Data in Brief 40 (2022) 107802

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Direct current geared motor data: Voltage, current, and speed measured under different experimental conditions

Wallace Pereira Neves dos Reis[a,c,*], Giselle Elias Couto[b], Orides Morandin Junior[c]

[a] *Federal Institute of Education, Science, and Tehcnology of Rio de Janeiro (IFRJ), Volta Redonda campus, Volta Redonda - 27215-350, Rio de Janeiro, Brazil*
[b] *Federal Center for Technological Education Celso Suckow da Fonseca (CEFET-RJ), Itaguaí Campus, Itaguaí - 23812-101, Rio de Janeiro, Brazil*
[c] *Computing Department, Federal University of São Carlos (UFSCar), São Carlos - 13565-905, São Paulo, Brazil*

A R T I C L E I N F O

Article history:
Received 18 October 2021
Revised 9 December 2021
Accepted 4 January 2022
Available online 6 January 2022

Keywords:
DC motor
Modeling
System Identification
Motor dynamics
Automated Guided Vehicle
Position control
Mobile robot

A B S T R A C T

This data article describes eleven datasets collected from laboratory individual tests with two DC motors of the same model. The motors are proposed to be used as the actuators of an Automated Guided Vehicle (AGV). Each dataset shares the same structure, with the measurement of twelve variables: instant of measurement, encoder pulse counts, calculated motor velocity, raw current, calculated current, raw voltage from output A1, raw voltage from output B1, calculated voltage from output A1, calculated voltage from output B1, potential difference applied to the motor terminals, motor status, and the Arduino analog output value in pulse width modulation (PWM). The data are helpful to model and identify the system considering its dynamics. Such consideration on control systems design, specifically on AGV position control, can improve the controller accuracy. It also can

* Corresponding author at: Federal Institute of Education, Science, and Tehcnology of Rio de Janeiro (IFRJ), Volta Redonda campus, Volta Redonda - 27215-350, Rio de Janeiro, Brazil.
E-mail address: wallace.reis@ifrj.edu.br (W.P.N. dos Reis).

https://doi.org/10.1016/j.dib.2022.107802

be useful to study robot design, and mobile robot and AGV simulation.

Specifications Table

Subject	Control and Systems Engineering
Specific subject area	Permanent magnetic direct current motors modeling and parameters identification. Identification of DC motor dynamic parameters to improve the accuracy of an AGV position controller.
Type of data	Table Graph Spreadsheet file
How the data were acquired	Data were acquired throughout laboratory individual tests with two motors of the same model. The primary data acquisition system is an Arduino Mega 2560. Using VNH2SP30 full-bridge motor drivers to control the input voltage using an Arduino PWM output, the system measures the motor's voltage, current, and speed. The voltage sensor is a simple voltage divider with resistors to adapt the motor voltage range (0-12V) to a range allowed to an Arduino analog input (0-4V). The current sensor, model ACS712, is a Hall effect-based linear current sensor with an input range -30 A to 30 A and a proportional analog output compatible with the Arduino analog input. The speed sensor combines an OMRON rotary incremental encoder with 1800 pulses per revolution, model E6B2-CWZ6C, and a quadrature encoder buffer based on the integrated circuit (IC) LS7366R communicating to the Arduino board using SPI protocol. The data were saved to comma-separated valuesfile via serial communication using the Arduino IDE software, the external tool ArduSpreadsheet, and the codes in [1] to generate input signals and acquired the data at 100Hz.
Data format	Raw Filtered
Parameters for data collection	The experiments were conducted in a controlled environment. The data collection process considers the following input signals under a no-load condition: sine and triangle waves (0.1 Hz, 1 Hz, and 10 Hz), a 0.25 Hz square wave, two Pseudorandom Binary Sequence signals (with 7 and 9 bits), and step functions. Only step functions were tested under a loaded condition with four load states: 1 kg, 2 kg, 3 kg, and 4 kg.
Description of data collection	A test platform was set up, allowing the experiments with and without a load attached to the motor shaft. The same sensors were used for both motors. Except for input voltages that do not start the DC motor, at least ten trials were conducted for each input signal and each motor. Besides the filtered values of voltage, current, and speed, the data also includes the time interval of each measurement, the raw values of the Arduino analog inputs, the number of encoder counts, and the reference values for motor control: the motor status (clockwise spin, counterclockwise spin, or brake), and the reference PWM output value that commands the motor driver.
Data source location	Federal Institute of Education, Science, and Tehcnology of Rio de Janeiro (IFRJ), Volta Redonda campus, Volta Redonda, Rio de Janeiro State, Brazil
Data accessibility	With the article or Repository name: Mendeley Data Data identification number: 10.17632/2rkpsss6fd.2 Direct URL to data: https://data.mendeley.com/datasets/2rkpsss6fd/2 Code accessibility Repository name: GitHub/Zenodo Data identification number: 10.5281/zenodo.5737539 Direct URL to data: https://zenodo.org/record/5737539#.YbKiprpv9hF

Value of the Data

- The presented data allows the modeling and system identification of a DC motor from different approaches due to the number of recorded variables. The number of input signals tested and the experiments under load adjustment conditions differentiate the dataset from others.
- Mobile robot and Automated Guided Vehicle (AGV) simulation research, robot design, and control systems design can benefit from the data. The actuator dynamics are usually neglected in such cases. However, the regular operation of an AGV includes transportation of different payloads, which impacts the motor torque and dynamic behavior, besides other variables. Therefore, the experiments stress the tested motors with concentrated loads to measure the dynamic behavior under diverse conditions.
- Besides the DC modeling for simulation and control of dynamic systems use, the data also allows comparisons between system identification algorithms and attends as a training dataset to intelligent control approaches.
- The data can be used to compare AGV position controllers' performance regarding position accuracy when considering the actuator dynamics.
- It is also useful to study the impact of the actuator dynamics modeling on the position control simulation of AGVs and its comparison to the actual vehicle.

1. Data Description

Data reported in this article describe the voltage, current, and velocity measurements of two DC motors of the same model. The motors are the actuators of an AGV in the development phase. As regular vehicle operation involves load transportation, it impacts the motor dynamics, mainly its torque. So, the experimental platform comprises the sensors to measure the variables mentioned above and the adjustment of a concentrated load in the motor shaft to emulate a load transportation condition. Similar experiments can be performed on the mounted vehicle.

Besides the processed measurements data of the DC motor, the data set also comprises the acquisition instant, the raw sensors readings, and the motor commands. Each dataset in the spreadsheet file consists of twelve variables recordings, divided into twelve columns as Table 1 shows, respectively described next. The recorded variables are common to every experiment and for both motors that were tested. The first column is the instant of measurements, the measurement time recorded in microseconds. The measurement with the Arduino board uses the timer interruption to generate a measurement rate of 100 Hz.

