DAB converter discrete ADRC control into real-time CHIL simulation of a MVDC/LVDC power grid

Alessio Clerici, Riccardo Chiumeo, Diego Raggini, Alessandro Veroni
Ricerca Sul Sistema Energetico - RSE S.p.A.
Via Rubattino 54, 20134
Milan, Italy
Tel.: +39 02 399 21
E-Mail: alessio.clerici@rse-web.it
URL: https://www.rse-web.it

Acknowledgements

This work has been financed by the Research Fund for the Italian Electrical System in compliance with the Decree of April 16, 2018.

Keywords

«Real-time simulation», «Distribution of electrical energy», «Microcontrollers», «Converter control», «Dual Active Bridge (DAB) DC-DC converter»

Abstract

Active Disturbance Rejection Control (ADRC) is implemented into commercial microcontroller to drive a Dual Active Bridge converter fed by a Medium Voltage Direct Current network. A real-time Control Hardware In the Loop system is successfully implemented and performance meets conventional off-line simulations results. ADRC proves to be a robust control strategy for a distribution network converters and Hardware In the Loop a valid technique for power electronics tests.

Introduction

Active Disturbance Rejection Control (ADRC) is currently developed for high-performance and high-dynamics demanding applications, like Permanent Magnet Synchronous Motors (PMSM) drivers and speed controllers [1] [2]; in such cases, ADRC stability margin is approximately the same as traditional PI controller, however, ADRC is superior in rapidity and overshoot performance [3].
According to these encouraging results, this work aims to implement ADRC in a Dual Active Bridge (DAB) power converter by means of real-time Control Hardware In the Loop (CHIL) simulations.
CHIL testing is widely used in automotive, aerospace, and robotics, while usage in power electronics is still in early stage [4].
The growing need for advanced grid support functions and the study of innovative distribution networks and power converters can be a first step to increase the development of such test technique in this field.
ADRC is based on optimal control strategy: the regulator, through an Extended State Observer (ESO), can estimate system variables, allowing disturbance rejection.
Professor Han theorized ADRC [5] starting from the PID controller analysis; in the last decade, ADRC has seen growing development both theoretical [6] and practical [7] [8] [9] [10].
The control is intrinsically versatile since it combines simple feasibility of PID with modern approach based on model analysis and state observers.
Compared to PID control, ADRC can achieve better setpoint tracking capability and superior disturbance rejection.
ESO operates in parallel to the controlled process; it is mathematically modeled on the process itself, keeping the characteristic matrices and the system order.
The observer acquires the input and output values of the controlled process and calculates a disturbance estimate.

Equation (1) describes first order state observer algorithm in Laplace domain.

$$\begin{cases} \hat{x}_1 = \dfrac{l_1 \cdot (y - \hat{x}_1) + \hat{f} + b_0 \cdot u}{s} \\ \hat{f} = \dfrac{l_2 \cdot (y - \hat{x}_1)}{s} \end{cases} \tag{1}$$

Equation (2) briefly models the concept of ADRC technique; $\dot{y}(t)$ represents the time derivative of process output. By definition of time derivative, it is possible to rewrite $\dot{y}(t)$ as the difference between reference setpoint *q(t)* and system output *y(t)* in time domain; an arbitrary constant K_A can thus modulate the control speed response.

$$\dot{y}(t) = u_0(t) = K_A \cdot \left[q(t) - y(t) \right] \tag{2}$$

Coefficients b_0, K_A, l_1 and l_2 have been derived from the theoretical analysis of [6].
Fig. 1 shows whole ADRC control structure to be implemented in digital simulations, with a detail of ESO transfer function.

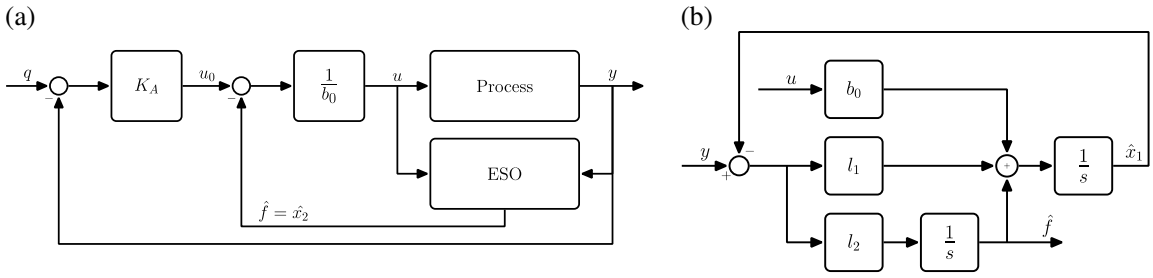

Fig. 1: ADRC block scheme. (a) Whole plant structure. (b) ESO transfer function.

