [results(g).P_high(fast), results(g).P_low(fast)];
    O_vec_fast = [ocp_high(fast),          ocp_low(fast)];
```

```
    %% Drop NaN profiles from both models equally (P2 active=NaN, P2 dropped from both)
    ok_Af = ~isnan(A_vec_fast);
    r2_A_slow = R2(A_vec_slow,          O_vec_slow);
    r2_P_slow = R2(P_vec_slow,          O_vec_slow);
    r2_A_fast = R2(A_vec_fast(ok_Af),   O_vec_fast(ok_Af));
    r2_P_fast = R2(P_vec_fast(ok_Af),   O_vec_fast(ok_Af));
    r2_PA     = R2([results(g).P_high, results(g).P_low],...
                   [results(g).A_high, results(g).A_low]);
    fprintf('\nFAST PROFILES (slew-limited — compared to OCP target)\n');
    fprintf('Prof | OCP_H  OCP_L | A_H    A_L   | P_H    P_L\n');
    fprintf('%s\n', repmat('-',1,60));
    for p = 1:8
        if slew_lim(p)
            fprintf(' %d   | %.3f  %.3f | %.3f  %.3f | %.3f  %.3f\n',...
                p, ocp_high(p), ocp_low(p),...
                results(g).A_high(p), results(g).A_low(p),...
                results(g).P_high(p), results(g).P_low(p));
        end
    end
    fprintf('\nR² summary:\n');
    fprintf('  Active  vs OCP (slow):       %.4f\n', r2_A_slow);
    fprintf('  Passive vs OCP (slow):       %.4f\n', r2_P_slow);
    fprintf('  Active  vs OCP (fast):       %.4f\n', r2_A_fast);
    fprintf('  Passive vs OCP (fast):       %.4f\n', r2_P_fast);
    fprintf('  Passive vs Active (overall): %.4f\n', r2_PA);
    not_seen = find(results(g).A_seen == 0);
    if ~isempty(not_seen)
        fprintf('  Active never visited profiles: %s\n', num2str(not_seen));
    end
end


%% ── R²_B: load-range robustness (slow profiles, 6 nominal currents) ─────
%  Reference = ocp_high * nominal (what the active current source hits exactly)
%  Files: passive_gridX_LYY.mat (18 files total)
nominal_list = [37.5,  75,  112.5,  150,  225,  300];
load_labels  = {'L25','L50','L75','L100','L150','L200'};
grids_b      = {'grid1','grid2','grid3'};
n_loads      = numel(nominal_list);
slow_profs   = find(~slew_lim);
R2B          = NaN(3, n_loads);
fprintf('\nR²_B — Load Range Robustness (slow profiles only)\n');
fprintf('%-10s', 'Grid');
```

```
for li = 1:n_loads; fprintf('  %8s', load_labels{li}); end
fprintf('\n%s\n', repmat('-',1,62));
for g = 1:3
    fprintf('%-10s', grids_b{g});
    for li = 1:n_loads
        fname = sprintf('passive_%s_%s.mat', grids_b{g}, load_labels{li});
        if ~isfile(fname); fprintf('  %8s', '---'); continue; end
        S   = load(fname);
        nom = nominal_list(li);
        ip_vals  = NaN(1, numel(slow_profs));
        ref_vals = NaN(1, numel(slow_profs));
        for si = 1:numel(slow_profs)
            p  = slow_profs(si);
            fi = find(p_frame_profile == p, 1);
            if isempty(fi); continue; end
            idx = S.t_sim >= frame_windows(fi,1) & S.t_sim <= frame_windows(fi,2);
            if sum(idx) < 10; continue; end
            ip_vals(si)  = max(S.I_sim(idx)) / nom;   % p.u.
            ref_vals(si) = ocp_high(p);                % p.u. target
        end
        valid = ~isnan(ip_vals) & ~isnan(ref_vals);
        if sum(valid) < 2; fprintf('  %8s', 'N/A'); continue; end
        ip  = ip_vals(valid);
        ref = ref_vals(valid);
        SS_res = sum((ip  - ref).^2);
        SS_tot = sum((ref - mean(ref)).^2);
        if SS_tot > 1e-12
            R2B(g,li) = 1 - SS_res/SS_tot;
            fprintf('  %8.4f', R2B(g,li));
        else
            fprintf('  %8s', 'N/A');
        end
    end
    fprintf('\n');
end
for g = 1:3
    fprintf('Grid %d:  R²_B best = %.4f,  worst = %.4f\n', ...
            g, max(R2B(g,:),[],'omitnan'), min(R2B(g,:),[],'omitnan'));
end
%% Figure: R²_B across load range
colors_b = {'b','r',[0.1 0.7 0.1]};


