# An on-the-fly DES strategy for the control of brushless motors in electric vehicles

**Gessé Marchioro** [*] **Tiago Possato** [**] **Mauro Pagliosa** [**]
**Marcelo Teixeira** [*]

[*] *Federal University of Technology Parana, Brazil,*
*(e-mail: gesse@alunos.utfpr.edu.br, mtex@utfpr.edu.br).*
[**] *Catarinense Federal Institute, IFC, Brazil,*
*(e-mail: tiago.possato@ifc.edu.br, mauro.pagliosa@ifc.edu.br)*

**Abstract:** Electric vehicles based on brushless motors have emerged as a promising alternative for the automotive industry to replace combustion models. A growing concern, however, is related to energy efficiency both in terms of consumption and recharging time. This fosters the development of on-the-fly auxiliary charging approaches to delay recharging as much as possible. One plausible alternative is harnessing energy from braking, called regenerative braking. This technique requires an enhanced operation of the frequency inverter, as it needs a different switching logic for the braking moments, which makes the programming more complex. This paper tests a *Discrete Event System* (DES) strategy to design, specify, and control a brushless motor, considering the energy savings from the precise handling of regenerative braking events. After modeling and obtaining a DES controller, as a result of applying *Supervisory Control Theory* (SCT) processed over a variable-based automata framework, we further transfer the controller logic to a simulator to quantify gains.

*Keywords:* Discrete Event System; Regenerative Braking; Brushless motor; BLDC.

## 1. INTRODUCTION

Guided by environmental and financial sustainability goals, the automotive industry has been undergoing disruptive changes with the gradual introduction of *Electric Vehicle* (EV) models (Mishra et al., 2022). EVs bring a lower pollutant emission rate as an advantage, standing as an alternative to the use of expensive and finite fossil fuels (Godfrey and Sankaranarayanan, 2018), besides being less noisy and allowing better energy utilization (Kvijayan and Beevi, 2018; Pasupuleti et al., 2022).

*Brushless Direct Current* (BLDC) motors are a good alternative of electric motors for EV, especially because they offer high torque within reduced dimensions (Riyadi and Setianto, 2019). Furthermore, BLDCs reduce friction (delaying maintenance), maximize power, and provide precise speed and torque control. Their quick responsiveness and high energy efficiency make brushless motors suitable especially for personal mobility devices, offering dynamic performance, reduced carbon footprint, and a more efficient driving experience.

Another advantage of BLDC motors is the possibility of implementing regenerative braking to recover energy from the counter electromotive force generated during braking (Riyadi and Setianto, 2019). This means converting kinetic into electrical energy and reusing it to feed the BLDC back. This technique brings significant benefits to EVs, such as savings in replacing mechanical braking devices, as braking can occur without mechanical contact (Dadashnialehi et al., 2015), improvements in energy efficiency, and increased vehicle autonomy (Pasupuleti et al., 2022), which can reach an order between 8-25% (Godfrey and Sankaranarayanan, 2018).

However, frequency inverters are required to drive and control BLDC motors. They perform electronic switching based on the motor's position and the intended action: direct or reverse acceleration or braking. In general, this logic is programmed empirically, based on the orchestration of multiple interdependent actions after every step, which tends to be a complex, exhaustive, and error-prone task. An alternative is to exploit the digital nature of the actions to automate manual programming to some extent.

This paper treats this problem as a Discrete Event System (DES) and applies Supervisory Control Theory (SCT) (Ramadge and Wonham, 1989) extended with variables (Teixeira et al., 2013) to synthesize a controller automatically, focusing on the possibility of energy recovery during braking moments. This controller is then transferred to a simulator to illustrate the methodology. Besides leading to a correct-by-construction controller, our approach is also an ease-to-replace strategy that can be implemented either over a centralized or a distributed hardware infrastructure.

## 2. BACKGROUND

BLDCs are brushless electric motors activated by direct current. Constructively, this type of motor has magnets in its rotor (rotating part) as an alternative to conventional topologies that use rings and brushes to generate an electromagnetic field in the rotating part.

