electricity

MDPI

Article

Modeling the Dynamics of Supercapacitors by Means of Riemann–Liouville Integral Definition

Ventura Avila-Rodriguez [1,†], Federico Leon-Zerpa [2,*], Jose Juan Quintana-Hernandez [3] and Alejandro Ramos-Martin [2,†]

[1] University Institute of Intelligent Systems and Numeric Applications in Engineering, University of Las Palmas de Gran Canaria, 35017 Las Palmas, Spain; ventura.avila101@alu.ulpgc.es
[2] Department of Process Engineering, University of Las Palmas de Gran Canaria, 35017 Las Palmas, Spain; alejandro.ramos@ulpgc.es
[3] Department of Electronic and Automatic Engineering, University of Las Palmas de Gran Canaria, 35017 Las Palmas, Spain; josejuan.quintana@ulpgc.es
* Correspondence: federico.leon@ulpgc.es
† These authors contributed equally to this work.

Abstract: The application of fractional calculus to obtain dynamic models for supercapacitors represents an alternative approach to obtaining simpler and more accurate models. This paper presents a model for the supercapacitor in the time domain, based on the use of the fractional or non-integer order integral. This fractional model is compared with the conventional simple model, which is typically used in industrial applications. This fractional integral-based model provides satisfactory fits in relation to the number of parameters used in the model. Furthermore, an interpretation of the effect of the application of fractional integration is presented for constant current charging and discharging processes at constant current, using the Riemann–Liouville definition for the non-integer order integral. Supercapacitors are devices that exhibit non-linear behavior, with a distinct charging and discharging operation. There are several methods of dynamic analysis for the characterization of supercapacitors. The information extracted from these methods is essential for understanding the behavior of supercapacitors and, thus, ensuring that processes involving supercapacitors are as efficient as possible. This paper presents a dynamic analysis based on charge and discharge operations with constant currents. The conclusion is that the fractional model provides fairly accurate fits.

Keywords: supercapacitor; modeling; fractional calculus; Riemann–Liouville integral

Citation: Avila-Rodriguez, V.; Leon-Zerpa, F.; Quintana-Hernandez, J.J.; Ramos-Martin, A. Modeling the Dynamics of Supercapacitors by Means of Riemann–Liouville Integral Definition. *Electricity* **2024**, *5*, 491–525. https://doi.org/10.3390/electricity5030025

Academic Editor: Jose Luis Rueda

Received: 21 June 2024
Revised: 26 July 2024
Accepted: 8 August 2024
Published: 13 August 2024

1. Introduction

The supercapacitor is a device that has emerged with the potential to improve energy storage systems [1,2]. They behave in a similar way to conventional capacitors, but have a larger electrode contact surface, which means higher capacitances. This leads to higher energy densities than conventional capacitors, and they are also characterized by higher power densities than batteries.

Supercapacitors typically have energy densities in the order of 300 times higher than conventional capacitors, and two-tenths that of batteries with lower energy densities. However, their power densities are 10 times higher than most batteries. Thanks to their energy and power densities, together with the associated low series resistance, they are an intermediate between batteries and conventional capacitors.

Supercapacitors are also known as ultracapacitors. Energy is stored by polarizing an electrolyte solution, although it is an electrochemical device in which virtually no chemical reactions occur in its energy storage mechanism. This mechanism is highly reversible, allowing the supercapacitors to be charged and discharged hundreds of thousands of times.

Supercapacitors are employed in automotive applications, serving as supplementary energy storage systems [3,4], hybrid energy systems [5–8], DVD players, computers, and other electronic equipment.

Supercapacitors are elements that exhibit non-linear behavior and undergo distinct charging and discharging processes. Obtaining accurate models for supercapacitors is crucial, as this enables the attainment of high efficiencies in their utilization in diverse industrial applications. This paper presents an analysis of the dynamics of supercapacitors for charging and discharging operations with constant currents.

The objective is to obtain simple models of the dynamics of electrochemical capacitors by using the non-integer or fractional order integral, which offer better fits than conventional models.

This paper is structured as follows: Section 2 provides a brief introduction to the applications of fractional calculus; Section 3 presents a brief description of supercapacitors; Section 4 presents the models analyzed in this article for the supercapacitors, and presents the identification method in the time domain to obtain the model parameters; Section 5 shows the materials and methods; Section 6 shows the results of the tests carried out; and, finally, Section 7 shows the conclusions.

2. Introduction to the Application of Fractional Calculus

The concept of the non-integer derivative is as old as conventional differential calculus. Fractional calculus has been employed to model a range of physical phenomena [9]. Additionally, it has been utilized in the field of control theory [10,11].

In nature, numerous systems exhibit fractional behavior, although many of these systems exhibit only slight fractionality. In this context, multi-problem solving techniques are proposed in [12]. Furthermore, the modeling of anomalous diffusion phenomena, defined in accordance with the fractional Fokker–Planck equation [13].

There are numerous real-world applications of fractional calculus, including the control of power converters [14], the modeling of mechanical systems [15,16], the description of viscoelastic materials using fractional models [17], the study of dielectrics [18], and electrochemical identification techniques (fruits, vegetables, fuel cells, batteries, supercapacitors) [19–24].

3. Supercapacitor Features

A supercapacitor has two porous, non-reactive electrodes immersed in an electrolyte, with a separator between the electrodes that allows ions to move across it.

Energy is stored in the form of separate charges in the electrochemical double layer formed at the interface. The thickness of the double layer depends on the concentration of the electrolyte and the size of the ions, and the capacitance values are contingent upon the ratio of ion sizes between the molecules [25]. On the other hand, the electrodes are made of porous materials with a high surface area, the size of these pores being of the order of nanometers, giving specific surface areas between 500 and 2000 $\frac{m^2}{g}$, and specific capacities for carbon electrodes of 75–175 F/g for aqueous electrolytes and 40–100 F/g for organic electrolytes [1,2].

Ions are transferred between the electrodes during charging and discharging by diffusion through the electrolyte.

4. Dynamic Characteristics of Supercapacitors

Figure 1 illustrates the voltage evolution of an electrochemical capacitor (Panasonic 50 F) when a constant current is applied (at the top charge current, $i_{charge} = 3$ A, and at the bottom discharge current, $i_{discharge} = -3$ A). It can be observed that the response to the constant current is not linear, as would be expected for a capacitor. This non-linear behavior can be observed in the non-proportional variation of the voltage with respect to the integral of the current. It is also important to note that the behavior of the electrochemical capacitor for charging differs from that for discharging.

A variety of dynamic models for electrochemical capacitors have been employed in a number of industrial applications. These include simple *RC* models based on circuits with an equivalent series resistor and an ideal capacitor [26–31], as well as models composed of *RC* networks of resistors and capacitors [32,33], and finally models based on variable capacitances (voltage-dependent $C(v(t))$), which involve differential equations with variable coefficients [34,35].

This paper proposes an alternative model based on the non-integer or fractional order integral, which achieves a satisfactory fit of the data in relation to the number of parameters used in the model. Furthermore, the impact of fractional integration on the electrochemical capacitor's behavior when charged or discharged at a constant current is elucidated, employing the definition of the Riemann–Liouville integral.

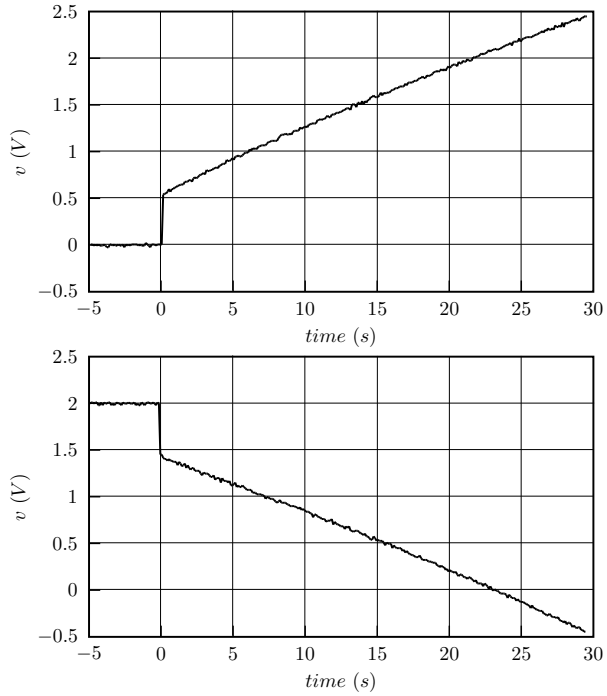

Figure 1. Supercapacitor behavior for charging and discharging operations.

4.1. Fractional Modeling

From Figure 1 and the typical response of the simple *RC* capacitor model, it can be established that the behavior of the electrochemical capacitor at constant current is non-linear and differs from that of the conventional model.

Figures 2 and 3 illustrate the simulation of the conventional *RC* model in comparison to the fractional model, for constant charge and discharge profiles. The analysis of these figures indicates that the following expression accurately represents the response of the supercapacitor:

$$\Delta v(t) = R\, i(t) + \Delta v_c(t) \tag{1}$$

The behavior of $\Delta v_c(t)$ differs between charging and discharging operations, and there is no linear relationship between the integral of the current $i(t)$ and the aforementioned variable. Consequently, the following expressions may be considered for the charging and discharging operations:

$$v(t) = v(0) + R_\alpha\, i(t) + \Delta v_{c\alpha}(t), \quad i(t) > 0 \tag{2}$$

$$v(t) = v(0) + R_\beta\, i(t) + \Delta v_{c\beta}(t), \quad i(t) < 0 \tag{3}$$

In expressions (2) and (3), the parameters R_α and R_β are equivalent series resistances. Conversely, the terms $\Delta v_{c\alpha}(t)$ and $\Delta v_{c\beta}(t)$ are linked to the fluctuations in the stored electric charge, which consequently influence the current $i(t)$. These terms exhibit a linear relationship in the conventional model (RC), as a uniform storage of the electric charge in the capacitor results in a uniform variation in the voltage in the capacitor. However, in the case of the supercapacitor, the storage of the electric charge and the voltage variation are not uniform. That is to say, identical amounts of electric charge do not cause the same voltage variation.

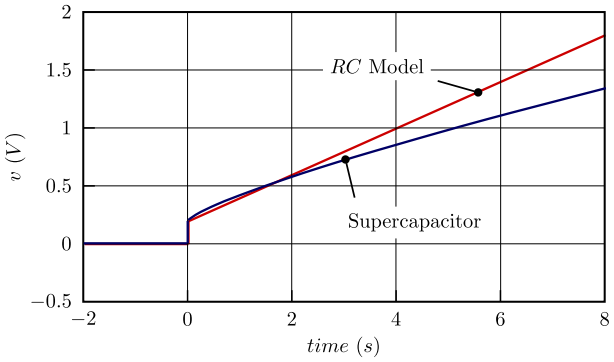

Figure 2. Supercapacitor response to constant charge current.