ADRC implementation into a digital, general-purpose microcontroller is the aim of the present article. Assuming T_s as sampling period, ESO transfer function in Laplace domain has been discretized by using the bilinear transformation (or Tustin's method) [11], whose general formulation is shown in (3):

$$y(n) = y(n-1) + \frac{T_s}{2} \cdot \left[x(n) + x(n-1) \right] \tag{3}$$

Test bench for ADRC analysis is on voltage regulation of a Dual Active Bridge (DAB) converter. DAB converter operates into a Control Hardware in the Loop (CHIL) simulation, where the Medium Voltage Direct Current (MVDC) grid of Fig. 2 is simulated.

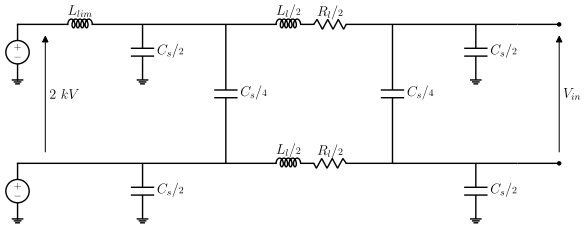

Fig. 2: MVDC grid model used in CHIL simulations.

MVDC grid parameters are shown in Table I:

Table I: **MVDC grid parameters**

Parameter	**Quantity**
Line length (l)	1 km
Line resistance (R_l)	0.0283 Ohm
Line inductance (L_l)	0.4 mH
Upstream inductance (L_{lim})	0.4 mH
Service capacitance (C_s)	0.92 µF

ADRC validation in CHIL context is then composed by two steps:

1) respect of hard real-time constraints and synchronization;
2) system response comparison with off-line implementation of the same ADRC control.

In next paragraph, discretized ADRC algorithm implementation strategy will be presented, with great focus on time constraints imposed by the hardware. The target is to maximize the system performances to guarantee an effective exchange of data with the CHIL real-time simulator.
Later on, a summary of non-idealities introduced by the CHIL simulation will serve to define a correct off-line reference model to validate CHIL real-time simulation results.
Both off-line and CHIL simulations results comparison will be shown in the third paragraph: ADRC characteristics will be investigated through the response of a DAB converter.
Paper nomenclature is reported in Table II:

Table II: **Nomenclature**

Symbol	Description
ADC	Analog to Digital Converter
ADRC	Active Disturbance Rejection Control
b_0	ADRC rated value of input variable disturbance
CHIL	Control Hardware in the Loop
CPU	Central Processing Unit
d	Quantized signal
DAB	Dual Active bridge (converter)
FPGA	Field Programmable Gate Array
IGBT	Insulated Gate Bipolar Transistor
ISR	Interrupt Service Routine
K_A	ADRC static gain
l_1	Luenberger matrix first coefficient
l_2	Luenberger matrix second coefficient
LVDC	Low Voltage Direct Current
MVDC	Medium Voltage Direct Current
N	ADC n° of bits
OC	Output compare
PID	Proportional Integral Derivative (control)
PMSM	Permanent Magnet Synchronous Motors
PWM	Pulse-Width Modulation
q	System setpoint
round( )	Round-to-nearest mathematical function
SPS	Single Phase Shift
T_c	Control period
$T_{s\mu c}$	Microcontroller discretization period
T_{sCPU}	Real-time simulator CPU module discretization period
T_{sFPGA}	Real-time simulator FPGA module discretization period
T_φ	Time-delay between DAB H-bridges voltages
V_{in}	DAB input voltage
V_{out}	DAB output voltage
x	System state
y	System output
ΔV_{MAX}	Sampled voltage range

Timing and discretization of real-time algorithm

ADRC implementation in a physical microcontroller for real-time simulation implies a discretization of the control strategy. Facing the inevitable computational limitations, the chosen time-step must not interfere with system dynamics.