The following two columns register the encoder pulse counts and the calculated motor speed in RPM. The quadrature encoder buffer circuit was configured to count each pulse rising. So, the `encoderCount` column shows the pulse count without clearing the value. The pulse count signal indicates the motor's rotation direction, being a positive signal related to a counterclockwise rotation and a negative signal related to a clockwise rotation. The motor velocity, showed in `Velocity` column, is calculated from the pulse counts considering the measurement interval of 0.01 seconds, according to Eq. (1), where V is the motor velocity in rotations per minute (RPM), $N = 1/17$ is the reduction rate of the gear box, $encoderCount(n)$ is the actual value of the encoder buffer, $encoderCount(n-1)$ is the last value of the encoder buffer used in the previous iteration, PPR is the number of pulses per revolution of the encoder model used, which is 1800 PPR, Δt is the sampling time of 0.01 s, and 60 s/1 min is a unit conversion constant. Positive velocity means a clockwise rotation direction to follow the voltage signal applied to the motor.

$$V = N\frac{\frac{-[encoderCount(n) - encoderCount(n-1)]}{PPR} \times \frac{60\,s}{1\,min}}{\Delta t} \tag{1}$$

The current measurement has two distinct columns. The first one, named `rawCurrent`, is a single raw measurement of the Arduino board analog input, with a 10 bits resolution. So, such values are in the range from 0-1023. The sensor range is from -30 A to 30 A, and its 0-5 V

Table 1
The motor data structure: few lines of a single trial to exemplify the data organization.

time	Encoder Count	Velocity	raw Current	Current	raw VoltageA1	raw VoltageB1	VoltageA1	VoltageB1	Motor Voltage	Motor Status	PWM
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
131540	-1119	190	618	9	23	798	0.33	11.49	11.17	1	237
141520	-2688	307	511	2.36	12	813	0.17	11.71	11.54	1	179
151536	-4532	361	510	0.34	9	820	0.13	11.81	11.68	1	100
161532	-6454	376	511	-0.01	256	831	3.66	11.97	8.31	1	33
171544	-8367	375	510	-0.03	276	836	3.94	12.04	8.1	1	1
181536	-10228	364	511	0.03	241	835	3.44	12.03	8.59	1	17
191560	-12045	356	538	0.39	266	835	3.8	12.03	8.23	1	76
201540	-13818	347	545	2.35	272	839	3.88	12.09	8.2	1	155
211576	-15637	356	545	3.64	900	834	12.85	12.01	-0.84	1	222
221548	-17561	377	554	3.57	9	822	0.13	11.84	11.71	1	254
231584	-19675	414	548	3.01	9	824	0.13	11.87	11.74	1	238
241568	-21939	443	512	0.93	6	824	0.09	11.87	11.78	1	179
251584	-24271	457	510	0.27	2	826	0.03	11.9	11.87	1	103
261584	-26600	456	511	0.01	0	830	0	11.96	11.96	1	35
271596	-28894	449	511	-0.01	176	832	2.51	11.98	9.47	1	1
281588	-31132	438	510	-0.03	170	832	2.43	11.98	9.56	1	17
291576	-33313	427	511	0.17	188	836	2.69	12.04	9.36	1	73
301588	-35456	420	540	1.33	191	833	2.73	12	9.27	1	152
311588	-37593	419	544	2.32	896	836	12.8	12.04	-0.75	1	220
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

output is proportional to the motor current, then the value 512 is the zero current value. The other column, `Current` is the calculated current based on the sensor model using a software library [2], which calculates the measured current according to Eq. (2), where i is the measured current, acc is the variable that accumulates ten readings of the Arduino analog input pin in a row, $ADC_{scale} = 1023$ is the analog to digital converter scale, $v_{ref} = 5\ V$ is the reference voltage, and $S_{sensor} = 0.066\ mV/A$ is the sensor sensibility, related to its current input range. Then, the current reading from the sensor is an average of ten readings.

$$i = \frac{(acc/10)}{ADC_{scale}\frac{v_{ref}}{S_{sensor}}} \tag{2}$$

Likewise, the voltage measures also display the raw measurements from both motor driver outputs, A1 and B1, since they have the same polarity. So, the data is divided into columns `rawVoltageA1` and `rawVoltageB1`. For each driver output, a voltage divider circuit with resistors converts the motor voltage from a 0-12 V scale to a 0-4 V scale, suitable for the Arduino analog input pins. As analog input readings, both columns register values in the range 0-1023.

The following two columns, named `VoltageA1` and `VoltageB1`, display the calculated voltage from each drive output, i.e., converted from the readings to the 0-12 V scale. The values are calculated by Eq. (3), where $x = \{A1,\ B1\}$, r_x is the reading of the respective Arduino analog input pin, 5/1024 is the constant voltage resolution of the analog to digital converter, and $R1_x$ and $R2_x$ are the measured resistance values of the resistors in the voltage divider circuit.

$$v_x = \frac{r_x \frac{5}{1024}}{\frac{R2_x}{R2_x + R1_x}} \tag{3}$$

The potential difference applied to the motor terminals is determined based on the measurements of each driver output. It is registered on column `MotorVoltage` and calculated by Eq. (4), where v_{motor} is the potential difference applied to the DC motor, v_{A1} is the measured voltage of driver output A1, and v_{B1} is the measured voltage of driver output B1.

$$v_{motor} = v_{B1} - v_{A1} \tag{4}$$

The last two columns of the data files register the motor status and the Arduino analog output value in pulse width modulation (PWM) used to command the motor driver. The column `MotorStatus` has three possible values: 0 for brake condition, 1 for the clockwise rotation direction, and 2 for the counterclockwise rotation direction. Lastly, the column `PWM` registers the values from Arduino analog output with a 0-255 range. It is the reference of the motor command.

The dataset is presented in a spreadsheet file, which tabs are divided into the carried experiment and the DC motor used, motor A or motor B. It includes experimental data from two DC motors submitted to ten different input voltage signals in a no-load condition and voltage steps with different amplitudes with a load attached to the motor shaft. Each experiment is equivalent to a different voltage input stimulus applied to the DC motor. With only four exceptions, stimuli were repeated at least ten times for each motor. Table 2 summarizes the dataset file, indicating the tested voltage input signal and the respective tab within the spreadsheet file.

Two Pseudorandom Binary Sequences (PRBS) were tested as the input voltage signal considering that the higher level is the maximum value of the PWM duty cycle, equivalent to 12 V, and the lower level is zero. A PRBS signal is a binary sequence generated with a deterministic algorithm to allow the reproduction of the sequence. Fig. 1 exemplifies the applied voltage to the DC motors from the PRBS7 signal and Fig. 2 from the PRBS9 signal considering a single experimental data.

The number of shift registers used indicates the sequence maximum number of bits. In the case of the PRBS7, it indicates that the sequence has $2^7 - 1 = 127$ bits before repeating the sequence. Analogously, the PRBS9 signal has $2^9 - 1 = 511$ bits before repeating the sequence.

In the experiments, each bit of the sequence spans approximately 240 milliseconds before transitioning to the next value. The recorded data do not cover the entire sequence, registering the first ten seconds of the PRBS7 signal and the first 20 seconds of the PRBS9 signal.

Table 2
Description of the stimuli signals tested, indicating the tabs' name and the number of trials recorded in the data file.