Advantages of BLDCs include (Pasupuleti et al., 2022): high power density (consequently, smaller size and weight);

high starting torque; silent operation; wide speed range; good speed/torque ratio; good energy efficiency; low maintenance requirement and the possibility of regenerative braking. Those features make BLDCs suitable for applications such as aeromodelling, household appliances, and electric vehicles, especially for personal mobility devices like bicycles, skateboards, scooters, wheelchairs, etc.

However, there are some drawbacks that come with the BLDCs. One of them is the high cost and scarcity of raw materials (especially magnets) and the need for a specific device to control the motor - a drive. Torque ripple is also a challenge to be solved, as BLDC motors have trapezoidal current, limiting applications to low power.

### *2.1 Construction of* BLDC

Two basic building blocks compose a BLDC motor: a rotating part (rotor); and a fixed part (stator). In the stator, a rotating electromagnetic field is formed by the current passing through coils. This causes the rotor, which is connected to magnets, to move due to the attraction and repulsion of the magnetic poles.

Most BLDC motors are composed of three sets of coils (phases), which can be accessed externally for their activation and control. Hall effect sensors (magnetic pole sensors) are used to determine how this activation is performed, which allows the rotor position to be determined. Motors with three magnetic sensors are commercially common, allowing the motor's position to be divided into six - with one or two sensors always activated simultaneously.

### *2.2 Driving a* BLDC *Motor*

To energize the stator coils in the correct sequence and at the correct time, a drive (frequency inverter) is required, controlled by a microcontroller, as shown in Fig. 1.

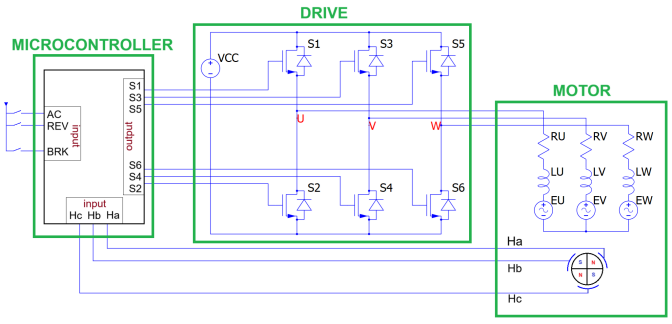



Figure 1. Interconnection: Motor - Inverter - Controller.

The inverter is built on six electronic switches (S1 to S6), which can be MOSFETs or IGBTs, and are activated to control the current flow through phases U, V, and W and, consequently, the generation of electromagnetic fields. The events generated by the inverter are coordinated by a microcontroller based on the current position of the Hall effect sensors (Ha, Hb, and Hc). To make the motor rotate in a specific direction, the operation must occur as shown in Fig. 2 and Fig. 3 (Tang, 2021).

For example, if the motor is rotated between 120° and 180°, with sensors Ha and Hb active, inverter switches

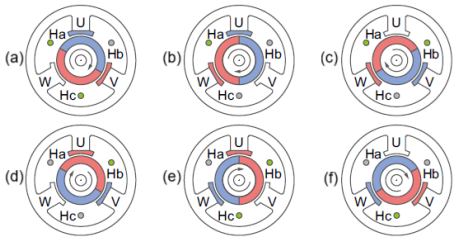



Figure 2. State of Hall sensors for each motor position.

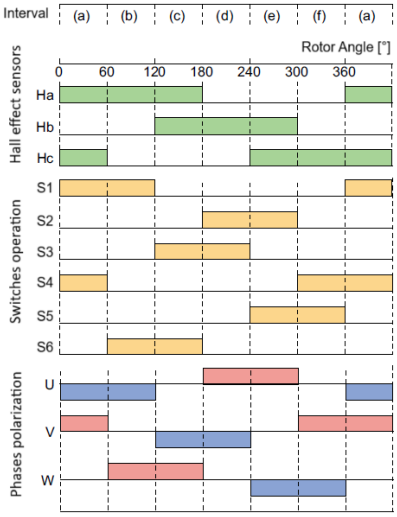



Figure 3. Expected inverter operation for rotor positions.