In a real-time system, control computation time, here called control cycle (or control period) T_c, cannot be greater than the microcontroller discretization period $T_{s\mu c}$. For the entire control cycle execution, this trivial condition must be always true:

$$T_c \leq T_{s\mu c} \tag{4}$$

Control is fed with inputs sampled at the beginning of the control cycle. In this way, $T_{s\mu c}$ also becomes the input sampling period and the output update rate.
Implementation strategy chosen in this work allows for an easy and effective structure, guaranteeing a constant $T_{s\mu c}$ period.
It makes use of the hardware resources of the available microcontroller (ST NUCLEO-F767ZI) and of the abstraction layer provided by the MicroPython programming language, already employed in previous CHIL setups [12], to keep the code simple.
The discretization period is tied to a high-precision hardware timer rising an interrupt to periodically update the control variables (the result of the computation of the ADRC algorithm).
The same timer also triggers ADRC code execution, allowing for a strict scheduling of the computation times. A high-level description of the process is shown in Fig. 3.

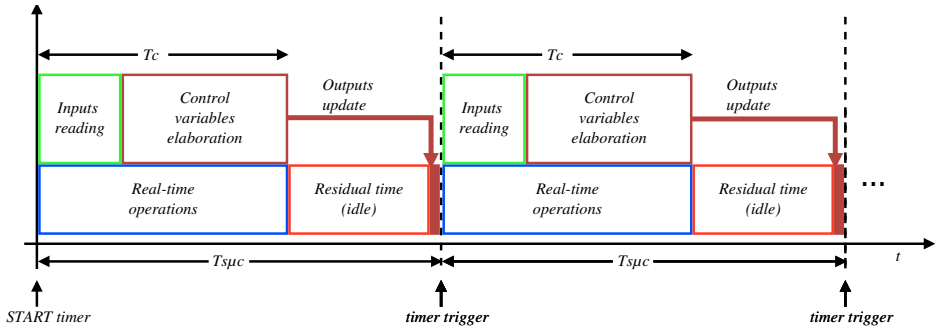

Fig. 3: Time diagram of the control cycle execution.

Notwithstanding a residual (idle) time, output update is realized at the end of $T_{s\mu c}$ to achieve a coherently discretized system. The new value of the control variable is then stored in the relevant output compare (OC) registers of the output channels. Those are natively managed by the hardware interrupts and thus run independently from the main execution cycle, effectively decoupling the two main tasks assigned to the microcontroller. This strategy allows for a stable and constant generation of all the control signals, minimizing the latency and jitter typical of software-based gate generation, leveraging the advanced ISR (Interrupt Service Routine) capabilities of the microcontroller. IGBT pulses are generated at the DAB switching frequency (5 kHz) and are therefore independent from $T_{s\mu c}$.
The selection of $T_{s\mu c}$ must then be related to the discretization periods of the various components of the CHIL simulation. In this work, an OPAL-RT simulator is used. The simulated model is divided between a CPU and an FPGA module, working according to the scheme of Fig. 4.

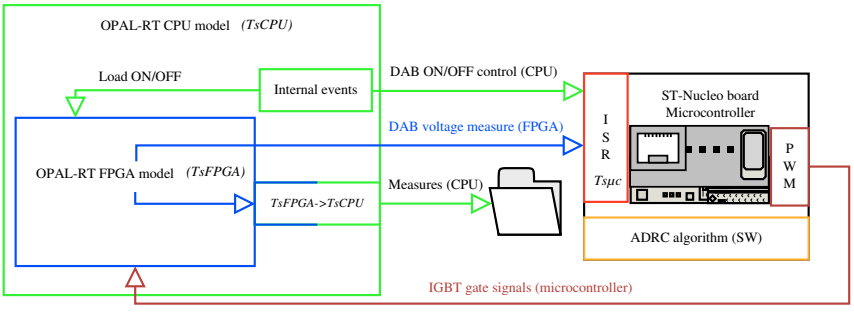

Fig. 4: CHIL simulation structure, composed by systems having different discretization periods: OPAL-RT CPU module (green), OPAL-RT FPGA module (blue) and ST-Nucleo microcontroller (orange). Code execution is scheduled by an ISR system (red), the PWM signals generation (brown) is managed by microcontroller hardware resources.