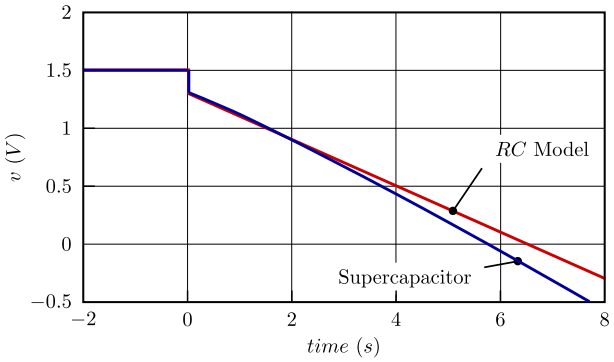

Figure 3. Supercapacitor response to constant discharge current.

This paper proposes the use of the non-integer order integral of the current ($i(t)$) to associate it with the non-uniform behavior of the voltage variation with respect to a constant current. Consequently, the following linear relationship is demonstrated below, as a consequence of the utilization of the non-integer order integral.

$$\Delta v_{c\alpha}(t) = \frac{\Delta q_\alpha(t)}{C_\alpha} = \frac{{}_0I_t^\alpha\, i(t)}{C_\alpha} \tag{4}$$

$$\Delta v_{c\beta}(t) = \frac{\Delta q_\beta(t)}{C_\beta} = \frac{{}_0I_t^\beta\, i(t)}{C_\beta} \tag{5}$$

where C_α and C_β are constant.

Using non-integer order integral, expressions (2) and (3) are in the following forms:

$$v(t) = v(0) + R_\alpha\, i(t) + \frac{1}{C_\alpha} i^{(-\alpha)}(t), \quad i(t) > 0 \tag{6}$$

$$v(t) = v(0) + R_\beta\, i(t) + \frac{1}{C_\beta} i^{(-\beta)}(t), \quad i(t) < 0 \tag{7}$$

The non-integer order integral has been employed to derive a non-integer order equation with constant coefficients (R_α, $\frac{1}{C_\alpha}$, R_β, $\frac{1}{C_\beta}$), which accounts for the charging and discharging behavior of the supercapacitor.

Given that the fractional integral of a constant is [9],

$$ {}_0I_t^\alpha \; C = \frac{C}{\Gamma(1+\alpha)} t^\alpha \tag{8} $$

it is possible to solve the expressions (6) and (7) as follows:

- For a constant current greater than zero ($i(t) = i > 0$):

$$ v(t) = v(0) + R_\alpha \, i(t) + \frac{1}{C_\alpha} \frac{i}{\Gamma(1+\alpha)} \, t^\alpha, \quad i(t) > 0 \tag{9} $$

- For a constant current less than zero ($i(t) = i < 0$):

$$ v(t) = v(0) + R_\beta \, i(t) + \frac{1}{C_\beta} \frac{i}{\Gamma(1+\beta)} \, t^\beta, \quad i(t) < 0 \tag{10} $$

The Riemann–Liouville definition of the non-integer order integral provides insight into the derivation of the linear relationship through the use of the non-integer order integral. The non-integer order integral of the current, as defined by the Riemann–Liouville definition, is presented below.

$$ {}_0I_t^\alpha \; i(t) = \frac{1}{\Gamma(\alpha)} \int_0^t i(\tau)(t-\tau)^{(\alpha-1)} d\tau, \quad t > 0, \alpha \in \mathbb{R}^+ \tag{11} $$

The Riemann–Liouville integral is a convolution, which by itself does not give much information about the behavior of the supercapacitor, but if a transformation is applied on the time scale such as the following [36]:

$$ g_t(\tau) = \frac{1}{\Gamma(\alpha+1)} [t^\alpha - (t-\tau)^\alpha] \tag{12} $$

it is possible to arrive at the following integral:

$$ {}_0I_t^\alpha \; i(t) = \int_0^t i(\tau) dg_t(\tau) \tag{13} $$

The difference between the integrals (11) and (13) can be seen in Figure 4, obtained from simulations, with the upper part showing the voltage $v(\tau)$ versus $g_t(\tau)$, on the conventional time scale, and the lower part showing the voltage $v(\tau)$ versus $g_t(\tau)$, on a transformed time scale. The latter curve shows a linear relationship between the integral of the current $i(t)$, and the variation of the voltage $\Delta v_{calpha}(t)$.

In summary, it can be stated that the non-integer order integral enables the construction of a model that can provide fits to real data with reduced error, with a small number of parameters, for the purposes of charging and discharging at a constant current.

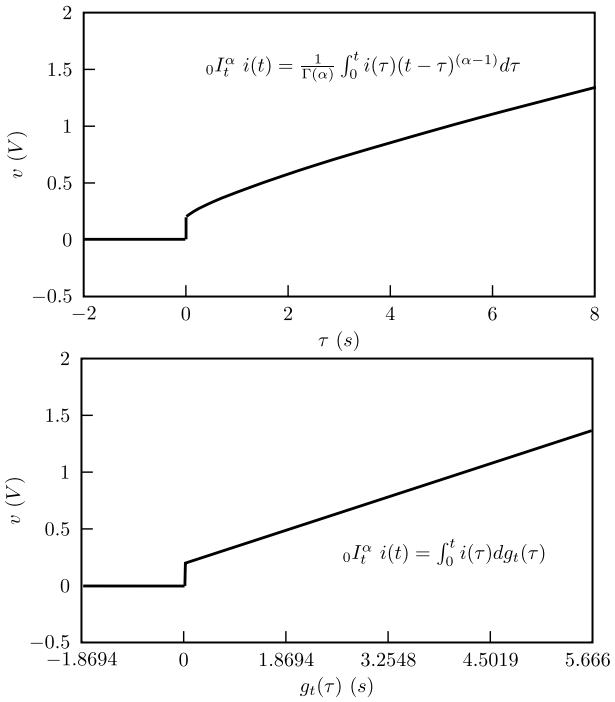

Figure 4. Voltage versus τ and $g_t(\tau)$.

4.2. Model Validation Method

For the determination of the parameters of the supercapacitor (R_α, C_α, R_β, C_β), the procedure developed in article [37] is used. To carry out the identification, the Grünwald–Letnikov approximation has been made use of for the non-integer order derivative [9] to obtain a numerical solution of the fractional integral (6) and (7), using the following expression:

$$_0I_t^\alpha f(t) \approx h^\alpha \sum_{j=0}^{k} b_j\, f(kh - jh) \tag{14}$$

$$b_j = (-1)^j \binom{-\alpha}{j} \tag{15}$$

where $\binom{-\alpha}{j}$ is a binomial coefficient. To determine the coefficients b_j, use of the following approach is proposed [9]:

$$b_j = \left(1 - \frac{1-\alpha}{j}\right) b_{j-1},\ \text{for } j = 1, 2, 3, \ldots N;\ \ b_0 = 1 \tag{16}$$

The numerical approximation of Equation (6) is shown below, with a discreet time t_m $(m = 2, 3, \ldots)$,

$$\Delta v_m = R_\alpha\, i_m + \frac{1}{C_\alpha}\, i_m^{(-\alpha)} \tag{17}$$

Only the charging operation is shown, because the numerical approximation is identical for the case of the discharging operation. Applying the Grünwald–Letnikov approximation, we obtain

$$\Delta v_m = R_\alpha\, i_m + \frac{1}{C_\alpha}\, h^\alpha \sum_{j=0}^{m} b_j\, i_{m-j} \tag{18}$$

In order to achieve identification with the fractional model, the following functional has been used:

$$E(\bar{a}) = \int_0^T [a_1\, i^{(-\alpha)}(t) + a_0\, i(t) - \Delta v(t)]^2 dt \approx min \tag{19}$$

With this functional, the coefficients of the equation can be determined from the experimental data $(i(t),\ v(t))$. The necessary condition to obtain the minimum of this functional is

$$\frac{\partial E(\bar{a})}{\partial \bar{a}} = 0 \tag{20}$$

If the continuous functions $\Delta v(t)$ and $i(t)$ are replaced by their discrete equivalents, furthermore, the integrals are replaced by summations for the time interval of operation, which remains unchanged, like in (21).

$$\begin{aligned} a_1 \sum_{m=0}^{M} (i_m^{(-\alpha)})^2 + a_0 \sum_{m=0}^{M} i_m\, i_m^{(-\alpha)} &= \sum_{m=0}^{M} \Delta v_m\, i_m^{(-\alpha)} \\ a_1 \sum_{m=0}^{M} i_m^{(-\alpha)}\, i_m + a_0 \sum_{m=0}^{M} i_m^2 &= \sum_{m=0}^{M} \Delta v_m\, i_m \end{aligned} \tag{21}$$

The parameters $\bar{a}$ can be determined from the system of Equation (21) by means of an iterative process of calculation. In order to obtain these parameters, and also the non-integer order of the integral α, it is necessary to carry out this process. This is due to the fact that the system of equations given in Equation (21) is non-linear. The iterative method is developed in greater detail in the aforementioned paper [37].

5. Materials and Methods

In order to ascertain the viability of the proposed model, it is necessary to conduct a series of laboratory tests, based on the assumptions made during its construction. This section presents the materials and methods developed in this work, as well as the implementation of the programming codes necessary for the development of the experimentation.

5.1. Materials

Figure 5 illustrates a basic diagram of the experimental design. The system comprises the following elements:

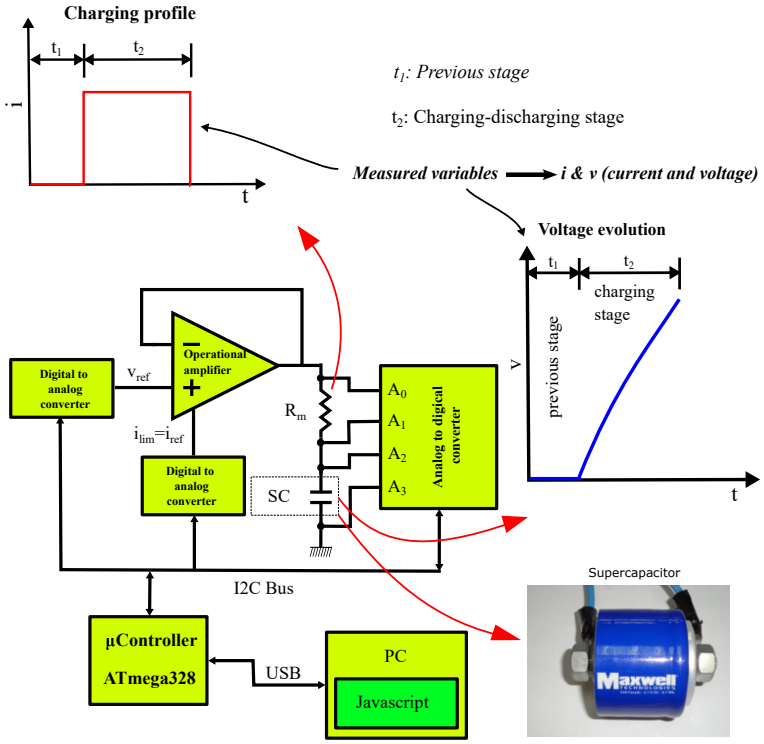

Figure 5. Basic diagram of the experimental design.