figure('Name','R²_B — Load Range Robustness');
```

```
hold on; grid on;
for g = 1:3
    plot(nominal_list, R2B(g,:), '-o', ...
         'Color', colors_b{g}, 'LineWidth', 2, 'MarkerSize', 8, ...
         'MarkerFaceColor', colors_b{g}, ...
         'DisplayName', sprintf('Grid %d', g));
end
yline(0.99, 'k--', 'LineWidth', 1.5, 'DisplayName', 'R²=0.99 reference');
yline(1.00, 'k:',  'LineWidth', 0.8);
xlabel('Nominal current [A]',  'FontSize', 12);
ylabel('R²_B',                 'FontSize', 12);
title({'R²_B: Passive vs Active Target Across Load Range', ...
       'Slow profiles only | Reference = FCN high\_frac × nominal'}, ...
      'FontSize', 12);
legend('Location', 'southwest', 'FontSize', 11);
xticks(nominal_list); xticklabels(load_labels);
ylim([0.97 1.005]);
for g = 1:3
    [~, worst] = min(R2B(g,:));
    if ~isnan(R2B(g,worst))
        text(nominal_list(worst), R2B(g,worst) - 0.001, ...
             sprintf('%.4f', R2B(g,worst)), ...
             'HorizontalAlignment', 'center', ...
             'FontSize', 9, 'Color', colors_b{g});
    end
end


%% ── Voltage comparison — Passive vs Active, 150 A nominal ──────────────
%  OCP limit: ±3% of 800 V = ±24 V
%  Note: Grid 3 sensor is at AC source terminal (not load terminal) — excluded.
OCP_V_abs = 0.03 * 800;   % 24 V
p_files_v = {'passive_grid1_L100.mat','passive_grid2_L100.mat','passive_grid3_L100.mat'};
a_files_v  = {'active_grid1_L100.mat', 'active_grid2_L100.mat', 'active_grid3_L100.mat'};
grid_labels_v = {'Grid 1 (R only)', 'Grid 2 (R+diode)', 'Grid 3 (AC source)*'};
Vmax_pas  = NaN(1,3); Vmax_act  = NaN(1,3);
Vmean_pas = NaN(1,3); Vmean_act = NaN(1,3);
Vstd_pas  = NaN(1,3); Vstd_act  = NaN(1,3);
Vvar_pas  = NaN(1,3); Vvar_act  = NaN(1,3);
for g = 1:3
    if isfile(p_files_v{g})
        sp  = load(p_files_v{g}); n = numel(sp.V_sim);
        idx = round(n*0.20):round(n*0.80);   % middle 20-80% avoids startup transients
        Vmax_pas(g)  = max(abs(sp.V_sim - 800));
```

```
        Vmean_pas(g) = mean(sp.V_sim(idx));
        Vstd_pas(g)  = std(sp.V_sim(idx));
        Vvar_pas(g)  = (max(sp.V_sim(idx)) - min(sp.V_sim(idx))) / 800 * 100;
    end
    if isfile(a_files_v{g})
        sa  = load(a_files_v{g}); n = numel(sa.V_sim);
        idx = round(n*0.20):round(n*0.80);
        Vmax_act(g)  = max(abs(sa.V_sim - 800));
        Vmean_act(g) = mean(sa.V_sim(idx));
        Vstd_act(g)  = std(sa.V_sim(idx));
        Vvar_act(g)  = (max(sa.V_sim(idx)) - min(sa.V_sim(idx))) / 800 * 100;
    end
end