Besides ADRC sampling rate, OPAL-RT simulation of the network is then discretized as follows:

$$T_{sCPU} = 20\,\mu s$$
$$T_{sFPGA} = 1\,\mu s \tag{5}$$

The ADRC regulator is used to control the DAB secondary voltage. This converter is then represented in the FPGA module, allowing for the real-time exchange of the gate signals and feedback measures between the machine and the implemented control. At each $T_{s\mu c}$ step, the microcontroller must sample the voltage value and compute a new reference for DAB control variable (time-delay between H-bridges voltages T_{φ}[1]). Considering the control characteristics of the DAB converter [13] and the typical execution time of the digital implementation of the ADRC algorithm, a 300 µs discretization period was selected.

$$T_{s\mu c} = 300\,\mu s \tag{6}$$

The result is the coexistence of three distinct discrete systems, with different discretization intervals, together with a "low level hardware" layer, responsible for the scheduling of microcontroller operations and the generation of pulses to the gates of the IGBTs.
The signals exchange between the different sections, where necessary, implies a "conversion" in terms of discretization period; for example, in order to store key measures, some signals coming from the FPGA module are transmitted to the CPU module and therefore must be properly decimated; this is necessary because the system allows the saving of measurements through CPU module only.

Signal quantization aspects

Off-line simulations have been developed by means of ATPDraw graphical interface software, thus using ATP (Alternative Transient Program) as model solver.
The same model has been then implemented in MATLAB/Simulink environment, by using SimPower System Toolbox blocks. OPAL-RT also uses SimPower System Toolbox blocks to program FPGA module, thus, once the model is set, translation between off-line and real time simulation is quite straightforward.
Effective comparison between ADRC CHIL results and off-line reference simulation is possible by slightly tweaking the off-line implementation: this is necessary to reduce the effects of non-idealities arising when implementing the control strategy in a physical microcontroller.
In this specific case, a key effect is the quantization of the ADC (Analog-to-Digital Converter), performed by the microcontroller to sample the feedback measure.
The simplest model to uniform quantization introduced by the ADC is reported in (7). As mentioned above, it is then straightforward to implement the same quantizer block in the reference off-line simulations and in CHIL real time environments:

$$d = \frac{\Delta V_{MAX}}{2^N - 1} \cdot round\left(\frac{2^N - 1}{\Delta V_{MAX}} y \right) \tag{7}$$

with: d is the quantized signal, y is the input signal from the sampling stage, N is the bit resolution of the hardware selected (in this case, 12-bit [14]).

Off-line vs. CHIL simulation comparison

The complexity of the implemented circuital model is not primarily relevant for the present work, as the focus is related to preliminary analysis of the ADRC on a microcontroller.
The model (Fig. 5), implemented both in the off-line and CHIL real-time simulation, consists of a 1 MW DAB converter which interconnects a MVDC grid to a LVDC one.
DAB is fed by MVDC at nominal 2000 V and must regulate LVDC voltage at nominal 750 V.

[1] In this work, for simplicity, the DAB converter is controlled in Single Phase Shift (SPS) mode [15].

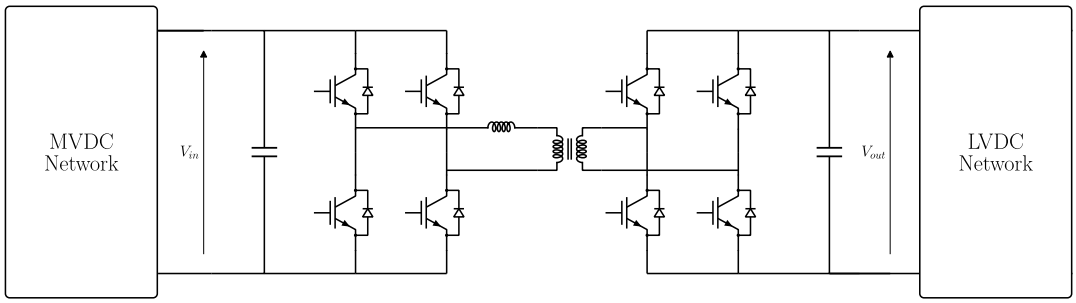

Fig. 5: Implemented model.