1. An operational amplifier OPA549 (Texas Instruments, Dallas, TX, USA). This operational amplifier can provide a nominal current of 8 A, plus it has a special input (i_{lim}) to limit the output current of the amplifier. This is an important feature for the implementation of galvanostatic charge and discharge processes by controlling the current.
2. An analog-to-digital conversion stage (Figure 6) based on the precision converter ADS1115 (Texas Instruments), which is mounted on a board from Adafruit Industries (New York, NY, USA). This stage has two analogue channels in the differential mode, for the measurement of charge–discharge current and supercapacitor voltage. The current $i(t)$ is measured from the voltage of a resistor in series with the supercapacitor $R_m = 0.1\ \Omega$, and the voltage is measured directly $v(t)$. The ADS1115 circuit has a resolution of 16 Bit, giving a resolution of 3 mV for the voltage measurement and 30 mA for the current.
3. Two digital-to-analog conversion stages (Figure 7) based on a 12 Bit resolution MCP4725 (Microchip, Chandler, AZ, USA), which are mounted on a board from Adafruit Industries. These stages are used to control the charge–discharge processes of the supercapacitor, and to adjust the voltage and current references.
4. A microcontroller model ATmega328 (Microchip) implemented on an embedded system, which is called Arduino Nano (Figure 8). This microcontroller is responsible for controlling the galvanostatic charging and discharging processes, as well as capturing data and sending it, via USB, to the PC to be saved. The microcontroller is responsible for adjusting the output voltage of the two digital-to-analogue converters: the first to set the target voltage to be applied to the output of the operational amplifier, which operates in a voltage follower configuration, and the second digital-to-analogue converter to set the charge or discharge current of the supercapacitor, thus achieving galvanostatic operation of the system. Therefore, by using the indicated devices, it is possible to implement a galvanostat that meets the requirements of the experimental design of this article, without the need for other devices.
5. A PC, which is responsible for controlling and configuring the microcontroller for the designed tests, as well as for storing the captured data obtained from the measurements. This is achieved by means of a program implemented in JavaScript language, under the Processing environment, Processing Foundation. This program is an interface to communicate and control the embedded system (Arduino Nano) with the PC. Processing is a flexible, open source, free, cross-platform application. There is a lot of information on the internet for programming on this platform.

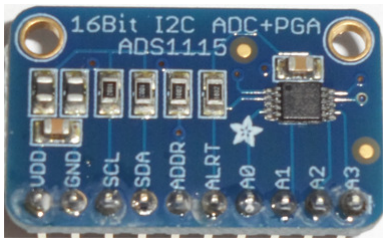

Figure 6. Analog-to-digital converter ADS1115.

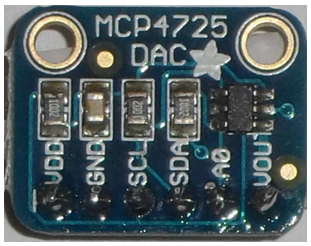

Figure 7. Digital-to-analog converter MCP4725.

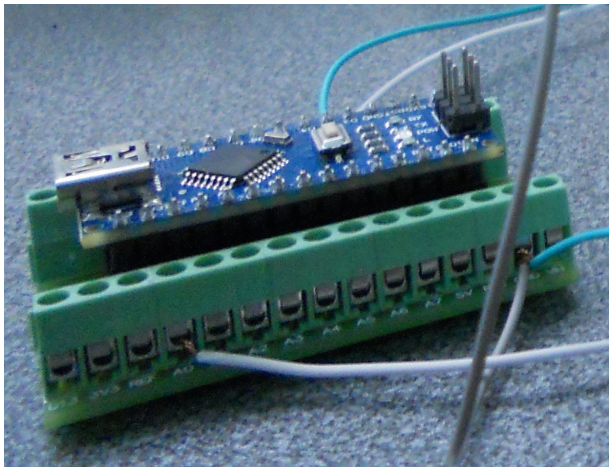

Figure 8. Arduino Nano.

Finally, a system photograph is displayed in Figure 9.

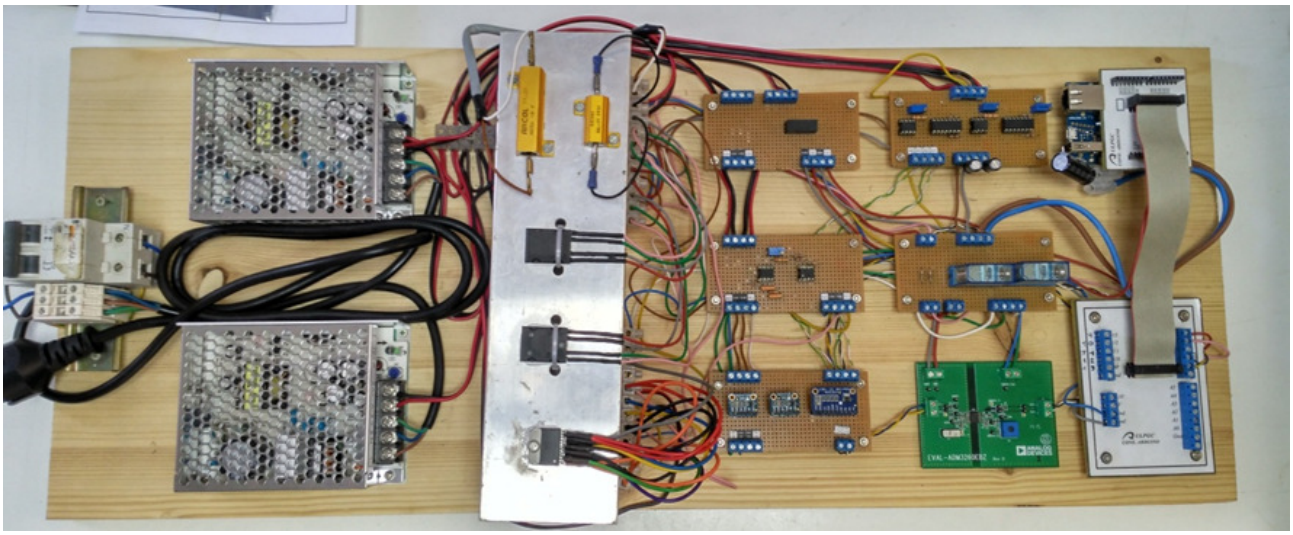

Figure 9. Photograph of the experimental design.

5.2. Microcontroller IDE and Interface GUI

This work has used the Arduino IDE to develop and program the microcontroller indicated above, as can be seen in Figure 10. The source code used is shown in Appendix A.

On the other hand, for communication, test management and storage of experimental data, an interface application has been implemented and developed in Processing, an image of which is shown in Figure 11. The source code used is shown in Appendix B. Processing is a flexible software sketchbook and a language for learning how to code. Processing is based on the Java programming language, although it is also possible to code in other languages, including JavaScript, Android, and Python, among others.

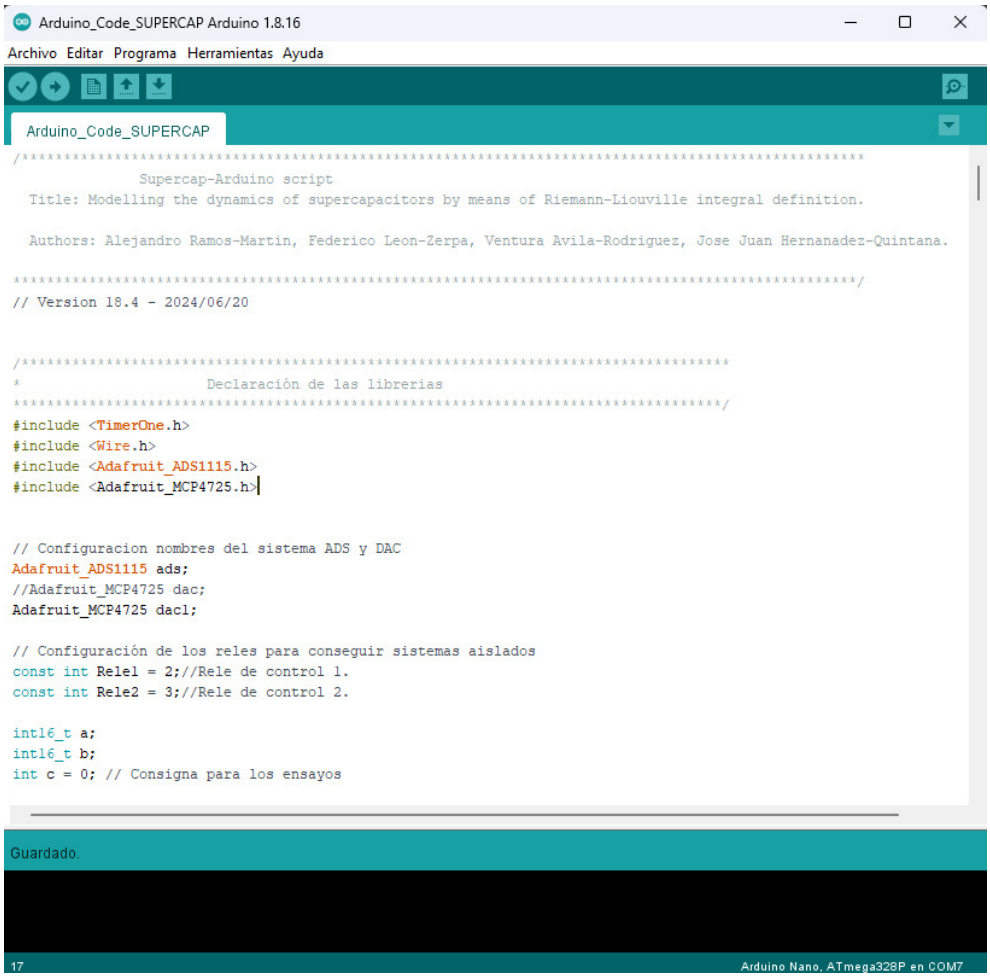

Figure 10. Microcontroller IDE.

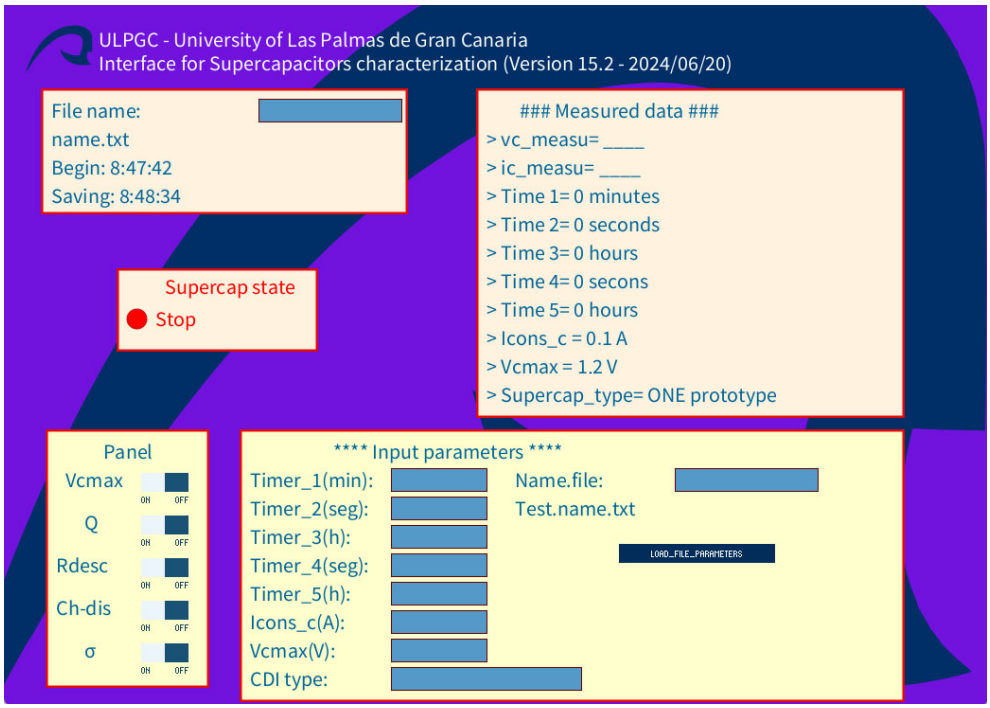

Figure 11. Interface Processing GUI.