Input Voltage Signal	Description	Tab Name	N° of Trials	Tab Name	N° of Trials
Pseudorandom Binary Sequence 7	A binary sequence with ten seconds long generated with a deterministic algorithm. Each bit has approximately 240 milliseconds of span. There is no load attached to the motor shaft.	PRBS7-MotorA	10	PRBS9-MotorB	11
Pseudorandom Binary Sequence 9	A binary sequence with twenty seconds long generated with a deterministic algorithm. Each bit has approximately 240 milliseconds of span. There is no load attached to the motor shaft.	PRBS9-MotorA	11	PRBS9-MotorB	11
Sine Wave 10 Hz	Sine wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 388 microseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Sine10Hz-MotorA	10	Sine10Hz-MotorB	10
Sine Wave 1 Hz	Sine wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 3.88 milliseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Sine1Hz-MotorA	10	Sine1Hz-MotorB	10
Sine Wave 0.1 Hz	Sine wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 38.8 milliseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Sine0.1Hz-MotorA	10	Sine0.1Hz-MotorB	10
Triangle Wave 10 Hz	Triangle wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 388 microseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Triangle10Hz-MotorA	10	Triangle10Hz-MotorB	10
Triangle Wave 1 Hz	Triangle wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 3.88 milliseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Triangle1Hz-MotorA	10	Triangle1Hz-MotorB	13
Triangle Wave 0.1 Hz	Triangle wave generated throughout the Arduino PWM output. The algorithm updates the PWM output every 38.8 milliseconds to generate an approximated 10 Hz wave. There is no load attached to the motor shaft.	Triangle0.1Hz-MotorA	10	Triangle0.1Hz-MotorB	10
Square Wave	Square Wave with alternate spin direction and 1/4 Hz. There is no load attached to the motor shaft.	SquareWave-MotorA	10	SquareWave-MotorB	10

(*continued on next page*)

Table 2 (*continued*)

Input Voltage Signal	Description	Tab Name	N° of Trials	Tab Name	N° of Trials
Step without load	Voltage step applied to the motor without attached load and a duration of 2.4 seconds. Trials were divided into step percentages regarding the PWM duty cycle. Steps amplitude: 3%, 4%, 5%, 7%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%.	StepNoLoad-MotorA	122	StepNoLoad-MotorB	152
Step with load	Voltage step applied to the motor with adjusted load attached to the motor shaft. Trials were divided into step percentages regarding the PWM duty cycle and load range (1 kg to 4 kg). The steps duration was modified due to the load, and it was adjusted to maximize the observation time. Steps amplitude: 40%, 50%, 60%, 70%, 80%, 90%, and 100%. Loads: 1 kg, 2 kg, 3 kg, and 4 kg.	StepLoaded-MotorA	283	StepLoaded-MotorB	315

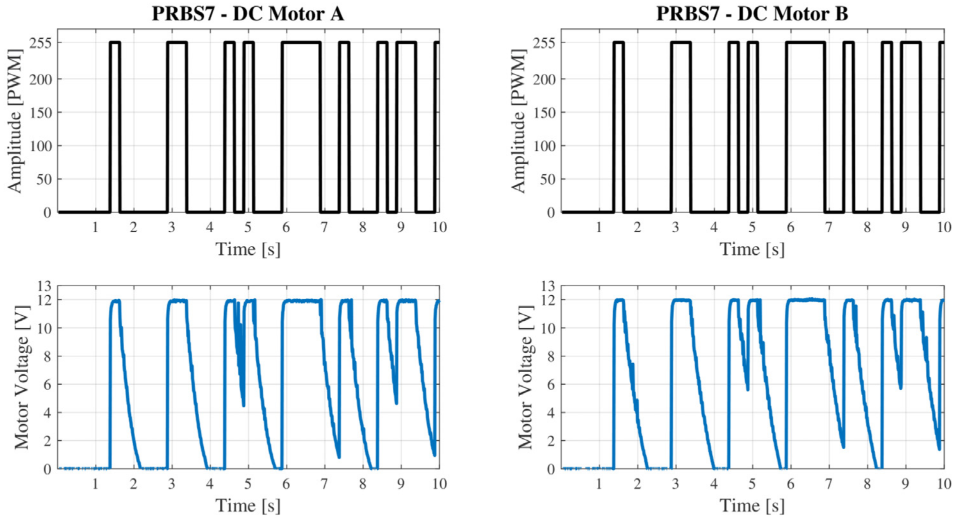

Fig. 1. PRSB7 signal applied to the DC motor A and B and the respective measured voltages.

The sine and triangular waves were tested with three frequencies, 0.1 Hz, 1 Hz, and 10 Hz. The Arduino algorithm uses lookup tables with 256 values to generate the reference signal in output PWM values. To generate the reference signals in each frequency, the algorithm updates the PWM output as described in Table 2, but referring to the same lookup table.

Fig. 3 shows the sine waves reference signal and the motor voltages measured for motor A, while Fig. 4 shows the data for motor B.

Likewise, Fig. 5 shows the triangle waves reference signal and respective measured voltages for motor A and Fig. 6 for motor B.

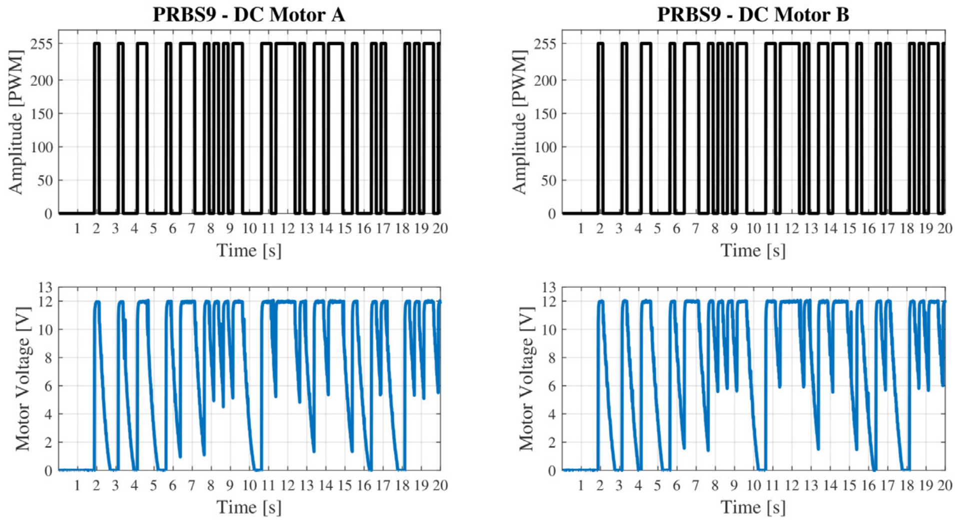

Fig. 2. PRSB9 signal applied to the DC motor A and B and the respective measured voltages.

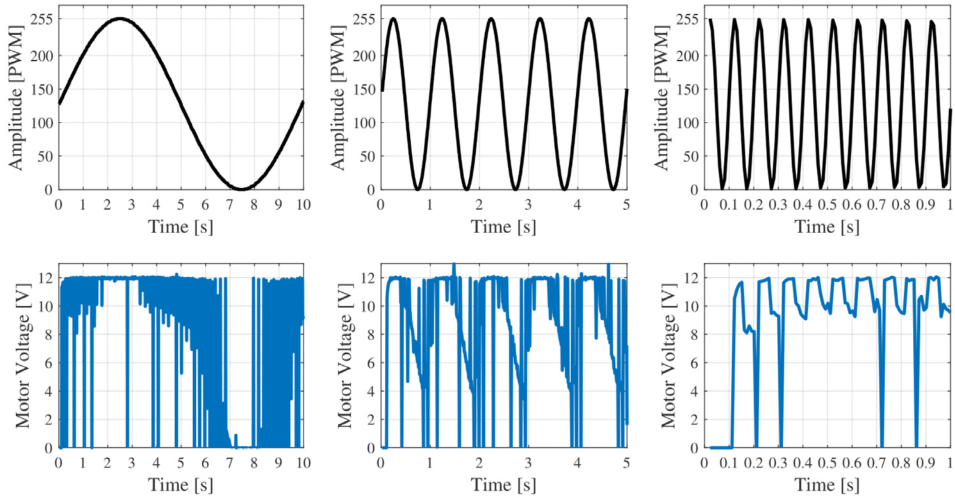

Fig. 3. Sine waves of 0.1 Hz, 1 Hz and 10 Hz, respectively, applied to the DC motor A and the respective measured voltages.