The simulation test setup has been arranged to verify the response of the ADRC regulator facing grid disturbances.
Events are organized as follows:

- t<0.1 s: the DAB regulates, at no-load, the voltage (750 V), no power transfer from MVDC to LVDC side;
- t=0.1s: a heavy resistive load is connected to DAB LVDC side (resistance $R_{load} = 0.5625\ \Omega$ has been calculated to absorb 1 MW at nominal 750 V); DAB must keep LVDC voltage stable;
- t=0.5s: the resistive load is disconnected; DAB must handle such heavy load rejection keeping LVDC voltage stable.

Comparison between off-line reference simulations (ATPDraw and Simulink) and CHIL test (OPAL-RT) is shown in Fig. 6 and Fig. 7.
Following signals have been plot:

- DAB power profile;
- DAB output voltage V_{out};
- DAB control variable[2] T_{φ}.

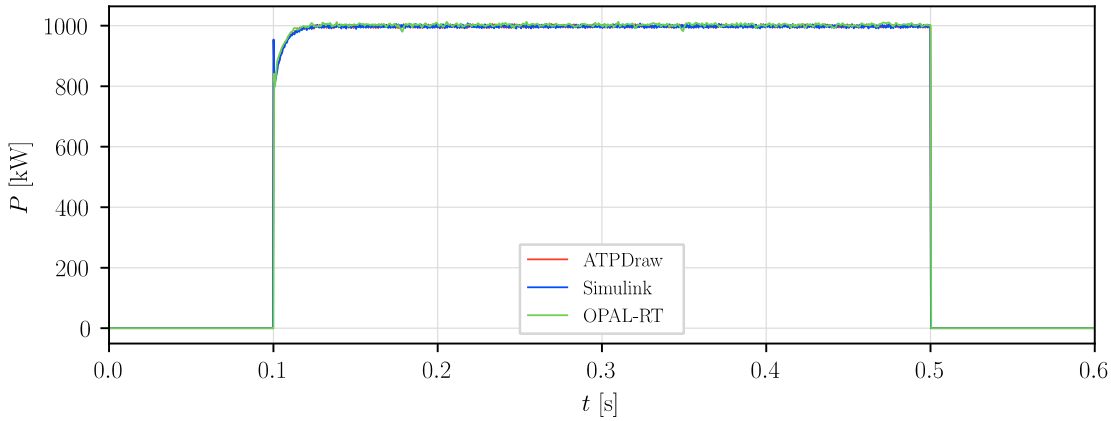

Fig. 6: DAB output power. Off-line reference simulations (ATPDraw and Simulink) and CHIL test (OPAL-RT).

[2] T_{φ} computed value is not readily available for storage in the real-time simulator as other variables, since the microcontroller outputs directly the gate signals. The value used here is the oscilloscope time-delay measure between PWM pulses applied to the converter.