S3 and S6 should be activated, causing current to flow into phase V and out of phase W. This current generates an electromagnetic field that will move the rotor to the position between 180° and 240°, causing a new combination of switches.

As there is a combination of outputs for each possibility of inputs, a truth table can be formed, as seen in Tab. 1.

Table 1. Truth table - forward acceleration.

| | Inputs | | | Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Ha** | **Hb** | **Hc** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** |
| a | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| b | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| c | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| d | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| e | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| f | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

Tab. 1 holds for moving the motor in a certain direction; however, if one wishes to perform motor movement in the opposite direction, the coil polarization must occur oppositely, requiring a different combination of switches. Thus, the correspondence must assume the Tab. 2.

Table 2. Truth table - backward acceleration.

| | Inputs | | | Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Ha** | **Hb** | **Hc** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** |
| a | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| b | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| c | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| d | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| e | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| f | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

For braking, which occurs electromagnetically, a different logic must also be assumed for the activation of the

electronic switches, and this will also depend on the spinning direction of the motor (Tab. 3 and Tab. 4).

Table 3. Truth table - forward braking.

| | Inputs | | | Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Ha | Hb | Hc | S1 | S2 | S3 | S4 | S5 | S6 |
| a | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| b | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| c | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| d | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| e | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| f | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4. Truth table - backward braking.

| | Inputs | | | Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Ha | Hb | Hc | S1 | S2 | S3 | S4 | S5 | S6 |
| a | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| b | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| c | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| d | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| e | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| f | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

When the vehicle needs speeds or torque control, Pulse Width Modulation (PWM) is applied to the upper switches when accelerating, and to the lower switches when braking. This technique relies on maintaining the switch in an *on* or *off* state based on a duty cycle (proportion of time) instead of operating it continuously. This process is repeated at a frequency significantly higher than the switching of the position sensors. Consequently, it is possible to control the power of the motor when accelerating, and the braking intensity.

The PWM can be controlled in closed-loop to regulate automatically some variables of the motor, like speed or torque. In this paper, the PWM control of the switches will be considered in opened-loop, where the pilot will control vehicle's speed since the focus is on personal devices.

Given the considerable amount of event combinations that have to be orchestrated by the controller, it is reasonable to imagine how hard it would be to program such a controller manually. An alternative is automatically deriving the controller from models, as discussed in the following.

### *2.3 Discrete Event Systems*

*Discrete Event Systems* (DESs) (Cassandras and Lafortune, 2008) define a class of systems that are oriented to events, evolving asynchronously in time. As such, they are more naturally described by languages, whose basic building blocks are *strings* formed by *events*, which are taken from a finite *alphabet* $\Sigma$.

Denote by $\Sigma^*$ the set of all finite *strings* of the form $\sigma_1\sigma_2\ldots\sigma_n$ of events from $\Sigma$, including the *empty string* $\varepsilon$. Then, a subset $L \subseteq \Sigma^*$ is called a *language*.

One way to recognize languages is via *Finite State Machines* (FSMs), which formally is a 5-tuple $A = (\Sigma, X, \rightarrow, x^\circ, X^m)$, where $\Sigma$ is the alphabet; $X$ is the set of states; $x^\circ \in X$ is the initial state; $X^m \subseteq X$ is the subset of marked states; and $\rightarrow \subseteq X \times \Sigma \times X$ is the *transition relation*. In this way, a transition between two any states $x_1, x_2 \in X$ with the event $\sigma \in \Sigma$ is represented by $x_1 \xrightarrow{\sigma} x_2$, and the same follows to $x_1 \xrightarrow{s} x_2$, for $s \in \Sigma^*$.