The square wave signal tests the DC motors in both rotation directions. It applies a maximum voltage pulse to the motor during 1 s, alternating the rotation direction, generating a 0.25 Hz wave from the DC motor perspective. Two and a half cycles of the signal were recorded in the experiments.

Fig. 7 shows the reference signals, the motor status values at each instant, and the measured motor voltage for both tested DC motors.

The last type of tested signal is the voltage step. It was applied to the DC motors without and with a load attached to the motor shaft. As described in Table 2, the signal amplitudes cover the output scale from the minimum value that starts the motor to the maximum voltage value under no-load conditions.

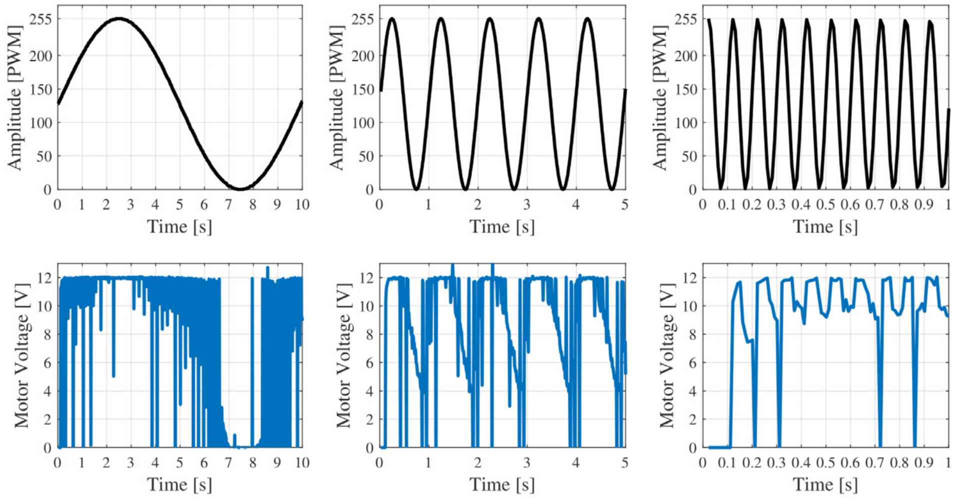

Fig. 4. Sine waves of 0.1 Hz, 1 Hz and 10 Hz, respectively, applied to the DC motor B and the respective measured voltages.

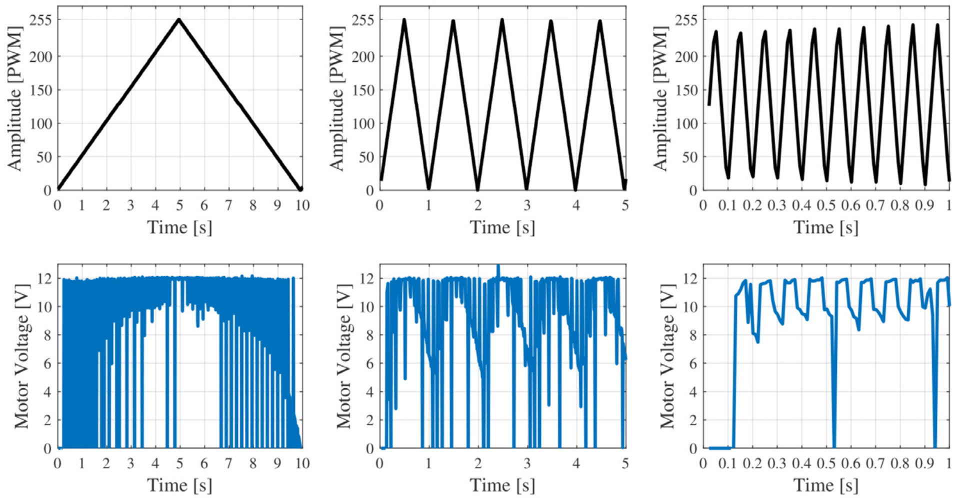

Fig. 5. Triangle waves of 0.1 Hz, 1 Hz and 10 Hz, respectively, applied to the DC motor A and the respective measured voltages.

To exemplify the recorded data, Fig. 8 shows six reference steps and their respective measured motor voltages. The bottom plots detail the measured voltage in a shorter time scale as the motor driver outputs a PWM signal.

Concerning the loaded condition, steps with an amplitude from 40% to 100 % were tested with four load stages, as described in Table 1. The recorded data has different time spans according to the tested load. Fig. 9 shows the input steps and the measured velocities to motor A with the load adjustment. The input is the same for each load value. Likewise, Fig. 10 shows the data from motor B trials.

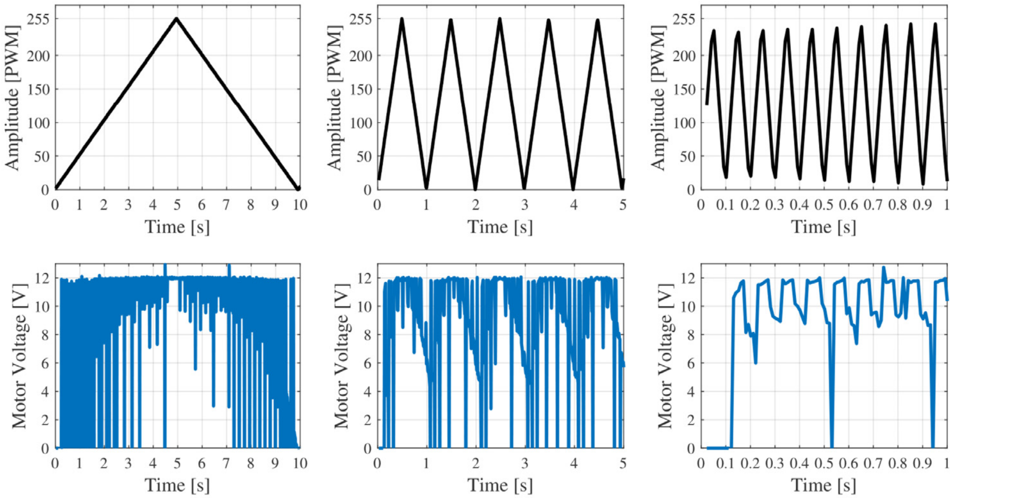

Fig. 6. Triangle waves of 0.1 Hz, 1 Hz and 10 Hz, respectively, applied to the DC motor B and the respective measured voltages.