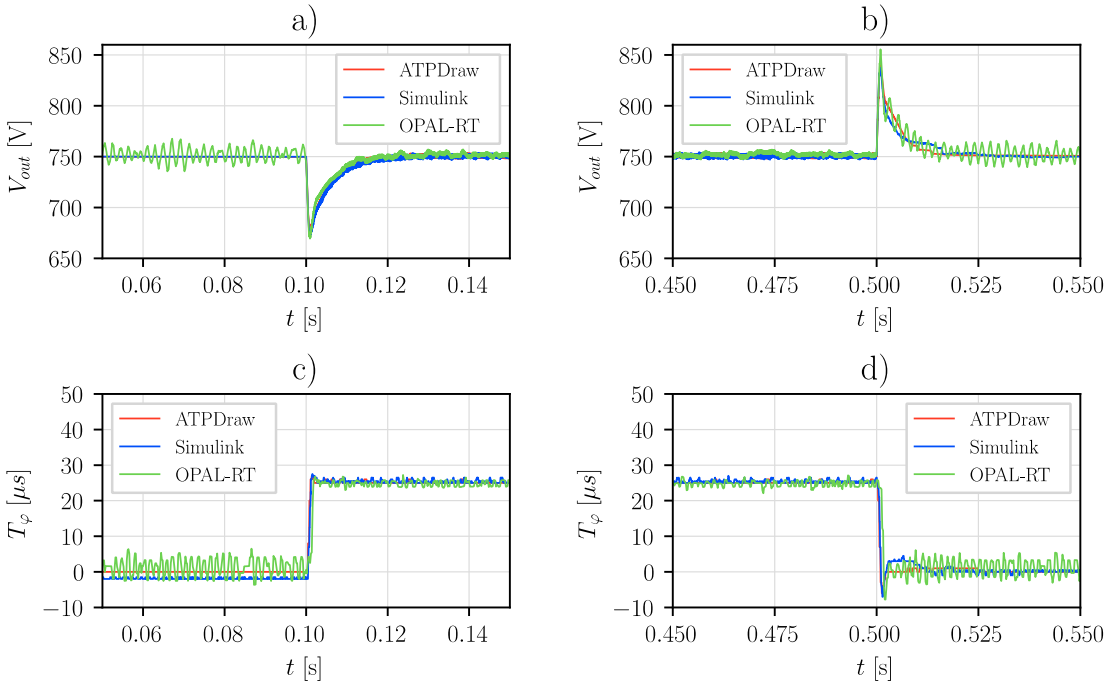

Fig. 7: DAB output voltage V_{out} and control variable T_φ at loading (a and c) and unloading (b and d). Off-line reference simulations (ATPDraw and Simulink) and CHIL test (OPAL-RT).

Both figures show a generally good level of correspondence between the results obtained off-line and real-time. The system response following each transient is correctly represented and shows no deviation across the various ADRC control algorithm implementations. V_{out} regulated voltage (Fig. 7 – a and b) shows the expected step variation at t = 0.10 s and t = 0.50 s, after which the control imposes an almost first order exponential dynamics to recover the nominal voltage.
The same can be said about T_φ converter control variable (Fig. 7 – c and d): it reaches the same steady state value corresponding to the nominal phase-shift of the converter. Transient deviations in rising and falling fronts are compatible with slight inaccuracies in measurement synchronization.
In Fig. 7 – c it can be observed a small steady-state error between ATPDraw and Simulink. This minor difference is given essentially by the behavior the DAB converter manifests at no-load: very small variations in control variable (T_φ) cause a consistent variation in controlled variable (V_{out}); as a result, in both cases output voltage is set properly at 750 V, with two slightly different values of T_φ in the two simulations. Such aspects can't be completely compensated, mainly depending in simulation environments asymmetries.
In CHIL results there is also a high-frequency component superimposed to control variable profile at no-load. Once again, despite OPAL-RT uses the same SimPower System blocks to implement the circuit, FPGA code translation and then compilation produces slight differences in the final model setup. Such effect is observable also in the regulated voltage profile, but it is greatly reduced at given load. Anyway, no significant impact of such oscillation can be seen in the general voltage regulation performance, the value being on average the same of reference off-line simulations.
The key phenomenon giving rise to this oscillation can be:

- the control variable higher instability in CHIL simulations can be due to the way T_φ is acquired. Since microcontroller computed value cannot be outputted, it has been acquired by means of an external oscilloscope capture setup. No-load values are then more difficult to measure and are therefore subject to lower accuracy;
- DAB constructive characteristic is such that at no-load small variations in control variable produce a very amplified effect over controlled variable. Although this characteristic is common

to all simulations (off-line and CHIL), in real-time this effect is further amplified by the non-ideality of the generated gate pulses.

Besides minor differences between off-line simulations and real-time CHIL test, it was possible to:

- verify the correct discretization of the off-line simulations in order to make a direct and consistent comparison with the CHIL test;
- confirm with CHIL simulation the proper implementation of discrete ADRC algorithm into the microcontroller board.