%% Voltage table
fprintf('\nVOLTAGE COMPARISON — Passive vs Active — 150 A Nominal\n');
fprintf('OCP limit: ±%.0f V (±3%% of 800 V)\n', OCP_V_abs);
fprintf('%-22s  %-8s  %-8s  %-8s  %-8s  %-8s  %-8s\n', ...
    'Grid', 'P_max', 'A_max', 'P_mean', 'A_mean', 'P_std', 'A_std');
fprintf('%-22s  %-8s  %-8s  %-8s  %-8s  %-8s  %-8s\n', ...
 '', 'dV[V]', 'dV[V]', 'V[V]', 'V[V]', '[V]', '[V]');
fprintf('%s\n', repmat('-',1,85));
for g = 1:3
    ocp_p = 'PASS'; if Vmax_pas(g) > OCP_V_abs; ocp_p = 'FAIL'; end
    ocp_a = 'PASS'; if Vmax_act(g) > OCP_V_abs; ocp_a = 'FAIL'; end
    fprintf('%-22s  %-8.3f  %-8.3f  %-8.3f  %-8.3f  %-8.4f  %-8.4f  Pas:%s  Act:%s\n', ...
        grid_labels_v{g}, Vmax_pas(g), Vmax_act(g), ...
        Vmean_pas(g), Vmean_act(g), Vstd_pas(g), Vstd_act(g), ocp_p, ocp_a);
end
fprintf('* Grid 3: sensor at AC source terminal — load-side drop not captured.\n');


%% Figure: voltage waveforms, all grids
colors_pas = {[0.8 0.1 0.1],[0.8 0.1 0.1],[0.8 0.1 0.1]};
colors_act = {[0.1 0.3 0.8],[0.1 0.3 0.8],[0.1 0.3 0.8]};


figure('Name','Voltage Comparison: Passive vs Active');
for g = 1:3
    subplot(3,1,g); hold on; grid on;
    if isfile(p_files_v{g})
        sp = load(p_files_v{g});
        plot(sp.t_sim,  sp.V_sim,  '-',  'Color',  colors_pas{g},  'LineWidth',  0.7,
'DisplayName','Passive');
    end
```

```
    if isfile(a_files_v{g})
        sa = load(a_files_v{g});
        plot(sa.t_sim,  sa.V_sim,  '-',  'Color',  colors_act{g},  'LineWidth',  1.2,
'DisplayName','Active');
    end
    yline(800,            'k-',  'LineWidth', 0.8, 'DisplayName','800V nominal');
    yline(800 + OCP_V_abs, 'r--', 'LineWidth', 1.2, 'DisplayName','+3% OCP (+24V)');
    yline(800 - OCP_V_abs, 'r--', 'LineWidth', 1.2, 'HandleVisibility','off');
    ylabel('Voltage [V]','FontSize',10); xlabel('Time [s]','FontSize',10);
    if g < 3
        title(sprintf('%s  |  Passive MaxΔV=%.3fV (%s)  |  Active MaxΔV=%.3fV', ...
              grid_labels_v{g}, Vmax_pas(g), ...

char(string(Vmax_pas(g)<=OCP_V_abs).replace('true','PASS').replace('false','FAIL')), ...
              Vmax_act(g)), 'FontSize', 10);
    else
        title(sprintf('%s  |  Sensor at source terminal — load-side drop not captured',
...
              grid_labels_v{g}), 'FontSize', 10);
    end
    legend('Location','best','FontSize',8); xlim([0 5]);
end
sgtitle({'Bus Voltage: Passive vs Active — 150 A Nominal (All 8 OCP Profiles)', ...
         'Red dashed = OCP ±3% limit (±24 V)'}, 'FontSize', 11);