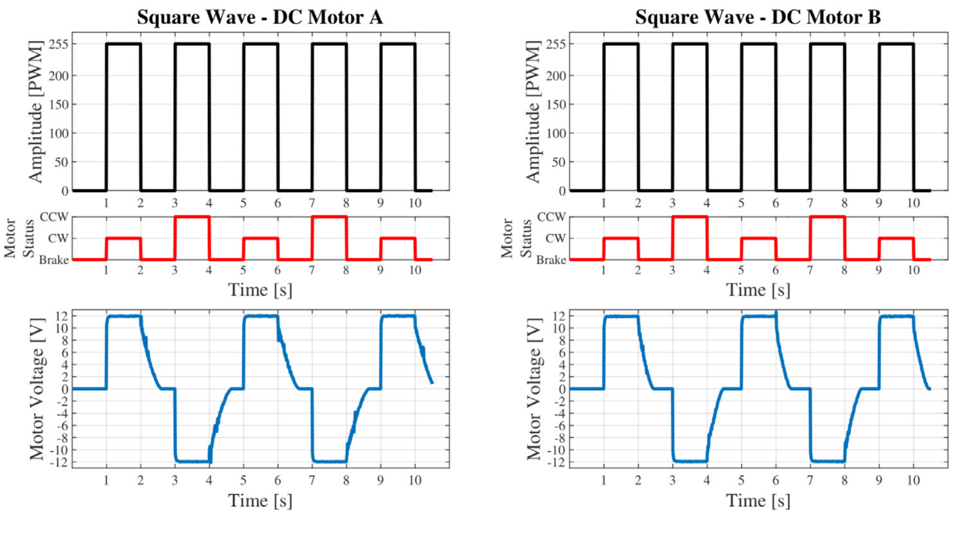

Fig. 7. Square wave signal with rotation direction inversion applied to the DC motor A and B and the respective measured voltages. The Motor Status graph shows the rotation direction at each period, generating a 0.25 Hz wave.

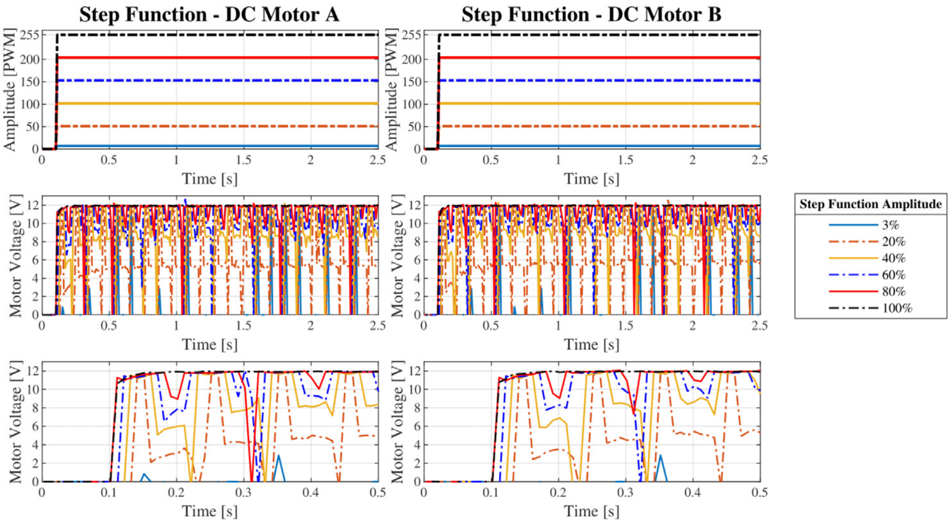

Fig. 8. Voltage steps applied to the DC motor A and B under no-load condition, and the respective measured voltages.

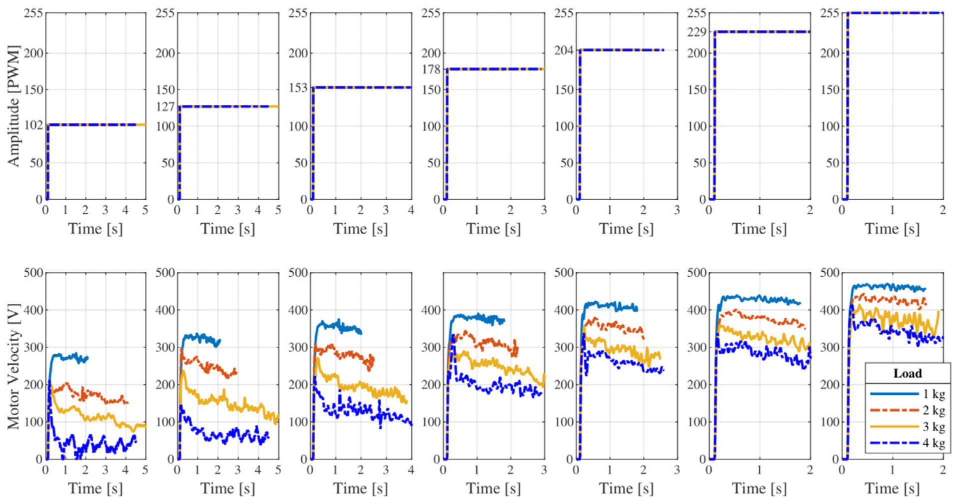

Fig. 9. Step signals applied to the DC motor A under loaded condition, and the respective measured velocities.

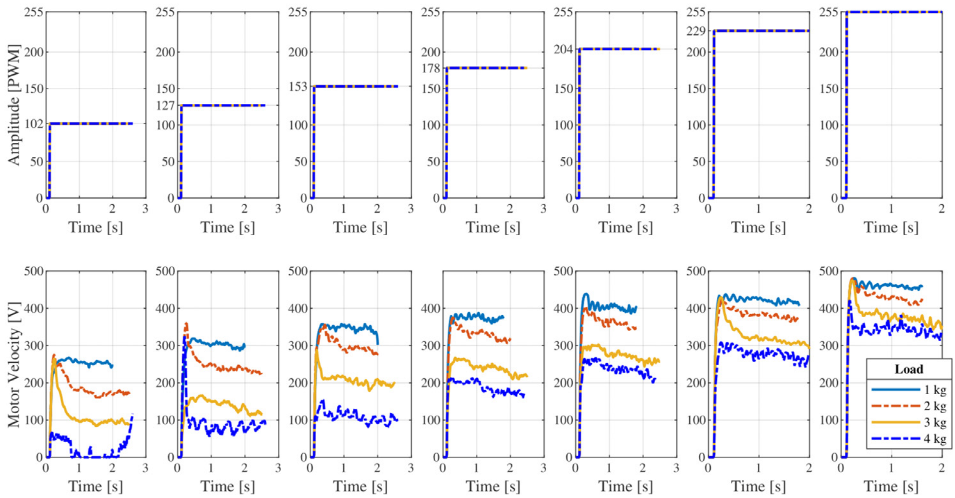

Fig. 10. Step signals applied to the DC motor B under loaded condition, and the respective measured velocities.

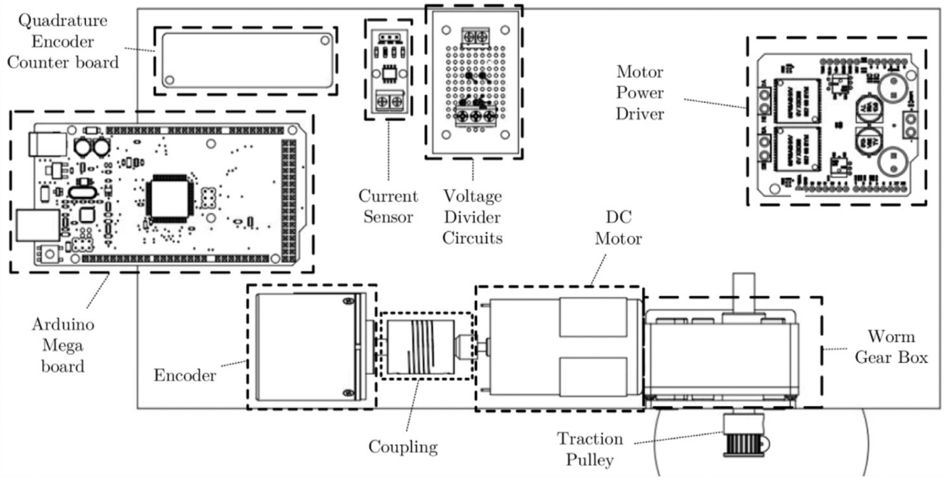

Fig. 11. Experimental platform layout.