Sometimes, complex phenomena of DESs may require more powerful FSM models to be expressed, such as the *Extended Finite-State Machines* (EFSMs), which are structures of states similar to conventional FSM, but augmented with formulas that update variables over transitions (Teixeira et al., 2013). A *variable* $v$ has a finite domain $\mathrm{dom}(v)$ and an initial value $v^\circ \in \mathrm{dom}(v)$, and its inclusion on a FSM extends the tuple $A$ to the 7-tuple $A_{\mathrm{v}} = (\Sigma, V, X, Q^\circ, X^m, P_V, \rightarrow)$, where: $\Sigma$ is the alphabet of events; $V = \{v_1, \ldots, v_n\}$ is the set of variables; $X$ is the finite set of states; $Q^\circ \subseteq X$ is the set of initial states; $X^m \subseteq X$ is the set of marked states; $\rightarrow \subseteq X \times \Sigma \times P_V \times X$ is the state transition relation, where $P_V$ is the set of Boolean formulas over $V$.

A transition in $A_{\mathrm{v}}$, from state $x$ to state $y$ with event $\sigma \in \Sigma$ and the update $p \in P_V$, now assume the form $x \xrightarrow{\sigma:p} y$. A formula $p \in P_V$ can either *update* or simply *test* variable values upon transitions. An update $p$ replaces the value carried by the variable in the current state with a new (valid) value in the reached state. Differently, test formulas (or *guards*) do not change any variable value. They simply test values associated with the current state and use the test result to disable or not the transition.

For control, a given *plant* (modelled either by an FSMs or an EFSMs) is expected to be restricted minimally in such a way that a given *specification* is satisfied while disabling only controllable events and not ever blocking.

For handling EFSMs, the use of parallel composition is often necessary, which combines multiple automata into a single system, capturing the combined behavior of the components. The states of the new EFSM are formed by the Cartesian product of the states of the individual EFSMs, and the transitions occur based on the transitions of the component EFSMs, considering shared and individual events (union). Formulas of the transitions (updates and guards) are combined to reflect the conditions and effects of the transitions in the original EFSMs.

*Supervisory Control Theory* (SCT) is a suitable approach to obtaining automatically a solution like this, and it has been defined both for FSMs (Cassandras and Lafortune, 2008) and EFSMs (Teixeira et al., 2013), and implemented in tools like Supremica (Malik et al., 2017).

## 3. A DES STRATEGY FOR BLDC MOTORS CONTROL

The logic that interprets the context and position of the motor and operates the inverter is typically implemented on a microcontroller. Thus, as input and output events are discrete, it is possible to model this logic from the perspective of DESs (Cassandras and Lafortune, 2008), which has finite state automata as a suitable modelling option and SCT as an effective synthesis processing operation that results in a correct-by-construction controller.

### *3.1 Plant Model*

The plant considered in this work includes the frequency inverter and switches, the BLDC motor and its position sensors, and the interface with the pilot or driver (the

intentions to accelerate, brake, and reverse). In this way, it is possible to observe the plant as a set of individual components that include:

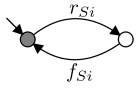
- Six inverter switches, whose behavior is defined by (de)activation events.
- Three position sensors, whose behavior is also based on (de)activation events, but they cannot be controlled. As the motor rotates, different combinations of the three sensors are reached. One or two sensors can possibly be activated simultaneously. The state where none or all sensors are active is also reachable, but only in failure or power loss cases.
- Four operation modes of the frequency inverter (besides standby), which can be enabled or disabled, but only one is activated at a time. These are also uncontrollable events since the driver makes the selection.

The events representing those plant components are shown in Tab. 5. The switches commands $(r_{Si}, f_{Si})$ are the only considered controllable events.