6. Results

The following are the charging and discharging tests performed on a 50 F supercapacitor. Figure 12 shows the charging results, applying a constant current of 3 A. The upper curve in Figure 12 shows the actual response of the supercapacitor on a conventional time scale τ, as well as the results from the fractional model. The parameter values for this capacitor are shown in Table 1, which have been obtained from the identification process. From these parameters, the following expression obtained from (9) is presented:

$$\begin{aligned} v(t) &= 0 + 0.1661 \cdot 3 + \frac{1}{29.6736} \frac{3}{\Gamma(1+0.8575)} \, t^{0.8575} \\ v(t) &= 0.4983 + 0.1067 \, t^{0.8575} \end{aligned} \tag{22}$$

The results of the fractional model, expression (22), are plotted in Figure 12 together with the experimental results.

In the lower curve of Figure 12, the experimental response is shown with a transformed time scale $g_t(\tau)$, determined from the identification.

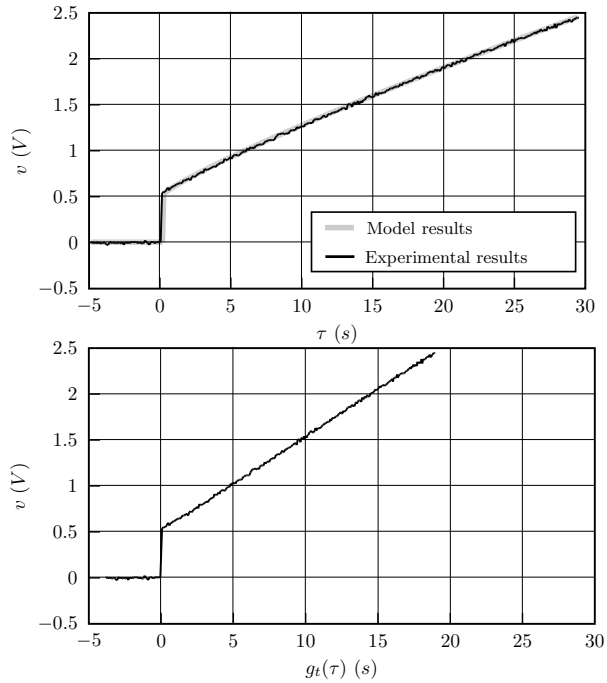

Figure 12. Charging operation, conventional time scale τ versus transformed time scale $g_t(\tau)$.

Table 1. Charging operation parameters.

R_α	C_α	α	Error
0.1661	29.6736	0.8575	0.00009

Figure 13 shows the discharge results, applying a constant current of −3 A. In the upper curves of the figure, the actual response of the capacitor on the conventional time scale τ is shown, as well as the results of the fractional model. The parameter values for this discharge process are presented in Table 2. Based on these parameters, the following expression (10) obtained from the table is given:

$$\begin{aligned} v(t) &= 2 - 0.1997 \cdot 3 - \frac{1}{64.9350} \frac{3}{\Gamma(1+1.0975)} \, t^{1.0975} \\ v(t) &= 2 - 0.5991 - 0.0442 \, t^{1.0975} \end{aligned} \tag{23}$$

The results of the fractional model, expression (23), are plotted in Figure 13, together with the experimental results.

The lower curve in Figure 13 shows the actual response with a transformed time scale $g_t(\tau)$.

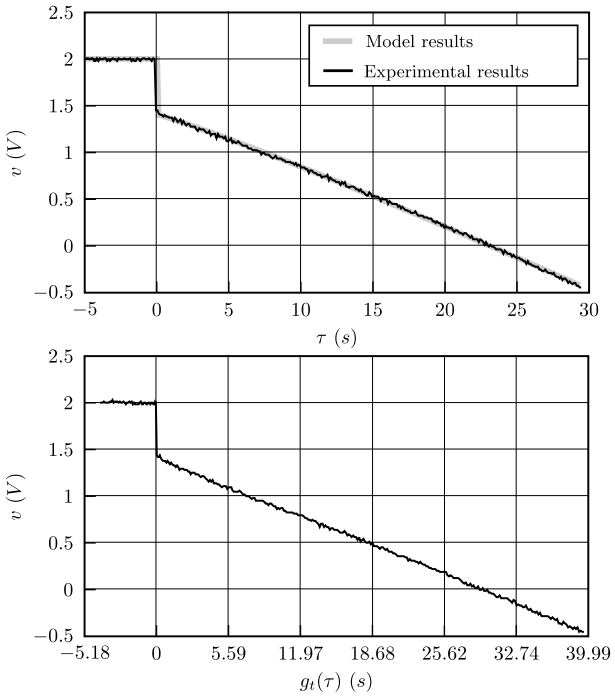

Figure 13. Discharge operation, conventional time scale τ versus transformed time scale $g_t(\tau)$.

Table 2. Discharge operation parameters.

R_β	C_β	β	Error
0.1997	64.9350	1.0975	0.00024

From the results of Tables 1 and 2, it can be seen that the error of the fit of the data are very small, because the model obtained from the parameters practically coincides with the experimental data (Figures 12 and 13).

7. Conclusions

This paper presents a mathematical model that accurately predicts the dynamic response of electrochemical capacitors to constant current charging and discharging. Furthermore, the effect of employing non-integer integration is elucidated through the use of the Riemann–Liouville integral. It can be concluded that the fractional model fits the data much better than the conventional (RC) models.