2. Experimental Design, Material and Methods

Table 3 shows the description and essential specifications of the sensors used in the experiments to measure the DC motor current, voltage, and speed, and the other devices as the Arduino Mega board, the encoder count board, the DC motors, the objects of the experiments, and the motor driver circuit used for motor actuation.

Fig. 11 shows the experimental platform layout, highlighting each component. To simplify, the devices' interconnections are not displayed. The proposed platform was designed as a portable experimental apparatus that can be used on different workbench heights. A traction pulley is attached to the DC motor shaft to test the motor with a concentrated load to resemble the AGV operation under a loaded condition. It also allows the exchange of the motor-encoder set, keeping the other measuring devices.

Table 3
Description and specifications of the sensors and other electronic devices used in the experiments.

Device	Description	Specifications
Current Sensor	Hall effect-based current sensor ACS712 [3]	Supply voltage: 5.0 V; Output sensitivity: 66 mV=A; Input current range: ±30 A; Bandwidth: 80 kHz; Output rise time in response to step input current 5 µs; Output voltage proportional to AC or DC currents
Voltage Sensor	Voltage divider circuit with three 100 kΩ resistors	Maximum input voltage: 12.0 V; Output range, approx.: 0 – 4 V; Output voltage proportional to the motor driver outputs A1 and B1
Rotary Incremental Encoder	Omron E6B2-CWZ6C [4]	Supply voltage: 5.0 V; Resolution (pulses/rotation): 1800; Current consumption: 80 mA max.; Output phases: A, B, Z; Phase difference between outputs: 90° ± 45° between A and B; Output configuration: NPN open-collector output; Maximum response frequency: 100 kHz
Arduino Board	Arduino Mega 2560 Rev3	Microcontroller: ATmega2560; Clock Speed: 16 MHz; Input Voltage (recommended): 7-12V; Operating Voltage: 5V; Digital I/O Pins: 54 (of which 15 provide PWM output); Analog Input Pins: 16; DC Current per I/O Pin: 20 mA;
Encoder Counter Board	Quadrature Counter with Serial Interface based on the LS7366R IC [5]	Operating Voltage: 3 V to 5.5 V; 5 V count frequency: 40 MHz; 32-bit programmable counter; 32-bit data register and comparator; 32-bit output register; Internal quadrature clock decoder and filter.
DC Motor	Worm Geared DC Motor Model: A58-555-1280	Nominal input voltage: 12.0 V; Rotating Speed (No-load): 470 RPM; Current (No-load): 0.3 A; Nominal Rotating Speed: 400 RPM; Nominal Current: 2 A; Nominal Torque: 6.8 kg.cm; Stall Torque: 17 kg.cm; Stall Current: 5 A
Motor Driver	SparkFun Monster Moto Shield based-on VNH2SP30-E full-bridge IC [6]	Supply Voltage, max.: 16 V; Maximum current rating: 30 A; Practical Continuous Current: 14 A; MOSFET on-resistance: 19 mΩ (per leg); Maximum PWM frequency: 20 kHz; Thermal Shutdown; Undervoltage and Overvoltage shutdown

Fig. 12 presents the electronics schematics of the experimental platform, detailing the connections between the device and Arduino pins used. Arduino's analog input pins read the current and voltage sensors, and the encoder pulses are primarily counted in a quadrature encoder buffer and then transmitted to the Arduino via serial communication using the Serial Peripheral Interface (SPI) protocol.

Fig. 13 shows the typical flowchart of the data acquisition Arduino code [1] used in the experiments. The first lane, called Initial Declarations, encompasses the inclusion of libraries, pins definitions, and the declaration of variables used throughout the code.

Next, in the void setup() lane, the necessary configurations are carried out to execute data acquisition, such as the configuration of the Arduino pins mode, sensor initialization, and the timer used to count the interruption period. Thus, the overflow of the timer, i.e., when it reaches the 10 ms count, is the interruption trigger, which is the routine responsible for reading the sensors and updating the command to the motor driver.

Still in Fig. 13, the lane named void loop() represents the main execution loop of data acquisition. At each interruption occurrence, the sensors are read, the necessary calculations are carried out, the variables of interest are transmitted via serial communication, and the command output of the full-bridge motor driver is updated.

The motor driver command signal is only sent when the motor start criterion is satisfied. For this work, the criterion was a time interval of 10 ms, i.e., the motor only starts after 10 ms have passed since the Arduino was started. Before that, the motor only receives a zero speed value, meaning the duty cycle is minimal, and the output of the motor drive is not enabled. Finally, if

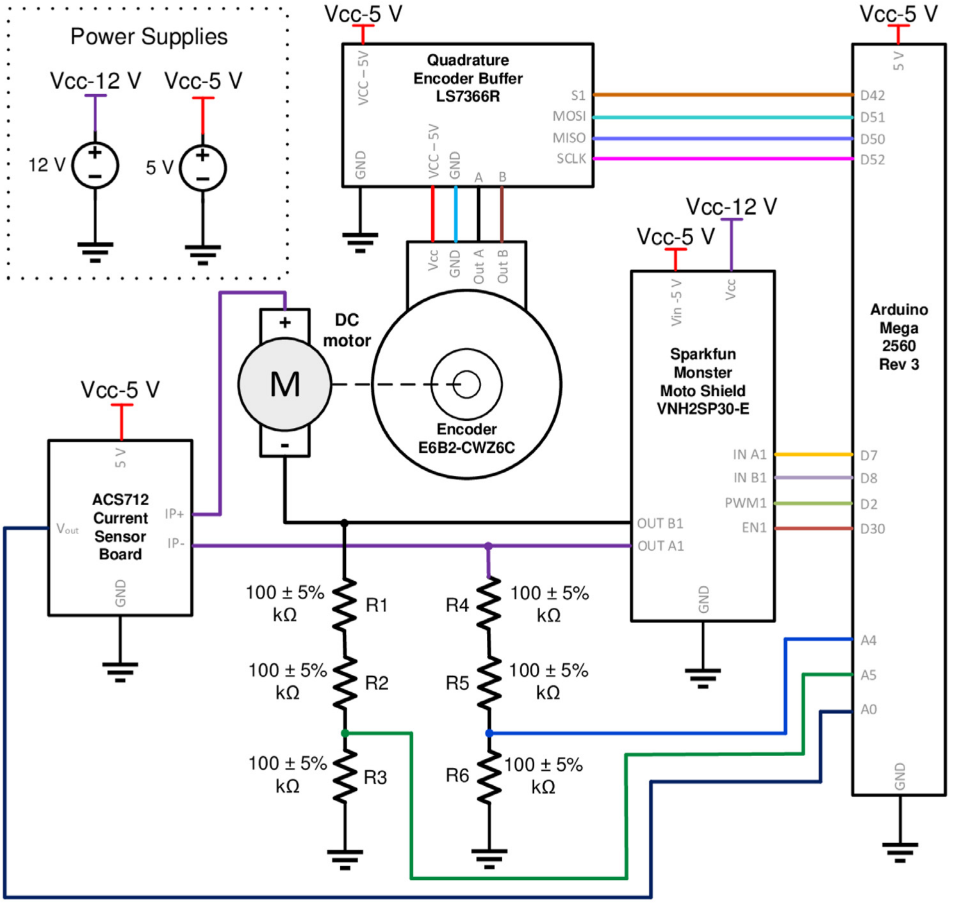

Fig. 12. Electronics schematics of the experimental platform. The resistors measured values are: R1 + R2 = 192:5 kΩ, R3 = 100 kΩ, R4 + R5 = 195 kΩ, and R6 = 100 kΩ.