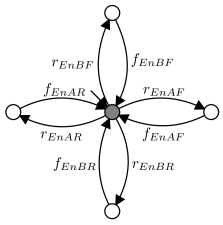
Table 5. Event descriptions and class.

| Event | Description |
|---|---|
| $r_{EnAF}, f_{EnAF}$ | Enable and Disable Forward Acceleration |
| $r_{EnBF}, f_{EnBF}$ | Enable and Disable Forward Braking |
| $r_{EnAR}, f_{EnAR}$ | Enable and Disable Backward Acceleration |
| $r_{EnBR}, f_{EnBR}$ | Enable and Disable Backward Braking |
| $r_{Ha}, f_{Ha}$ | Raise and Fall Signal of Position sensor Ha |
| $r_{Hb}, f_{Hb}$ | Raise and Fall Signal of Position sensor Hb |
| $r_{Hc}, f_{Hc}$ | Raise and Fall Signal of Position sensor Hc |
| $r_{Si}, f_{Si}$ | Enable and Disable Switch Si, for i from 1 to 6 |

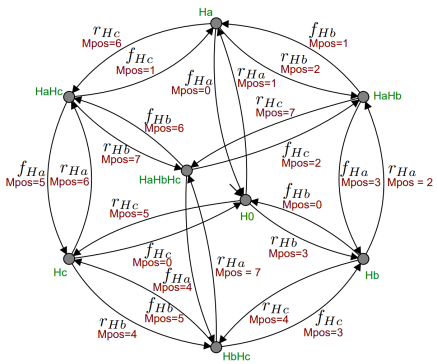
Using this event set, we can introduce the plant model $G$, which results from the automata composition in Fig. 4, i.e., $G = G_{Inv} \parallel G_{Mod} \parallel G_{Pos}$. Supremica tool was used for design, processing, simulation, and tests.




(a) $G_{Inv} = \|_{i=1}^{6} G_{Inv}i$ modeling the inverter.




(b) $G_{Mod}$ modeling the operation mode.




(c) $G_{Pos}$ modeling the position sensors.

Figure 4. Plant model composition.

Each model $G_{inv}i$ in Fig. 4(a) models an inverter switch $i$, for $i = 1, \cdots, 6$. The event $r_{Si}$ represents the rising edge of activation, and $f_{Si}$ is the falling edge.

Model $G_{Mod}$ in Fig. 4(b) represents the selection of the inverter's operating mode, which can be pulling the motor forward or reverse, and braking while spinning forward or reverse. The mode selection depends on the combination of brake and accelerator pedals. In this work, they have been abstracted into events of activating or deactivating such modes. The intensity of braking or acceleration is not considered, but it could, by applying a PWM signal to the selected keys proportionally to the action intensity.

Finally, the automaton $G_{Pos}$ (Fig. 4(c)) designs the dynamics of position sensors, where one or two sensors are always activated simultaneously, with each combination representing a motor position. Cases where all three sensors or none are activated are possible in fault situations. In this model, a variable was employed to register the position of the motor, called $Mpos$, and its domain is an integer from 0 to 7. Tab. 6 describes the correspondence.

Table 6. Correspondence of the Motor Position

| H1 | H2 | H3 | Condition | Mpos |
|---|---|---|---|---|
| 0 | 0 | 0 | - | 0 |
| 1 | 0 | 0 | b | 1 |
| 1 | 1 | 0 | c | 2 |
| 0 | 1 | 0 | d | 3 |
| 0 | 1 | 1 | e | 4 |
| 0 | 0 | 1 | f | 5 |
| 1 | 0 | 1 | a | 6 |
| 1 | 1 | 1 | - | 7 |

### *3.2 Specifications*

The plant is expected to satisfy the following rules:

- $E_{Dis}$ (Fig. 5(a)): if the inverter is on standby, the activation of the switches must be prohibited. However, if there is a command to accelerate or brake, they can be activated based on other specifications.
- $E_{AvoidSC}$ (Fig. 5(b)(c)(d)): if two switches from the same arm of the inverter are activated simultaneously (S1 and S2, S3 and S4, or S5 and S6), a short circuit will occur, potentially damaging them. Therefore, under no circumstances can such activation occur, leading to the creation of three specifications, one to prevent the issue in each of the arms.