Author Contributions: Conceptualization, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; methodology, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; software, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; validation, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; formal analysis, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; investigation, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; resources, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; data curation, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; writing—original draft preparation, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; writing—review and editing, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; visualization, V.A.-R., F.L.-Z., J.J.Q.-H. and A.R.-M.; supervision, J.J.Q.-H. and A.R.-M.; project administration, F.L.-Z. and A.R.-M.; funding acquisition, F.L.-Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by Project PID2022-138389OB-C32 Convocatoria de Generación de conocimiento 2022 Ministerio de Ciencia e Innovación del Gobierno de España (Ministry of Science and Innovation of the Spanish Government). This research was co-funded by Project PIE 2023-60 CONVOCATORIA DE PROYECTOS DE INNOVACIÓN EDUCATIVA 2023 de la Universidad de Las Palmas de Gran Canaria.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the availability of the technical resources in the laboratories of the University of Las Palmas de Gran Canaria, which were essential for the design and development of the instruments and devices necessary for the completion of this research project.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Microcontroller Source Code

```
/*************************************************************
              Supercap-Arduino script
  Title: Modelling the dynamics of supercapacitors by means of
Riemann-Liouville integral definition.

  Authors: Alejandro Ramos-Martin, Federico Leon-Zerpa,
Ventura Avila-Rodriguez, Jose Juan Hernanadez-Quintana.

*************************************************************/
// Version 18.4 - 2024/06/20


/*********************************
*    Declaration of the libraries
*********************************/
#include <TimerOne.h>
#include <Wire.h>
#include <Adafruit_ADS1115.h>
#include <Adafruit_MCP4725.h>


// Configuration ADS and DAC system names
Adafruit_ADS1115 ads;
//Adafruit_MCP4725 dac;
Adafruit_MCP4725 dac1;

// Relay configuration for isolated systems
const int Rele1 = 2;//Rele de control 1.
const int Rele2 = 3;//Rele de control 2.

int16_t a;
int16_t b;
int c = 0; // Test assignment



unsigned long control_tiempos=0;
uint16_t Vcuenta1=0;
uint16_t Vcuenta2=2048;
uint16_t Icons=2048;
int16_t results;
int16_t results1;

/*****************************************
*Serial port and timer configuration
*****************************************/

void setup()
{
```

```
  Serial.begin(9600);

  pinMode(Rele1, OUTPUT);
  pinMode(Rele2, OUTPUT);
  digitalWrite(Rele1, HIGH);
  digitalWrite(Rele2, HIGH);
  Timer1.initialize(100000);
  Timer1.attachInterrupt(intu);
  dac1.begin(0x62);
  //dac.begin(0x63);
  ads.begin();

   ads.setGain(GAIN_ONE);

 /****************************************
  * DAC and relays configuration
  ****************************************/

  dac1.setVoltage(2048, false);
  digitalWrite(Rele1, HIGH);//Relay NC
  digitalWrite(Rele2, HIGH);//Relay NC

}

void intu()
{
  Serial.print(a);
  Serial.print(",");
  Serial.print(b);
  Serial.print(",");
  Serial.print(Icons);
  Serial.print(",");
  Serial.print(Vcmax);
  Serial.print(",");
  Serial.println(estado);

  control_tiempos=control_tiempos+1;

}
/*******************************************
*          Tests are carried out
*******************************************/

void loop(void)
{
  Timer1.detachInterrupt();
    switch (inicio){
    //Time to 0 seconds
case 0: //Wait or stop state.
        digitalWrite(Rele1, HIGH);//Relay NC.
        digitalWrite(Rele2, HIGH);//Relay NC.
        results1 = ads.readADC_Differential_0_1();
        results = ads.readADC_Differential_2_3();
        a = results1;
        b = results;
        control_tiempos=0;
        Timer1.attachInterrupt(intu);
      break;

case 1: //Charging with maximum voltage setpoint.
      switch (estado){
         case 1:    //Force precharge status.
          digitalWrite(Rele1, HIGH);//Open relay (NC)
          digitalWrite(Rele2, HIGH);//Open relay (NC)

          dac1.setVoltage(2048, false);
          results1 = ads.readADC_Differential_0_1();
          results = ads.readADC_Differential_2_3();
          a = results1;
```

```
          b = results;
          if(control_tiempos>=tiempo_1){
              estado=2;
              control_tiempos=0;
          }
          Timer1.attachInterrupt(intu);
        break;

        case 2:    //Forced state of charge only with voltage limit.

         digitalWrite(Rele1, LOW);//Open relay (NA).
         digitalWrite(Rele2, LOW);//Open relay (NA).


         dac1.setVoltage(Icons, false);



         results1 = ads.readADC_Differential_0_1();
         results = ads.readADC_Differential_2_3();
          a = results1;
          b = results;

         if (a > Vcmax) {
            estado=3;//Self-discharge status
            control_tiempos=0;

            digitalWrite(Rele1, HIGH);//Relay NC.
            digitalWrite(Rele2, HIGH);//Relay NC.

            // Corriente
            dac1.setVoltage(2048, false);

          }

          Timer1.attachInterrupt(intu);
        break;

        case 3://Force self-discharge status.
          results1 = ads.readADC_Differential_0_1();
          results = ads.readADC_Differential_2_3();
            a = results1;
            b = results;

          if(control_tiempos>=tiempo_3){

              estado=4;
              inicio=0;
              control_tiempos=0;
          }

          Timer1.attachInterrupt(intu);
        break;

        }
       //End switch (estado)
    break;

case 2://Charging with electrical charge setpoint.
      switch (estado){
          case 1:    //Force precharge status.

            digitalWrite(Rele1, HIGH);//Relay NC.
            digitalWrite(Rele2, HIGH);//Relay NC.

            dac1.setVoltage(2048, false);

           results1 = ads.readADC_Differential_0_1();
           results = ads.readADC_Differential_2_3();
```

```
      a = results1;
      b = results;

    if(control_tiempos>=tiempo_1){
        estado=2;
        control_tiempos=0;
    }
    Timer1.attachInterrupt(intu);
    break;

   case 2://Force state of charge with time limit 2.

    digitalWrite(Rele1, HIGH);//Relay NC.
    digitalWrite(Rele2, HIGH);//Relay NC.


    dac1.setVoltage(Icons, false);


    results1 = ads.readADC_Differential_0_1();
    results = ads.readADC_Differential_2_3();
    a = results1;
    b = results;

    if (a > Vcmax) {
      estado=3;// Self-discharge status
      control_tiempos=0;

      digitalWrite(Rele1, LOW);//Relay NA.
      digitalWrite(Rele2, LOW);//Relay NA.

      dac1.setVoltage(2048, false);

    }

    if(control_tiempos>=tiempo_2){
        estado=3;// Self-discharge status
        control_tiempos=0;

        digitalWrite(Rele1, HIGH);//Relay NC.
        digitalWrite(Rele2, HIGH);//Relay NC.

        dac1.setVoltage(2048, false);

    }
    Timer1.attachInterrupt(intu);
  break;

    case 3:// Force self-discharge status.

        results1 = ads.readADC_Differential_0_1();
        results = ads.readADC_Differential_2_3();

        a = results1;
        b = results;

      if(control_tiempos>=tiempo_3){
          estado=4;// Test completed
          inicio=0;
          control_tiempos=0;
      }

    Timer1.attachInterrupt(intu);

    break;
}//Fin switch (estado)
  break;
```

```
case 3:// Indefinite discharge is forced with discharge resistance.

            digitalWrite(Rele1, HIGH);//Relay NC.
            digitalWrite(Rele2, HIGH);//Relay NC.


            dac1.setVoltage(2048, false);


              results1 = ads.readADC_Differential_0_1();
              results = ads.readADC_Differential_2_3();
              a = results1;
              b = results;

               control_tiempos=0;
               Timer1.attachInterrupt(intu);

          break;

case 4:// CDI charging and discharging
      switch (estado){
         case 1:    //Force precharge status.
          digitalWrite(Rele1, HIGH);//Open relay (NC)
          digitalWrite(Rele2, HIGH);//open relay (NC)

          dac1.setVoltage(2048, false);
          results1 = ads.readADC_Differential_0_1();
          results = ads.readADC_Differential_2_3();
          a = results1;
          b = results;
          if(control_tiempos>=tiempo_1){
              estado=2;
              control_tiempos=0;
          }
          Timer1.attachInterrupt(intu);
        break;

        case 2:

         digitalWrite(Rele1, LOW);//Open relay (NA).
         digitalWrite(Rele2, LOW);//Open relay (NA).


         dac1.setVoltage(Icons, false);


         results1 = ads.readADC_Differential_0_1();
         results = ads.readADC_Differential_2_3();
          a = results1;
          b = results;

         if (a > Vcmax) {
            estado=3;//Self-discharge status
            control_tiempos=0;

            digitalWrite(Rele1, HIGH);//Relay NC.
            digitalWrite(Rele2, HIGH);//Relay NC.


            dac1.setVoltage(2048, false);
          }

          Timer1.attachInterrupt(intu); //The interruption of data sending is enabled.
        break;

        case 3:

          dac1.setVoltage(Icons, false);
```

```
         results1 = ads.readADC_Differential_0_1();
         results = ads.readADC_Differential_2_3();
          a = results1;
          b = results;

        break
        case 4:

         dac1.setVoltage(Icons, false);

         results1 = ads.readADC_Differential_0_1();
         results = ads.readADC_Differential_2_3();
          a = results1;
          b = results;

         if (a > Vcmax) {
            estado=3;
            control_tiempos=0;

            digitalWrite(Rele1, HIGH);
            digitalWrite(Rele2, HIGH);

            // Corriente
            dac1.setVoltage(2048, false);
          }

          Timer1.attachInterrupt(intu);
        break;

        case 5://Force self-discharge status.
          results1 = ads.readADC_Differential_0_1();
          results = ads.readADC_Differential_2_3();
            a = results1;
            b = results;
          if(control_tiempos>=tiempo_3){

              estado=4;//Test completed
              inicio=0;
              control_tiempos=0;
          }

          Timer1.attachInterrupt(intu);
        break;
    }

  }//End switch(inicio)

}//End void loop()


/*************************************************
*           Serial Event Stage
*************************************************/
  /*
   *   C ---> The order of the test
   *   Por ejemplo:
   *   Inicio --> 2
   *   Tiempo_1-> 50
   *   Tiempo_2-> 200
   *   Tiempo_3-> 50
   **  Tiempo_4-> 100
   **  Tiempo_5-> 100
   *   Icons  --> 1500
   *   Vcmax  --> 1000
   */

void serialEvent(){
```

```
    if (Serial.peek() == 'c') {
       Timer1.detachInterrupt();
       Serial.read();
       inicio = Serial.parseInt();
       tiempo_1 = Serial.parseInt();
       tiempo_2 = Serial.parseInt();
       tiempo_3 = Serial.parseInt();
       tiempo_4 = Serial.parseInt();
       tiempo_5 = Serial.parseInt();
       Icons = Serial.parseInt();
       Vcmax = Serial.parseInt();

         estado=1;
         control_tiempos=0;
         Timer1.attachInterrupt(intu);
     }
    while (Serial.available() > 0){
      Serial.read();
    }
}
```