the necessary data acquisition has been performed, the execution is stopped. The Arduino codes used for each signal input tested are available in [1].

The experimental procedure consists in defining the DC motor input signal from Table 1 and repeating the stimuli at least ten times, recording the sensors data in a .csv file using Ardu Spreadsheet [7], an Arduino IDE tool. For each tested input, the motor conditions are kept the same. In the case of the motor under load conditions, the platform allows the installation of a pulley and a cable so that the DC motor can pull an experimental load.

The platform was attached to a workbench for the execution of the experiments, and its height has limited the load excursion. Therefore, the structure height has limited the DC motors input stimuli duration. Fig. 14 shows the accessory attached to the motor shaft to adjust the experimental load, and Fig. 15 exhibits the real platform used to record the data.

3. DC Motor Efficiency Estimation

Efficiency in a DC motor is generally defined as the ratio between the power output and power input. During the conversion of electrical energy into mechanical energy, several losses

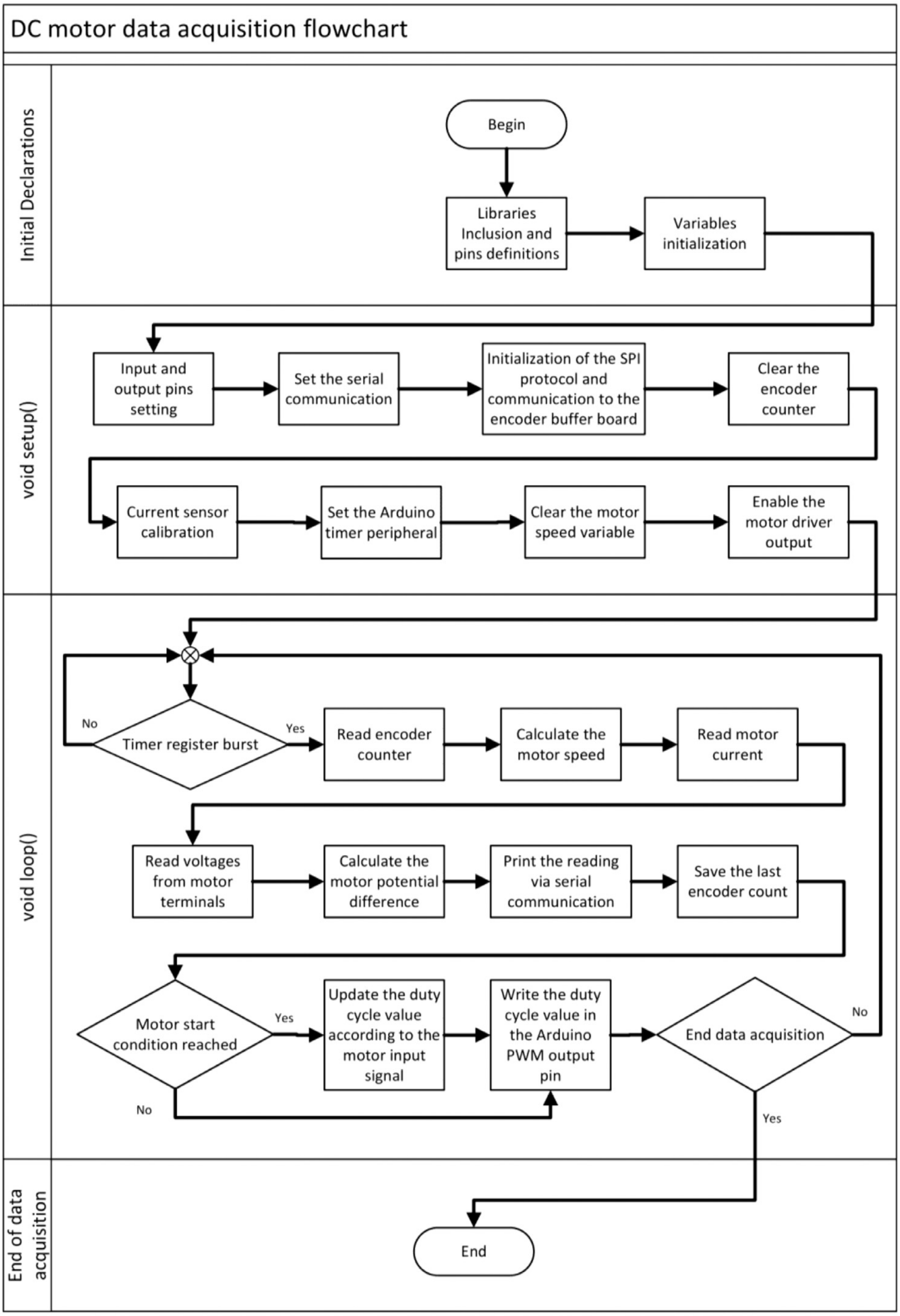

Fig. 13. Flowchart of the data acquisition code embedded in Arduino.

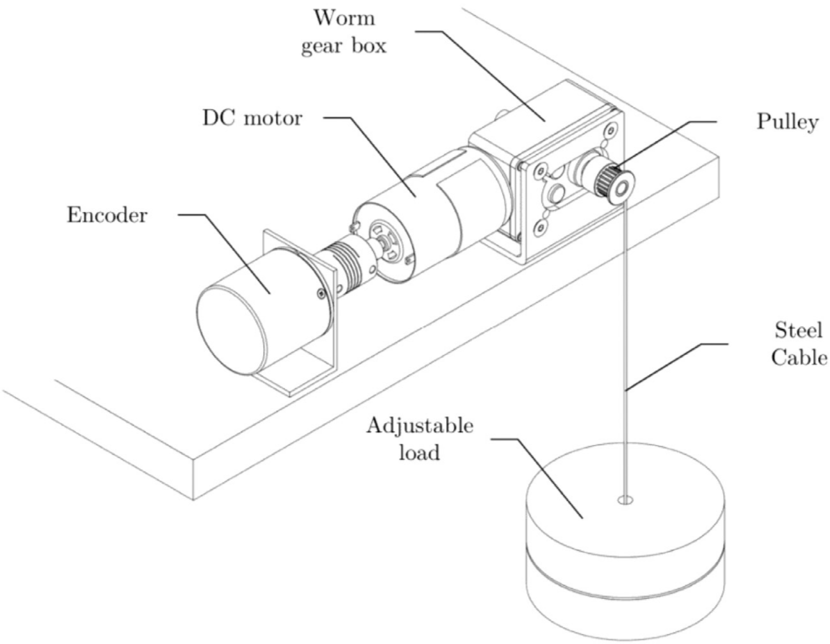

Fig. 14. Load adjustment apparatus.

occur [8]. These losses are electrical, such as copper or magnet, and mechanical, such as friction losses. The experimental platform does not intend to measure them. However, from the data of loaded condition experiments, one can estimate the efficiency of the DC motor.