%% Figure: zoomed transition at t ≈ 2.5 s (P3→P4 boundary)
figure('Name','Voltage Zoom — Profile Transition (t≈2.5s)');
t_zoom_start = 2.45; t_zoom_end = 2.65;
for g = 1:2   % Grid 3 excluded
    subplot(1,2,g); hold on; grid on;
    if isfile(p_files_v{g})
        sp   = load(p_files_v{g});
        mask = sp.t_sim >= t_zoom_start & sp.t_sim <= t_zoom_end;
        plot(sp.t_sim(mask)*1e3, sp.V_sim(mask), '-', 'Color', colors_pas{g}, 'LineWidth',
1.5, 'DisplayName','Passive');
    end
    if isfile(a_files_v{g})
        sa   = load(a_files_v{g});
        mask = sa.t_sim >= t_zoom_start & sa.t_sim <= t_zoom_end;
        plot(sa.t_sim(mask)*1e3, sa.V_sim(mask), '-', 'Color', colors_act{g}, 'LineWidth',
1.5, 'DisplayName','Active');
    end
    yline(800,            'k-',  'LineWidth',0.8,'DisplayName','800V');
    yline(800 + OCP_V_abs, 'r--', 'LineWidth',1.2,'DisplayName','+3% OCP');
    yline(800 - OCP_V_abs, 'r--', 'LineWidth',1.2,'HandleVisibility','off');
```

```
    xlabel('Time [ms]','FontSize',11); ylabel('Voltage [V]','FontSize',11);
    title(sprintf('%s — Profile transition at t≈2.5s', grid_labels_v{g}),'FontSize',11);
    legend('Location','best','FontSize',9);
end
sgtitle({'Zoomed Voltage at Profile Transition (P3→P4 boundary)', ...
         'Passive inductive spike visible  |  Active: flat'}, 'FontSize',12);


%% Figure: summary bar chart (max ΔV, grids 1 & 2 only)
figure('Name','Voltage Summary Bar Chart');
hold on; grid on;
x = 1:2;
bar(x-0.2, Vmax_pas(1:2), 0.35, 'FaceColor',[0.8 0.2 0.2], 'DisplayName','Passive');
bar(x+0.2, Vmax_act(1:2), 0.35, 'FaceColor',[0.2 0.4 0.8], 'DisplayName','Active');
yline(OCP_V_abs, 'r--', 'LineWidth', 2, 'DisplayName', sprintf('OCP = %.0fV',OCP_V_abs));
set(gca,'XTick',x,'XTickLabel',{'Grid 1','Grid 2'},'FontSize',12);
ylabel('Max |V_{bus} - 800V|  [V]','FontSize',12);
title({'Max Voltage Deviation from 800V Nominal', ...
       '(Grid 3 excluded: sensor at source terminal)'},'FontSize',12);
legend('Location','northeast','FontSize',11);
for g = 1:2
    text(g-0.2, Vmax_pas(g)+0.05, sprintf('%.3fV',Vmax_pas(g)), ...
         'HorizontalAlignment','center','FontSize',10,'FontWeight','bold');
    text(g+0.2, Vmax_act(g)+0.05, sprintf('%.3fV',Vmax_act(g)), ...
         'HorizontalAlignment','center','FontSize',10,'FontWeight','bold');
end
ylim([0, max([Vmax_pas(1:2), Vmax_act(1:2), OCP_V_abs])*1.3]);
```

A.2. Parameters tuning for grid variations

```
%% Grid Parameter Identification — V-I fit from ABB load files
%% Operating points: mean steady-state current and voltage drop per load level
%  Middle 30% of each simulation used to avoid profile transition ramps
I_ss = [263, 525, 788, 1050, 1575, 2100];   % mean I_high [A]
dV   = [0.5, 1.6, 2.7,  3.8,   6.2,  8.9]; % 800 - V_mean [V]
%% Grid 1: purely resistive fit, forced through origin (dV = R * I)
R_grid1 = (I_ss * I_ss') \ (I_ss * dV');
fprintf('Grid 1:  R = %.6f Ohm\n', R_grid1);
%% Grid 2: resistive + constant offset fit (dV = V_diode + R * I)
A       = [ones(6,1), I_ss'];
params  = A \ dV';
V_diode = params(1);
R_grid2 = params(2);
```