Appendix B. Processing Source Code

```
/*********************************************************************************************
              Supercap-Processing script GUI
  Title: Modelling the dynamics of supercapacitors by means of
  Riemann-Liouville integral definition.

  Authors: Alejandro Ramos-Martin, Federico Leon-Zerpa,
         Ventura Avila-Rodriguez, Jose Juan Hernanadez-Quintana.

*********************************************************************************************/
// Version 15.2 - 2024/06/20

import processing.serial.*;
import controlP5.*;

  String  nom_archivo="name.txt",para_nom_archivo="name.txt",estado_CDI="Stop";
  String  hora_inicio="Begin: "+hour()+":"+minute()+":"+second();
  String  tiempo_1="0",tiempo_2="0",tiempo_3="0", tiempo_4="0", tiempo_5="0", tipo_CDI="ONE";
  String  I_c="0,1",V_c="1,2";
  String  tipo_ensayo="0";

  PrintWriter archivo;//Data file test pointer.
  PrintWriter fichero;//Parameters file pointer.

ControlP5 cp5;

Serial myPort;
  boolean serialInited;

   int muestras_segundo=10;
   int finalizado=0;
   int xPos = 1;
   int esquina_x=600;
   int esquina_y=100;
   int ancho=1200;
   int alto=600;
   int pocision_alto=0;
   int guardar_datos=-1,inicio=0;
   String dato_1="____",dato_2="____",dato_3="____",dato_4="____";
   String dato_5="____",dato_6="____";
   PShape bot;

// Button configuration

  int ancho_ventana=3020,alto_ventana=1500;
  int x_r1=500,y_r1=90,dx_r1=450,dy_r1=345;
```

```
  int mx_r1=10,my_r1=30,dy_t=30;
  int x_r2=40,y_r2=90,dx_r2=385,dy_r2=130;
  int mx_r2=10,my_r2=30;
  int ancho_campo_texto=210;
  int x_resto=40,y_resto=250,dy_resto=25;
  int x_r3=40,y_r3=300,dx_r3=210,dy_r3=35;
  int mx_r3=10,my_r3=30;
  int ancho_boton=110,alto_boton=20,ancho_mitad_boton=55;
  int x_r4=250,y_r4=450,dx_r4=700,dy_r4=285;
  int mx_r4=10,my_r4=30;
  int ancho_campo_texto_4=150;
  int x_r5=45,y_r5=450,dx_r5=170,dy_r5=270;
  int mx_r5=10,my_r5=30;
  int x_r6=120,y_r6=280,dx_r6=210,dy_r6=85;
  int mx_r6=10,my_r6=30;

  int lf = 10;
  int BAUD_RATE=9600;

  String inString=null;

void setup () {


  cp5 = new ControlP5(this);


  cp5.addButton("load_file_parameters")
     .setValue(0)
     .setPosition(x_r4+40*mx_r4,y_r4+4*my_r4)
     .setSize(ancho_boton+ancho_mitad_boton,alto_boton)
     ;


  cp5.addTextfield("nom_archivo")
     .setPosition(x_r2+2*mx_r2+ancho_campo_texto, y_r2+my_r2/3)
     .setSize(150,25)
     .setCaptionLabel("")
     .setColorBackground(color(84,153,199))
     .setColorActive(color(100,0,0))
     .setColorForeground(color(100,0,0))
     .setFont(createFont("times new roman",20))
     ;


  cp5.addTextfield("para_nom_archivo")
     .setPosition(x_r4+25*mx_r4+ancho_campo_texto, y_r4+4*my_r4/3)
     .setSize(150,25)
     .setCaptionLabel("")
     .setColorBackground(color(84,153,199))
     .setColorActive(color(100,0,0))
     .setColorForeground(color(100,0,0))
     .setFont(createFont("times new roman",20))
     ;


  cp5.addTextfield("tiempo_1")
     .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+dy_t/3)
     .setSize(100,25)
     .setCaptionLabel("")
     .setColorBackground(color(84,153,199))
     .setColorActive(color(100,0,0))
     .setColorForeground(color(100,0,0))
     .setFont(createFont("times new roman",20))
     ;


  cp5.addTextfield("tiempo_2")
     .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+4*dy_t/3)
```

```
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;


cp5.addTextfield("tiempo_3")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+7*dy_t/3)
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;

cp5.addTextfield("tiempo_4")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+10*dy_t/3)
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;

cp5.addTextfield("tiempo_5")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+13*dy_t/3)
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;


cp5.addTextfield("I_c")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+16*dy_t/3)
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;


cp5.addTextfield("V_c")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+19*dy_t/3)
    .setSize(100,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
    .setFont(createFont("times new roman",20))
    ;


cp5.addTextfield("tipo_CDI")
    .setPosition(x_r4+mx_r4+ancho_campo_texto_4, y_r4+my_r4+22*dy_t/3)
    .setSize(200,25)
    .setCaptionLabel("")
    .setColorBackground(color(84,153,199))
    .setColorActive(color(100,0,0))
    .setColorForeground(color(100,0,0))
```

```
    .setFont(createFont("times new roman",20))
    ;


  cp5.addToggle("tension_max")
    .setPosition(x_r5+10*mx_r5,y_r5+3*my_r5/2)
    .setSize(50,20)
    .setCaptionLabel("on                off")
    .setColorLabel(color(21,67,96))
    .setColorBackground(color(235,245,251))
    .setColorActive(color(26,82,118))
    .setValue(false)
    .setMode(ControlP5.SWITCH)
    ;


  cp5.addToggle("carga_culombios")
    .setPosition(x_r5+10*mx_r5,y_r5+6*my_r5/2)
    .setSize(50,20)
    .setCaptionLabel("on                off")
    .setColorLabel(color(21,67,96))
    .setColorBackground(color(235,245,251))
    .setColorActive(color(26,82,118))
    .setValue(false)
    .setMode(ControlP5.SWITCH)
    ;



  cp5.addToggle("descarga_resistencia")
    .setPosition(x_r5+10*mx_r5,y_r5+9*my_r5/2)
    .setSize(50,20)
    .setCaptionLabel("on                off")
    .setColorLabel(color(21,67,96))
    .setColorBackground(color(235,245,251))
    .setColorActive(color(26,82,118))
    .setValue(false)
    .setMode(ControlP5.SWITCH)
    ;


  cp5.addToggle("carga_descarga")
    .setPosition(x_r5+10*mx_r5,y_r5+12*my_r5/2)
    .setSize(50,20)
    .setCaptionLabel("on                off")
    .setColorLabel(color(21,67,96))
    .setColorBackground(color(235,245,251))
    .setColorActive(color(26,82,118))
    .setValue(false)
    .setMode(ControlP5.SWITCH)
    ;

  cp5.addToggle("conductimetro")
    .setPosition(x_r5+10*mx_r5,y_r5+15*my_r5/2)
    .setSize(50,20)
    .setCaptionLabel("on                off")
    .setColorLabel(color(21,67,96))
    .setColorBackground(color(235,245,251))
    .setColorActive(color(26,82,118))
    .setValue(false)
    .setMode(ControlP5.SWITCH)
    ;

/*************************************************************************
                Configuration window GUI
*************************************************************************/


  size(1040,740);
```

```
  stroke(227,10,10);

  println(Serial.list());


  myPort = new Serial(this, Serial.list()[0], 9600);
  myPort.clear();
  myPort.buffer(80);


  inString = myPort.readStringUntil(lf);
  inString = null;


  background(113,023,220);

  bot = loadShape("ULPGC.svg");
  shape(bot, 22, 22, 72, 52);
  shape(bot, -100,-100);
  color(26,82,118);
  String s_1="ULPGC - University of Las Palmas de Gran Canaria";
  String s_2="Interface for Supercapacitors characterization (Version 15.2 - 2024/06/20)";


  textSize(22);
  fill(248, 249, 249);
  text(s_1, 100, 45);
  text(s_2, 100, 70);

}

/*********************************************************************************
                        Test configuration
*********************************************************************************/

void tension_max(boolean theFlag) {
  if(theFlag==true) {
    fichero = createWriter("Ensayo."+nom_archivo);
    fichero.print("Test 1: ");
    fichero.println("Charge Vcmax y self-discharge");
    fichero.print("CDI type: ");
    fichero.println(tipo_CDI);
    fichero.print("Time 1: ");
    fichero.println(tiempo_1);
    fichero.print("Time 2: ");
    fichero.println(tiempo_2);
    fichero.print("Time 3: ");
    fichero.println(tiempo_3);
    fichero.print("Ic: ");
    fichero.println(I_c);
    fichero.print("Vcmax: ");
    fichero.println(V_c);
    fichero.flush();
    fichero.close();

// Test 1: with reference Vcmax and selfdischarge

    inicio = 1;

     String Vc_arduino=str(convierte_Vcmax_cuentas(V_c));
     String Ic_arduino=str(convierte_Icons_cuentas(I_c));
     String tiempo_3_arduino=str(convierte_tiempo_horas(tiempo_3));
     String tiempo_1_arduino=str(convierte_tiempo_minutos(tiempo_1));
     String tiempo_2_arduino=str(convierte_tiempo_segundos(tiempo_2));

     String i="";
     String t1="";
     String t2="";
     String t3="";
```

```
      String ic="";
      String vcmax="";

       i=str(inicio);
        t1=tiempo_1_arduino;
        t2=tiempo_2_arduino;
        t3=tiempo_3_arduino;
        ic=Ic_arduino;
        vcmax=Vc_arduino;


      tipo_ensayo="1"; // Maximum voltage test Vcmax

         myPort.write("c");
         myPort.write(i);
         myPort.write(",");
         myPort.write(t1);
         myPort.write(",");
         myPort.write(t2);
         myPort.write(",");
         myPort.write(t3);
         myPort.write(",");
         myPort.write(ic);
         myPort.write(",");
         myPort.write(vcmax);

    hora_inicio="Comienzo: "+hour()+":"+minute()+":"+second();

    if(archivo==null){

     }else{
      archivo.flush();
      archivo.close();
     }
       archivo = createWriter(nom_archivo);// a text file is created
       nom_archivo="I-"+nom_archivo;
      }else{

    //STOP de CDI

      String i=""+0;
      String t1=""+0;
      String t2=""+0;
      String t3=""+0;
      String ic=""+4095;
      String vcmax=""+1000;

    tipo_ensayo="0";

       myPort.write("c");
       myPort.write(i);
       myPort.write(",");
       myPort.write(t1);
       myPort.write(",");
       myPort.write(t2);
       myPort.write(",");
       myPort.write(t3);
       myPort.write(",");
       myPort.write(ic);
       myPort.write(",");
       myPort.write(vcmax);

    if(archivo==null){
      } else {
        archivo.flush();
        archivo.close();
      }
   }
}
```

```
void carga_culombios(boolean theFlag) {
  if(theFlag==true) {

    fichero = createWriter("parameters."+nom_archivo);
    fichero.print("Test 2: ");
    fichero.println("Current density");
   fichero.print("CDI type");
    fichero.println(tipo_CDI);
    fichero.print("Time 1: ");
    fichero.println(tiempo_1);
    fichero.print("Time 2: ");
    fichero.println(tiempo_2);
    fichero.print("Time 3: ");
    fichero.println(tiempo_3);
    fichero.print("Ic: ");
    fichero.println(I_c);
    fichero.print("Vcmax: ");
    fichero.println(V_c);
    fichero.flush();
    fichero.close();

    //Ensayo 2: Electric charge Q
    /*
    I=Coulombs/time ---> With t2 is another test for charge-discharge
    */

    inicio = 2;

     String Vc_arduino=str(convierte_Vcmax_cuentas(V_c));
     String Ic_arduino=str(convierte_Icons_cuentas(I_c));
     String tiempo_3_arduino=str(convierte_tiempo_horas(tiempo_3));
     String tiempo_1_arduino=str(convierte_tiempo_minutos(tiempo_1));
     String tiempo_2_arduino=str(convierte_tiempo_segundos(tiempo_2));

     String i=""+inicio;
     String t1=""+tiempo_1_arduino;
     String t2=""+tiempo_2_arduino;
     String t3=""+tiempo_3_arduino;
     String ic=""+Ic_arduino;
     String vcmax=""+Vc_arduino;


      tipo_ensayo="2";

        myPort.write("c");
        myPort.write(i);
        myPort.write(",");
        myPort.write(t1);
        myPort.write(",");
        myPort.write(t2);
        myPort.write(",");
        myPort.write(t3);
        myPort.write(",");
        myPort.write(ic);
        myPort.write(",");
        myPort.write(vcmax);

    hora_inicio="Begin: "+hour()+":"+minute()+":"+second();

    if(archivo==null){

    } else {
        archivo.flush();
        archivo.close();
       }
       archivo = createWriter(nom_archivo);
           nom_archivo="I-"+nom_archivo;
   } else {
```

```
      String i=""+0;
      String t1=""+0;
      String t2=""+0;
      String t3=""+0;
      String ic=""+4095;
      String vcmax=""+1000;

       tipo_ensayo="0";

         myPort.write("c");
         myPort.write(i);
         myPort.write(",");
         myPort.write(t1);
         myPort.write(",");
         myPort.write(t2);
         myPort.write(",");
         myPort.write(t3);
         myPort.write(",");
         myPort.write(ic);
         myPort.write(",");
         myPort.write(vcmax);

    if(archivo==null){

    } else {
       archivo.flush();
       archivo.close();
      }

    }
}


void descarga_resistencia(boolean theFlag) {
  if(theFlag==true) {



    inicio = 3;

      String i=""+inicio;
      String t1=""+tiempo_1;
      String t2=""+tiempo_2;
      String t3=""+tiempo_3;
      String ic=""+I_c;
      String vcmax=""+V_c;

      tipo_ensayo="3";

         myPort.write("c");
         myPort.write(i);
         myPort.write(",");
         myPort.write(t1);
         myPort.write(",");
         myPort.write(t2);
         myPort.write(",");
         myPort.write(t3);
         myPort.write(",");
         myPort.write(ic);
         myPort.write(",");
         myPort.write(vcmax);

    if(archivo==null){

    } else {
       archivo.flush();