The loads applied to the motor shaft are known. Considering that the adjustable load inflicts a traction force T to the center of the motor shaft at a 90 degrees angle, as Fig. 16 shows. Disregarding the steel cable mass, the load torque τ_{load} $[N.m]$ is calculated according to Eq. (5), where $m[kg]$ is the mass of the load, g $[m/s^2]$ is the gravitational acceleration, and $d = 6.1\ mm = 0.0061\ m$ is the distance between the center of the motor shaft and the pulley.

$$\tau_{load} = mgd \tag{5}$$

With the estimated load torque value, the power output P_{out} is calculated by Eq. (6), where the motor velocity V from Eq. (1), in RPM, is converted to radians per second, so the final unit is Watts [W].

$$P_{out} = \tau_{load} V \frac{2\pi}{60} \tag{6}$$

Eq. (7) gives the input power P_{in}, in W, where i is current given by Eq. (2), and v_x is the voltage obtained by Eq. (3). Eq. (8) calculates the motor efficiency η, relating the power input and the estimated produced power output.

$$P_{in} = v_x i \tag{7}$$

$$\eta = \frac{P_{out}}{P_{in}} \times 100\% \tag{8}$$

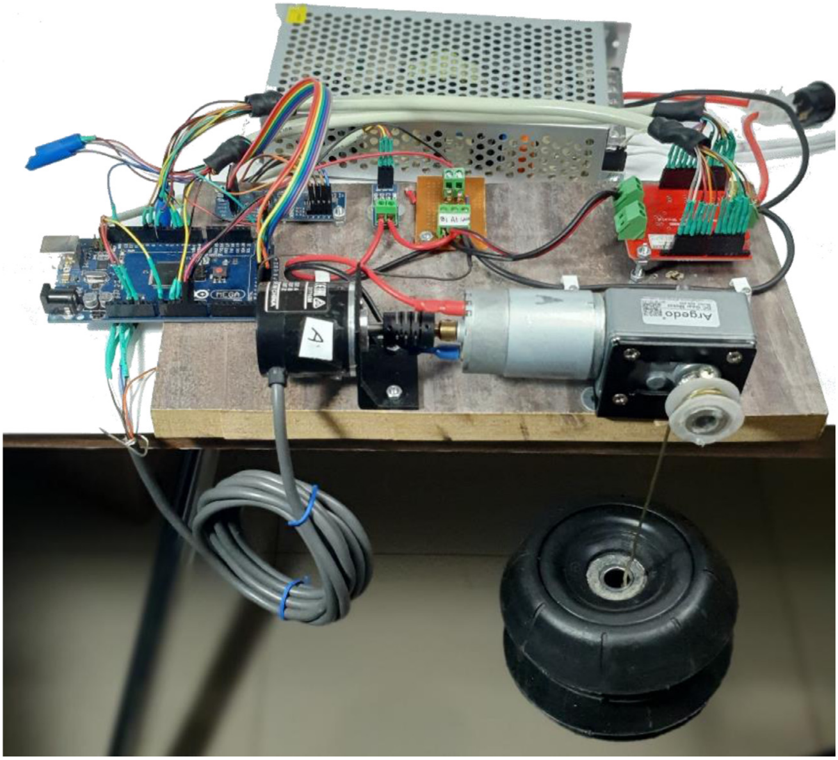

Fig. 15. Experimental platform developed to DC motor experiments.

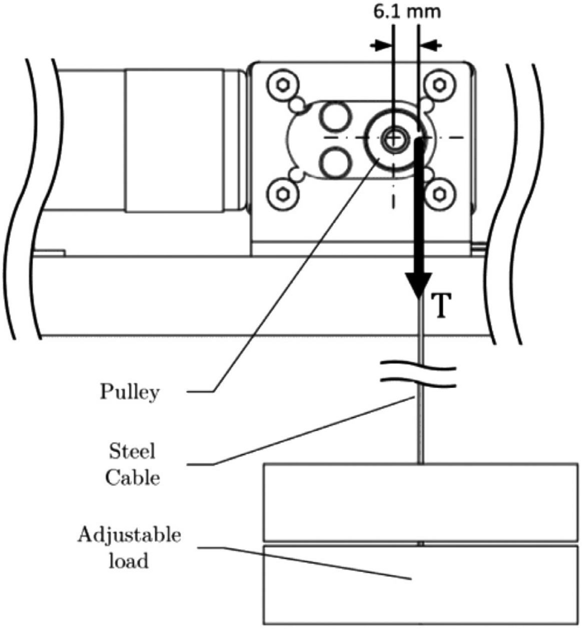

Fig. 16. Study of the torque output developed by the DC motors.

Appendix A. Supplementary data

Supplementary data to this article can be found online at Data in Brief paper's website page or at [9] https://data.mendeley.com/datasets/2rkpsss6fd/2

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships which have, or could be perceived to have, influenced the work reported in this article.

CRediT Author Statement

Wallace Pereira Neves dos Reis: Conceptualization, Methodology, Software, Investigation, Writing – original draft; **Giselle Elias Couto:** Investigation, Writing – review & editing; **Orides Morandin Junior:** Conceptualization, Supervision, Writing – review & editing.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Wallace dos Reis was financed in part by the Federal Institute of Education, Science and Technology of Rio de Janeiro – IFRJ, campus Volta Redonda.

Supplementary materials

Supplementary material associated with this article can be found in the online version at doi:10.1016/j.dib.2022.107802.

References

[1] W. P. N. dos Reis, WallaceP/ArduinoDCMotorDataAcquisition: DC motor data acquisition Arduino files, 2021. doi:10.5281/zenodo.5737539.
[2] R. Koptiev, 2018, ACS712 Arduino Library, URL: https://github.com/rkoptev/ACS712-arduino.
[3] Fully integrated, hall-effect-based linear current sensor IC with 2.1 kVRMS isolation and a low-resistance current conductor, Allegro Microsystems, Rev. (2020) 19.
[4] General-purpose Encoder with External Diameter of 40 mm - E6B2-C, OMRON (2017).
[5] 32-Bit Quadrature Counter with Serial Interface - LS7366R, LSI Computer Systems, Inc., 2007.
[6] Automotive fully integrated H-bridge motor driver, ST Microelectronics, Rev (2008) 8.
[7] I. Luuk, 2021, Logging arduino serial output to CSV/Excel, https://circuitjournal.com/arduino-serial-to-spreadsheet.
[8] K.W. Klontz, Permanent magnet motor with tested efficiency beyond ultra-premium/ie5 levels, in: Proceedings of the ACEEE Summer Study on Energy Efficiency in Industry, 2017, pp. 1–12.
[9] Morandin Junior, Orides; dos Reis, Wallace; Couto, Giselle (2021), "Permanent Magnet Direct Current Worm Geared Motor Data: Voltage, Current, and Speed", Mendeley Data, V2, doi:10.17632/2rkpsss6fd.2