```
fprintf('Grid 2:  V_diode = %.4f V,  R = %.6f Ohm\n', V_diode, R_grid2);
%% R² for both fits
SS_tot   = sum((dV - mean(dV)).^2);
dV_pred1 = R_grid1 * I_ss;
dV_pred2 = V_diode + R_grid2 * I_ss;
fprintf('R² Grid 1 (R only):   %.4f\n', 1 - sum((dV - dV_pred1).^2) / SS_tot);
fprintf('R² Grid 2 (R+diode):  %.4f\n', 1 - sum((dV - dV_pred2).^2) / SS_tot);
%% Figure: V-I characteristic with both fits
figure;
plot(I_ss, dV, 'ko', 'MarkerSize', 10, 'DisplayName', 'ABB data');
hold on;
plot(I_ss, dV_pred1, 'b-',  'LineWidth', 2, ...
    'DisplayName', sprintf('Grid 1: R = %.5f Ohm', R_grid1));
plot(I_ss, dV_pred2, 'r--', 'LineWidth', 2, ...
    'DisplayName', sprintf('Grid 2: Vd = %.3f V, R = %.5f Ohm', V_diode, R_grid2));
legend; grid on;
xlabel('Mean current [A]');
ylabel('Voltage drop [V]');
title('V-I characteristic extracted from all six load percentages');
```

A.3. Extract values per model, per grid, per load

```
%% LOAD 100 SCENARIO — nominal_current = 150A
%% Passive Grid 1
set_param('PassiveLoadModel_Grid1/Nominal current', 'Value', '150');
out = sim('PassiveLoadModel_Grid1');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('passive_grid1_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('passive_grid1_L100 saved');

%% Passive Grid 2
set_param('PassiveLoadModel_Grid2/Nominal current', 'Value', '150');
out = sim('PassiveLoadModel_Grid2');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('passive_grid2_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('passive_grid2_L100 saved');

%% Passive Grid 3
set_param('PassiveLoadModel_Grid3/Nominal current', 'Value', '150');
```

```
out = sim('PassiveLoadModel_Grid3');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('passive_grid3_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('passive_grid3_L100 saved');

%% Active Grid 1
set_param('ActiveLoadModel_Grid1/Nominal current', 'Value', '150');
out = sim('ActiveLoadModel_Grid1');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('active_grid1_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('active_grid1_L100 saved');

%% Active Grid 2
set_param('ActiveLoadModel_Grid2/Nominal current', 'Value', '150');
out = sim('ActiveLoadModel_Grid2');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('active_grid2_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('active_grid2_L100 saved');