       archivo.close();
    }
```

```
  } else {

     String i=""+0;
     String t1=""+0;
     String t2=""+0;
     String t3=""+0;
     String ic=""+4095;
     String vcmax=""+1000;

      tipo_ensayo="0";//CDI Stop

         myPort.write("c");
         myPort.write(i);
         myPort.write(",");
         myPort.write(t1);
         myPort.write(",");
         myPort.write(t2);
         myPort.write(",");
         myPort.write(t3);
         myPort.write(",");
         myPort.write(ic);
         myPort.write(",");
         myPort.write(vcmax);

   if(archivo==null){}
    else{
      archivo.flush();
      archivo.close();
    }

  }
}
void carga_descarga (boolean theFlag) {
  if(theFlag==true) {
    fichero = createWriter("test_4."+nom_archivo);
    fichero.print("Test 4: ");
    fichero.println("Charge and discharge of the CDI");
    fichero.print("CDI type: ");
    fichero.println(tipo_CDI);
    fichero.print("Time 1: ");
    fichero.println(tiempo_1);
    fichero.print("Time 2: ");
    fichero.println(tiempo_2);
    fichero.print("Time 3: ");
    fichero.println(tiempo_3);
    fichero.print("Time 4: ");
    fichero.println(tiempo_4);
    fichero.print("Time 5: ");
    fichero.println(tiempo_5);
    fichero.print("Ic: ");
    fichero.println(I_c);
    fichero.print("Vcmax: ");
    fichero.println(V_c);
    fichero.flush();
    fichero.close();


    inicio = 4;

     String Vc_arduino=str(convierte_Vcmax_cuentas(V_c));
     String Ic_arduino=str(convierte_Icons_cuentas(I_c));
     String tiempo_3_arduino=str(convierte_tiempo_horas(tiempo_3));
     String tiempo_1_arduino=str(convierte_tiempo_minutos(tiempo_1));
     String tiempo_2_arduino=str(convierte_tiempo_segundos(tiempo_2));
     String tiempo_4_arduino=str(convierte_tiempo_segundos(tiempo_4));
     String tiempo_5_arduino=str(convierte_tiempo_horas(tiempo_5));

     String i="";
```

```
 String t1="";
 String t2="";
 String t3="";
 String t4="";
 String t5="";
 String ic="";
 String vcmax="";

  i=str(inicio);
   t1=tiempo_1_arduino;
   t2=tiempo_2_arduino;
   t3=tiempo_3_arduino;
   t4=tiempo_4_arduino;
   t5=tiempo_5_arduino;

   ic=Ic_arduino;
   vcmax=Vc_arduino;


 tipo_ensayo="4";

    myPort.write("c");
    myPort.write(i);
    myPort.write(",");
    myPort.write(t1);
    myPort.write(",");
    myPort.write(t2);
    myPort.write(",");
    myPort.write(t3);
    myPort.write(",");
    myPort.write(t4);
    myPort.write(",");
    myPort.write(t5);
    myPort.write(",");
    myPort.write(ic);
    myPort.write(",");
    myPort.write(vcmax);

hora_inicio="Comienzo: "+hour()+":"+minute()+":"+second();

if(archivo==null){

 }else{
  archivo.flush();
  archivo.close();
 }
   archivo = createWriter(nom_archivo);
   nom_archivo="I-"+nom_archivo;
  }else{

//STOP CDI

  String i=""+0;
  String t1=""+0;
  String t2=""+0;
  String t3=""+0;
  String t4=""+0;
  String t5=""+0;
  String ic=""+4095;
  String vcmax=""+1000;

tipo_ensayo="0";

   myPort.write("c");
   myPort.write(i);
   myPort.write(",");
   myPort.write(t1);
   myPort.write(",");
   myPort.write(t2);
```

```
        myPort.write(",");
        myPort.write(t3);
        myPort.write(",");
        myPort.write(t4);
        myPort.write(",");
        myPort.write(t5);
        myPort.write(",");
        myPort.write(ic);
        myPort.write(",");
        myPort.write(vcmax);

    if(archivo==null){
      } else {
        archivo.flush();
        archivo.close();
      }
   }
}
void conductimetro(boolean theFlag) {
  if(theFlag==true) {
    fichero = createWriter("Test."+nom_archivo);
    fichero.print("Test 5: ");
    fichero.println("conductimeter CDI");
    fichero.print("CDI type: ");
    fichero.println(tipo_CDI);
    fichero.print("Time 1: ");
    fichero.println(tiempo_1);
    fichero.print("Time 2: ");
    fichero.println(tiempo_2);
    fichero.print("Time 3: ");
    fichero.println(tiempo_3);
    fichero.print("Ic: ");
    fichero.println(I_c);
    fichero.print("Vcmax: ");
    fichero.println(V_c);
    fichero.flush();
    fichero.close();



    inicio = 5;

     String Vc_arduino=str(convierte_Vcmax_cuentas(V_c));
     String Ic_arduino=str(convierte_Icons_cuentas(I_c));
     String tiempo_3_arduino=str(convierte_tiempo_horas(tiempo_3));
     String tiempo_1_arduino=str(convierte_tiempo_minutos(tiempo_1));
     String tiempo_2_arduino=str(convierte_tiempo_segundos(tiempo_2));

     String i="";
     String t1="";
     String t2="";
     String t3="";
     String ic="";
     String vcmax="";

      i=str(inicio);
       t1=tiempo_1_arduino;
       t2=tiempo_2_arduino;
       t3=tiempo_3_arduino;
       ic=Ic_arduino;
       vcmax=Vc_arduino;


     tipo_ensayo="5";

        myPort.write("c");
        myPort.write(i);
        myPort.write(",");
        myPort.write(t1);
```

```
        myPort.write(",");
        myPort.write(t2);
        myPort.write(",");
        myPort.write(t3);
        myPort.write(",");
        myPort.write(ic);
        myPort.write(",");
        myPort.write(vcmax);

    hora_inicio="Begin: "+hour()+":"+minute()+":"+second();

    if(archivo==null){

     }else{
      archivo.flush();
      archivo.close();
     }
       archivo = createWriter(nom_archivo);// Se crea un TXT
       nom_archivo="I-"+nom_archivo;
      }else{

    //STOP de CDI

      String i=""+0;
      String t1=""+0;
      String t2=""+0;
      String t3=""+0;
      String ic=""+4095;
      String vcmax=""+1000;

    tipo_ensayo="0";

       myPort.write("c");
       myPort.write(i);
       myPort.write(",");
       myPort.write(t1);
       myPort.write(",");
       myPort.write(t2);
       myPort.write(",");
       myPort.write(t3);
       myPort.write(",");
       myPort.write(ic);
       myPort.write(",");
       myPort.write(vcmax);

    if(archivo==null){
      } else {
        archivo.flush();
        archivo.close();
      }
   }
}


/*****************************************************************************
                    Interface configuration
*****************************************************************************/

void draw () {
// Comand DRAW
   fill(255, 243, 224);
   rect(x_r1,y_r1,dx_r1,dy_r1);
   color(250,0,0,205);
   fill(0, 102, 153);
    text("         ### Measured data ###", x_r1+mx_r1, y_r1+my_r1);
    text("> vc_measu= "+dato_1, x_r1+mx_r1, y_r1+my_r1+dy_t);
    text("> ic_measu= "+dato_2, x_r1+mx_r1, y_r1+my_r1+2*dy_t);
    text("> Time 1= "+tiempo_1+" minutes", x_r1+mx_r1, y_r1+my_r1+3*dy_t);
    text("> Time 2= "+tiempo_2+" seconds", x_r1+mx_r1, y_r1+my_r1+4*dy_t);
```

```
    text("> Time 3= "+tiempo_3+" hours", x_r1+mx_r1, y_r1+my_r1+5*dy_t);
    text("> Time 4= "+tiempo_4+" secons", x_r1+mx_r1, y_r1+my_r1+6*dy_t);
    text("> Time 5= "+tiempo_5+" hours", x_r1+mx_r1, y_r1+my_r1+7*dy_t);
    text("> Icons_c = "+I_c+" A", x_r1+mx_r1, y_r1+my_r1+8*dy_t);
    text("> Vcmax = "+V_c+" V", x_r1+mx_r1, y_r1+my_r1+9*dy_t);
    text("> Supercap_type= "+tipo_CDI+" prototype", x_r1+mx_r1, y_r1+my_r1+10*dy_t);


  fill(255, 243, 224);
  rect(x_r2,y_r2,dx_r2,dy_r2);
  color(250,0,0,205);
  fill(0, 102, 153);
   text("File name:",x_r2+mx_r2, y_r2+my_r2);
   text(nom_archivo,x_r2+mx_r2, y_r2+my_r2+dy_t);
   text(hora_inicio,x_r2+mx_r2, y_r2+my_r2+2*dy_t);
   text("Saving: "+hour()+":"+minute()+":"+second(),x_r2+mx_r2, y_r2+my_r2+3*dy_t);

   fill(255, 255, 255);
   strokeWeight(1);
   color(250,0,0,255);

  strokeWeight(2);

// Input parameter panel interface configuration

   fill(255, 355, 205);
   rect(x_r4,y_r4,dx_r4,dy_r4);
   color(250,0,0,205);
   fill(0, 102, 153);
    text("                    **** Input parameters **** ", x_r4+mx_r4, y_r4+my_r4);
    text("Timer_1(min):  ", x_r4+mx_r4, y_r4+my_r4+dy_t);
    text("Timer_2(seg):  ", x_r4+mx_r4, y_r4+my_r4+2*dy_t);
    text("Timer_3(h):  ", x_r4+mx_r4, y_r4+my_r4+3*dy_t);
    text("Timer_4(seg):  ", x_r4+mx_r4, y_r4+my_r4+4*dy_t);
    text("Timer_5(h):  ", x_r4+mx_r4, y_r4+my_r4+5*dy_t);
    text("Icons_c(A):  ", x_r4+mx_r4, y_r4+my_r4+6*dy_t);
    text("Vcmax(V):  ", x_r4+mx_r4, y_r4+my_r4+7*dy_t);
    text("CDI type: ", x_r4+mx_r4, y_r4+my_r4+8*dy_t);
    text("Name.file:",x_r4+29*mx_r4, y_r4+2*my_r4);
    text("Test."+para_nom_archivo,x_r4+29*mx_r4, y_r4+2*my_r4+dy_t);




   fill(255, 355, 205);
   rect(x_r5,y_r5,dx_r5,dy_r5);
   color(250,0,0,205);
   fill(0, 102, 153);
    text("Panel", x_r5+6*mx_r5, y_r5+my_r5);
    text("Vcmax ", x_r5+2*mx_r5, y_r5+my_r5+1*dy_t);
    text("Q", x_r5+4*mx_r5, y_r5+my_r5+5*dy_t/2);
    text("Rdesc", x_r5+mx_r5, y_r5+my_r5+8*dy_t/2);
    text("Ch-dis", x_r5+mx_r5, y_r5+my_r5+11*dy_t/2);
    text("chi", x_r5+4*mx_r5, y_r5+my_r5+14*dy_t/2);



   fill(255, 243, 224);
   rect(x_r6,y_r6,dx_r6,dy_r6);
   color(255,255,0,0);
   fill(255, 0, 0);
   ellipse(x_r6+2*mx_r4, y_r6+3.5*my_r6/2, 20, 20);
   text(estado_CDI,x_r6+4*mx_r6,y_r6+2.5*my_r6/1.25);
   text("Supercap state", x_r6+5*mx_r6, y_r6+0.85*my_r6);
}


void serialEvent (Serial myPort) {
```

```
  try{


inString = myPort.readStringUntil('\n');
  inString = trim(inString);

  if (inString != null) {
    String[] lista = split(inString,",");
     dato_1=lista[0];
     dato_2=lista[1];
     dato_3=lista[4];



   estado_CDI=" ";

    switch(int(dato_3)) {
      case 1:
        estado_CDI="Precarga";
       break;
      case 2:
        estado_CDI="Cargando";
       break;
     case 3:
        estado_CDI="Auto-descarga";
       break;
     case 4:
        estado_CDI="Finalizado";
      break;
    case 0:
        estado_CDI="Stop";
      break;
  }
/*****************************************************************************
                        Save data to file txt
*****************************************************************************/

guardar_datos=int(dato_3);

  if(guardar_datos>3){
     guardar_datos=0;
     archivo.close();
  }

    if(tipo_ensayo=="0"){
    guardar_datos=0;
    }

       if(guardar_datos>0){
         archivo.print(year()+"  ");              // Year
         archivo.print(month()+"  ");            // Month
         archivo.print(day()+"  ");             // Day

         archivo.print(hour()+"  ");          // Hours
         archivo.print(minute()+"  ");       // minutes
         archivo.print(second()+"  ");      // Seconds

         archivo.print(tipo_ensayo+"  ");

         archivo.print(dato_3+"  ");
         archivo.print(dato_1+"  ");
         archivo.println(dato_2+"  ");

        archivo.flush();
       }

    }
}
   catch(RuntimeException e) {
```

```
      println(e);
    }

}

/*************************************************************************************

*************************************************************************************/

public void load_file_parameters(int theValue) {

    println("Boton del Evento: "+theValue);

    String lines[] = loadStrings("Ensayo."+para_nom_archivo);
    String list[] = split(lines[1], ':');
      tipo_CDI=list[1];
        list = split(lines[2], ':');
      tiempo_1 =list[1];
        list = split(lines[3], ':');
      tiempo_2 =list[1];
        list = split(lines[4], ':');
      tiempo_3 =list[1];
        list = split(lines[5], ':');
      I_c =list[1];
        list = split(lines[6], ':');
      V_c =list[1];
       para_nom_archivo="nombre.txt";
}
/*************************************************************************************
                          Subrutines
*************************************************************************************/

    int convierte_tiempo_horas(String t_string) {

      float tiempo_float=0;
      int tiempo_int_cuentas=0;

          tiempo_float=float(t_string);
          tiempo_int_cuentas=round(3600*tiempo_float*muestras_segundo);
        return tiempo_int_cuentas;
    }

    int convierte_tiempo_minutos(String t_string) {

       float tiempo_float=0;
       int tiempo_int_cuentas=0;

          tiempo_float=float(t_string);
          tiempo_int_cuentas=round(60*tiempo_float*muestras_segundo);
       return tiempo_int_cuentas;
    }

    int convierte_tiempo_segundos(String t_string) {

        float tiempo_float=0;
        int tiempo_int_cuentas=0;

          tiempo_float=float(t_string);
          tiempo_int_cuentas=round(tiempo_float*muestras_segundo);
        return tiempo_int_cuentas;
    }

    int convierte_Icons_cuentas(String Icons_string) {

        float Ic_float=0;
        int Icons_int_cuentas=2048;

          Ic_float=float(Icons_string);
          Ic_float=Ic_float-.0;
```

```
      Ic_float=-14741.88*Ic_float+2047.5;
      Icons_int_cuentas=round(Ic_float);
    return Icons_int_cuentas;
}

int convierte_Vcmax_cuentas(String Vcmax_string) {

    float Vcmax_float=1.2;
    int Vcmax_int_cuentas=0;
    float sensibilidad=0.125;
    float Voffset=0.7;

      Vcmax_float=float(Vcmax_string);
      Vcmax_float=(Vcmax_float+Voffset)*1000/sensibilidad;
      Vcmax_int_cuentas=round(Vcmax_float);
    return Vcmax_int_cuentas;
}
```