In the last analysis, real-time timing constraints have been assessed during multiple runs of simulation cycles. According to Table III, timings of different control cycle computational phases are well below maximum allowed time. The detail of typical microcontroller cycle execution is reported here below:

Table III: Measured control cycle execution timing

	Input readings	**Control**	**Output conditioning**	**Idle**
Measure(μs)	*105*	*75*	*35*	*85*

Conclusion

This work addressed the feasible implementation on a general-purpose microcontroller board of an ADRC control for use in real-time CHIL simulations. ADRC has been used into an OPAL-RT simulator environment to regulate output voltage of a DAB converter and compensate heavy load variations.
Performances obtained from real-time CHIL simulations have been compared with the same configuration in off-line simulation environments (ATPDraw and Simulink).
The comparison is positive: the proposed control is sufficiently robust through the simulation in CHIL environment, and discretization process didn't introduce substantial changes to the algorithm studied in the ideal scheme of continuous time control.
This result represents a first step for the development of future real-time CHIL implementations of ADRC control for more complex circuit structures interconnecting different power electronics systems.

References

[1] Y. Gao, H. Zhao, K. Ma and X. Huo, "Control Design for Current Loop of PMSM Using A Modified Model-Compensation ADRC Controller," *2020 39th Chinese Control Conference (CCC),* p. 2545–2550, 2020.

[2] F. Deng and Y. Guan, "PMSM Vector Control Based on Improved ADRC," *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE),* p. 154–158, 2018.

[3] L. E. I. Yang, X. U. Jing, M. Shuang, H. A. O. Qiang and S. Yue, "A Comparative Study of The First Order Linear ADRC and PI Controller in The Speed Control System of Permanent Magnet Synchronous Motor," *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS),* p. 498–503, 2020.

[4] H. Magnago, H. Figueira, O. Gagrica and D. Majstorovic, "HIL-based certification for converter controllers: Advantages, challenges and outlooks (Invited Paper)," *2021 21st International Symposium on Power Electronics (Ee),* p. 1–6, 2021.

[5] J. Han, "From PID to Active Disturbance Rejection Control," *IEEE Transactions on Industrial Electronics,* vol. 56, no. 3, pp. 900-906, 2009.

[6] G. Herbst, "A Simulative Study on Active Distrubance Rejection Control (ADRC) a control tool for Practitioners," *Electronics,* vol. 2, no. 3, pp. 246-279, Aug. 2013.

[7] B. Sun; Z. Gao, "A DSP-Based Active Disturbance Rejection Control Design for a 1-kW H-Bridge DC–DC Power Converter," *IEEE Transactions on Industrial Electronics,* vol. 52, no. 5, pp. 1271-1277, October 2005. doi:10.1109/TIE.2005.855679..

[8] Y. X. Su; C. H. Zheng; B. Y. Duan, "Automatic disturbances rejection controller for precise motion control," *IEEE Transactions on Industrial Electronics,* vol. 52, no. 3, p. 814–823, 2005. doi:10.1109/TIE.2005.847583.

[9] Q. Zheng; Z. Gao, "On practical applications of active disturbance rejection control," in *In Proceedings of the 29th Chinese Control Conference*, 2010.

[10] R. Mandonski; M. Nowicki; P. Herman, "Application of active distrubance rejection controller to water supply system.," in *In Proceedings of the 33rd Chinese Control Conference*, Nanjing, China, 2014.

[11] A. Oppenheim, Discrete Time Signal Processing Third Edition, Pearson, Ed., 2010.

[12] A. Clerici, R. Chiumeo and C. Gandolfi, "Real Time Control Hardware in The Loop test of a novel MVDC solid-state breaker," in *Proc. 22nd European Conf. Power Electronics and Applications (EPE'20 ECCE Europe)*, 2020.

[13] A. Clerici, R. Chiumeo, D. Raggini and A. Veroni, "Digital strategy to compensate offset currents into DC-DC Dual Active Bridge converter simulations," *2021 IEEE Fourth International Conference on DC Microgrids (ICDCM),* p. 1–6, July 2021.

[14] ST Microelectronics, "STM32F765xx - STM32F767xx - STM32F768Ax - STM32F769xx Datasheet - production data," DS11532 Rev 7, 2021.

[15] O. Yade, J.-Y. Gauthier, X. Lin-Shi, M. Gendrin and A. Zaoui, "Modulation strategy for a Dual Active Bridge converter using Model Predictive Control," in *2015 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, Valparaiso, 2015.