%% Active Grid 3
set_param('ActiveLoadModel_Grid3/Nominal current', 'Value', '150');
out = sim('ActiveLoadModel_Grid3');
t_sim = out.tout;
I_sim = out.logsout.getElement('I_sim').Values.Data;
V_sim = out.logsout.getElement('V_sim').Values.Data;
save('active_grid3_L100.mat', 't_sim', 'I_sim', 'V_sim');
disp('active_grid3_L100 saved');
```

List of Figures

Figure 1-1 - Actual power envelope of a GPU rack during a Large Language Model (LLM) run [1] 17

Figure 1-2 - Power envelope over a full cycle [1] 17

Figure 2-1 - Top-level Simulink model 19

Figure 2-2 - Internal signal flow showing FCN, integrator, and feedback path 20

Figure 2-3 - Zooming into 0.46–0.56 s shows the staircase descent of Profile 2 to 1 transition 22

Figure 2-4 - Simulation scope output for T=5s 25

Figure 3-1 - Controlled current source circuit (Simulink) 28

Figure 4-1 - Passive load model circuit in Simulink 29

Figure 4-2 - Passive load model current scope output 32

Figure 4-3 - Profile 7 to 3 transitions. Left-current model. Right-passive model 33

Figure 4-4 - Passive load during the 7 to 3 frame transition: exponential transitions due to RL time constant τ = L/R" and "inductive overshoot at t ≈ 2.5 s 34

Figure 4-5 - t=0.5 s transition between 1 and 2 frames (profiles 2 to 1). Left-active model. Right-passive model 34

Figure 6-1 - V-I characteristic of the supply extracted from all six load percentages . 42

Figure 6-2 - Passive load model with Grid 1 (V + R) at the supply side 44

Figure 6-3 - Active load model with Grid 1 (V+R) 44

Figure 6-4 - Passive load model with Grid 2 (V + R + Diode) 45

Figure 6-5 - Active load model (subsystem) with Grid 2 (V + R + Diode) 45

Figure 6-6 - Passive load model with Grid 3 (controlled voltage source) 46

Figure 6-7 - Active load model with Grid 3, full system view including the controlled voltage source and the load profile selector 47

List of Tables

Table 1-1 - Table 6 from "OCP Diablo 400" documentation [2] 18
Table 1-2 - Table 8 from "OCP Diablo 400" document [2] 18
Table 2-1 - Phase State Machine summary 22
Table 2-2 - Full Threshold Analysis- complete table - All 8 Lines @ 150A Nominal ... 24
Table 2-3 - Summary of parameters and their effects on the current waveform 26
Table 4-1 - All the resistances explained 31
Table 4-2 - Passive load model values 35
Table 5-1 - Summary of the 2 models 38
Table 7-1 - load percentages, nominal_current values and measured bus voltage variation in each file 50
Table 7-2 - Profile classification at 150 A nominal 53
Table 7-3 - Extracted current levels vs OCP targets for slow profiles at 150 A nominal 56
Table 7-4 - Extracted fast-profile vs OCP target current levels at 150 A nominal 57
Table 7-5 - R^2 of active and passive models against OCP targets for fast 58
Table 7-6 - R^2_B: passive model slow-profile accuracy vs theoretical active target across six load levels for the 3 grids 59
Table 7-7 - Voltage deviation from 800 V nominal at 150 A nominal. *Grid 3 voltage sensor is at the AC source terminal 60
Table 7-8 - Summary of all comparison metrics 60

List of symbols

Variable	Description	SI unit
Electrical Quantities		
V	DC bus voltage (nominal 800 V)	V
I	Electrical current	A
I_high	High (peak/active) current threshold for a given OCP profile	A
I_low	Low (idle) current threshold for a given OCP profile	A
I_nominal	Nominal (rated) current used to scale the OCP profile	A
I_avg	Average current	A
I_rms	Root-mean-square current	A
P	Electrical power	W
dI/dt	Rate of change of current (slew rate)	A/s
Circuit Components		
R	Electrical resistance (generic)	Ω
R_grid	Series cable resistance between PSU and server rack	Ω
R_aux / high_R	Auxiliary resistor, always connected; delivers idle current I_low	Ω
RL_jump/ base_R	Switched parallel resistor, connected during active pulse to raise current to I_high	Ω
R_I_high	Ideal resistance for delivering I_high (computed as V/ I_high)	Ω
R_I_low	Ideal resistance for delivering I_low (computed as V/ I_low)	Ω
L	Electrical inductance (generic)	H
L_slewRate	Series inductor enforcing physical slew rate (L = V / sr)	H
C	Electrical capacitance (generic)	F
C_DC	Bulk DC bus capacitor; limits voltage transients from current steps	F
C1 / C_1	Load-side capacitor in the active (controlled current source) model	F
Time and Simulation Parameters		
τ	RL time constant, $\tau = L / R$	s
t_high	Duration of the high-current (rising / active) phase per OCP profile	s
t_low	Duration of the low-current (falling / idle) phase per OCP profile	s
T_s	Simulation timestep	s
tmr	Simulation timer (elapsed time from start)	s
tmr0	Timer snapshot at the last profile-change or phase-reset event	s

delta	Elapsed time within the current phase: delta = tmr − tmr0	s
Model and FCN Variables		
sr	PSU slew rate (1 A/μs = 1 × 10^6 A/s for most profiles; 0.1 A/μs for Profile 4)	A/s
gain	Rate-of-change output of the FCN, integrated to produce current: I(t) = I(t−1) + gain · T_s	A/s
old_value	Current value at the previous timestep, fed back into the FCN for threshold clamping	A
initial_sign	Direction flag (±1) used by the FCN to steer current toward the correct threshold when a profile change occurs	–
line	Profile index (integer 1-8), selects the active row of the OCP Table 8 parameter arrays	–
D	Duty cycle of the pulse generator driving the switched relay in the passive model	–
scale	Scaling factor applied to resistance values to account for nominal current setting	–
Performance Metrics		
R^2	Coefficient of determination; measures how closely extracted current levels match reference targets (OCP or achievable)	–
R^2_B	Load-range robustness metric; R^2 computed across six load levels to assess accuracy over the full operating range	–
ΔV	Voltage deviation from the 800 V nominal bus voltage	V