References

1. Conway, B. *Electrochemical Supercapacitors: Scientific Fundamentals and Technological Applications*, 1st ed.; Kluwer Academic/Plenum: New York, NY, USA, 1999.
2. Burke, A. Ultracapacitors: Why, how, and where is the technology. *J. Power Sources* **2001**, *91*, 37–50. [CrossRef]
3. Lhomme, W.; Delarue, P.; Barrade, P.; Bouscayrol, A.; Rufer, A. Design and Control of a supercapacitor storage system for traction applications. In Proceedings of the Industry Applications Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005, Hong Kong, China, 2–6 October 2005; pp. 2013– 2020.
4. Kang, H.W.; Lee, H.S.; Rhee, J.H.; Lee, K.A. DC Voltage Source Based on a Battery of Supercapacitors with a Regulator in the Form of an Isolated Boost LCC Resonant Converter. *Energies* **2023**, *16*, 6721. [CrossRef]
5. Abbey, C.; Joos, G. Supercapacitor Energy Storage for Wind Energy Applications. *IEEE Trans. Ind. Appl.* **2007**, *43*, 769–776. [CrossRef]
6. Al-Tameemi, Z.H.A.; Lie, T.T.; Foo, G.; Blaabjerg, F. Optimal Coordinated Control of DC Microgrid Based on Hybrid PSO?GWO Algorithm. *Electricity* **2022**, *3*, 346–364. [CrossRef]
7. Pelosi, D.; Gallorini, F.; Alessandri, G.; Barelli, L. A Hybrid Energy Storage System Integrated with a Wave Energy Converter: Data-Driven Stochastic Power Management for Output Power Smoothing. *Energies* **2024**, *17*, 1167. [CrossRef]
8. Phor, L.; Kumar, A.; Chahal, S. Electrode materials for supercapacitors: A comprehensive review of advancements and performance. *J. Energy Storage* **2024**, *84*, 110698. [CrossRef]
9. Podlubny, I. *Fractional Differential Equations*, 1st ed.; Academic Press: San Diego, CA, USA, 1999.
10. Axtell, M.; Bise, M. Fractional calculus applications in control systems. In Proceedings of the IEEE 1990 National Aerospace and Electronics Conference, Dayton, OH, USA, 21–25 May 1990.
11. Petras, I.; Podlubny, I.; OLeary, P.; Dorkac, L.; Vinagre, B. *Analogue Realization of Fractional Order Controllers*, 1st ed.; FBERG, Technical University of Kosice: Kosice, Slovakia, 2002.
12. Oldham, K.; Spanier, J. *The Fractional Calculus: Integration and Differentiations of Arbitrary order*, 1st ed.; Academic: New York, NY, USA, 1974.
13. Sokolov, I.M.; Klafter, J.; Blumen, A. Fractional Kinetics. *Phys. Today* **2002**, *55*, 48–55. [CrossRef]
14. Calderon, A.; Vinagre, B.; Feliu, V. Fractional order control strategies for power electronic buck converters. *Signal Process.* **2006**, *86*, 2803–2819. [CrossRef]
15. Oustaloup, A.; Sabatier, J.; Moreau, X. From fractal robustness to the CRONE approach. *Proc. ESAIM* **1998**, *5*, 177–192. [CrossRef]
16. Liu, S.; Sun, H.; Yu, H.; Miao, J.; Zheng, C.; Zhang, X. A framework for battery temperature estimation based on fractional electro-thermal coupling model. *J. Energy Storage* **2023**, *63*, 107042. [CrossRef]
17. Vinagre, B.; Feliu, V.; Feliu, J. Frequency domain identification of a flexible structure with piezoelectric actuators using irrational transfer function model. In Proceedings of the 36th Conference on Decision and Control, San Diego, CA, USA, 12 December 1997; pp. 1278–1280.
18. Nigmatullin, R.; Mehaute, A.L. Is there geometrical/physical meaning of the fractional integral with complex exponent? *J. Non-Cryst. Solids* **2005**, *351*, 2888–2899. [CrossRef]
19. Jesus, I.; Tenreiro Machado, J.; Bohaventura Cunha, J. Fractional electrical dynamics in fruits and vegetables. In Proceedings of the IFAC Workshop FDA. IFAC, Porto, Portugal, 19–21 July 2006.
20. Haschka, M.; Ruger, B.; Krebs, V. Identification of the electrical behavior of a solid oxide fuel cell in the time domain. In Proceedings of the IFAC Workshop FDA, Bordeaux, France, 19–21 July 2004; pp. 327–333.
21. Quintana, J.; Ramos, A.; Nuez, I. Identification of the fractional impedance of ultracapacitors. In Proceedings of the IFAC Workshop FDA. IFAC, Porto, Portugal, 19–21 July 2006.
22. Martin, R.; Quintana, J.; Ramos, A.; Nuez, I. Modeling of electrochemical double layer capacitors by means of fractional impedance. *Comput. Nonlinear Dyn. ASME* **2008**, *3*, 021303. [CrossRef]

23. Martin, R.; Quintana, J.; Ramos, A.; Nuez, I. Fractional equivalent impedance of electrochemical double layer capacitors combinations. *J. Eur. Des Syst. Autom.* **2008**, *42*, 923–938. [CrossRef]
24. Maity, S.; Saha, M.; Saha, P.; Khanra, M. Fractional calculus-based modeling and state-of-charge estimation of supercapacitor. *J. Energy Storage* **2024**, *81*, 110317. [CrossRef]
25. Chaban, V.V.; Andreeva, N.A.; Fileti, E.E. Graphene/ionic liquid ultracapacitors: Does ionic size correlate with energy storage performance? *New J. Chem.* **2018**, *42*, 18409–18417. [CrossRef]
26. Spyker, R.; Nelms, R. Optimization of double-layer capacitor arrays. *IEEE Trans. Ind. Appl.* **2000**, *36*, 194–198. [CrossRef]
27. Spyker, R.; Nelms, R. Classical equivalent circuit parameters for a double-layer capacitor. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 829–836. [CrossRef]
28. Zhai, N.; Zhang, D.; Xu, D. Design and optimization for a supercapacitor application system. In Proceedings of the 2006 International Conference on Power System Technology, Chongqing, China, 22–26 October 2006; pp. 1–4.
29. Zhong, Y.; Zhang, J.; Li, G.; Liu, A. Research on Energy Efficiency of Supercapacitor Energy Storage System. In Proceedings of the 2006 International Conference on Power System Technology, Chongqing, China, 22–26 October 2006; pp. 1–4.
30. Ramos, A.; Quintana, J.; Martin, R.; Nuez, I. Esquema de control para un convertidor de carga-descarga para supercondensadores. In Proceedings of the SAAE I 2007, Puebla, Mexico, 10–12 September 2007; pp. 43–48.
31. Ramos, A. Convertidor de Potencia con Almacenamiento Energetico, para la Interconexion de Sistemas Electricos de Reducida Potencia. Ph.D. Thesis, Las Palmas de G.C. University, Las Palmas, Spain , 2008.
32. Wei, T.; Qi, X.; Qi, Z. An improved ultracapacitor equivalent circuit model for desing of energy storage power systems. In Proceedings of the International Conference on Electrical Machines and Systems, Seoul, Republic of Korea, 8–11 October 2007; pp. 69–73.
33. Conway, B.; Pell, W. Power limitations of supercapacitor operation associated with resistance and capacitance distribution in porous electrode devices. *J. Power Sources* **2002**, *105*, 169–181. [CrossRef]
34. Belhachemi, F.; Raul, S.; Davat, B. A phisical based of power electric double-layer supercapacitors. In Proceedings of the Conference Record of the 2000 IEEE Industry Applications, Rome, Italy, 8–12 October 2000; pp. 3069–3076.
35. Kurzweil, P.; Frenzel, B. Capacitance Characterization Methods and Ageing Behaviour of Supercapacitors. In Proceedings of the 15th International Seminar On Double Layer Capacitors, Deerfield Beach, FL, USA, 5–7 December 2005; pp. 1–12.
36. Podlubny, I. Geometric and physical interpretation of fractional integration and fractional differentiation. *Int. J. Theory Appl. Fract. Calc. Appl. Anal.* **2002**, *5*, 367–386.
37. Dorcak, L.; Lesko, V.; Kostial, I. Identification of Fractional-Order Dynamical Systems. In Proceedings of the 12th International Conference on Process Control and Simulation ASRTP'96, Tahoe, CA, USA, 7–12 January 1996; pp. 62–68.