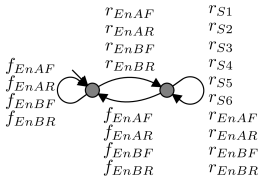
The truth tables are implemented through the following specifications:

- $E_{AF}$ (Fig. 6(a)): rules the activation of the inverter switches when the intention is to accelerate forward (Tab. 1). If this operating mode is disabled, the control of all switches is permitted, leaving the restrictions to be enforced by other specifications. However, when forward acceleration is initiated, only the switches corresponding to the motor position, identified by the variable Mpos, are allowed.
- $E_{AR}$ (Fig. 6(b)): rules the activation of switches when there is reverse acceleration. Again, when not accelerating, the control of all switches is unrestricted. However, during reverse acceleration, the activation is

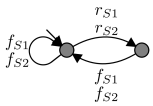
enforced according to the motor position and the logic from Tab. 2.

- $E_{BF}$ (Fig. 6(c)): models the situation where braking is needed when the vehicle is moving forward.
- $E_{BR}$ (Fig. 6(d)): as there might be a need to brake when the vehicle is moving in reverse (such as backing downhill), $E_{BR}$ is for this situation, following Tab. 4.

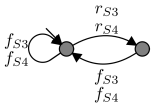
Fig. 5 and Fig. 6 show these automata.



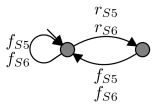
(a) $E_{Dis}$ controlling the inverter in standby.



(b) $E_{AvoidSC1}$: avoid short-circuit - arm 1.

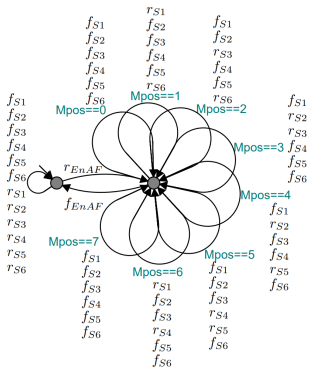


(c) $E_{AvoidSC2}$: avoid short-circuit - arm 2.

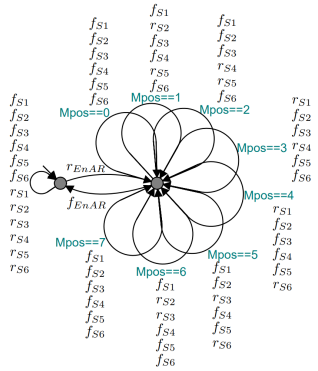


(d) $E_{AvoidSC3}$: avoid short-circuit - arm 3.

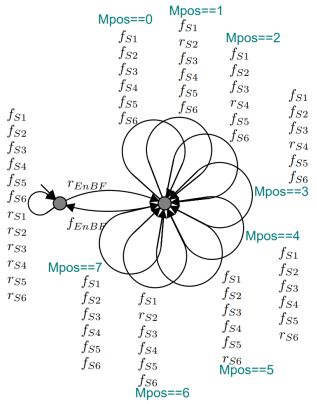
Figure 5. Plant Specifications.



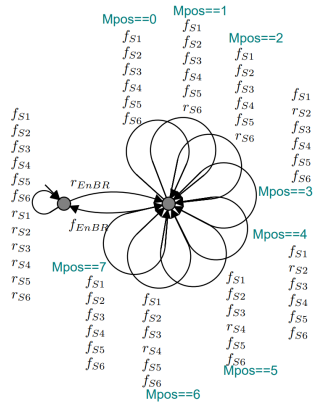
(a) $E_{AF}$: forward acceleration.



(b) $E_{AR}$: reverse acceleration.



(c) $E_{BF}$: forward braking.



(d) $E_{BR}$: reverse braking.

Figure 6. Specifications for the operation modes.

### *3.3 Modeling statistics and simulation*

The plant $G$ is modeled with 2560 states; the specification $E$ with 244 states; and $K$ has 1080 states, the same as $S$.

A simulation was created using Simulink to evaluate the performance of the proposed control system, as it contains pre-configured models of the components used in this project. The simulation (Fig. 7) includes the battery, inverter, brushless motor, and controller. The controller block includes the supervisor obtained from the modelling approach in this paper, which was converted into $C$ code using DEScMaker tool (Possato et al., 2024). DEScMaker utilizes a file from the Supremica software with the synthesized supervisor and exports a $C$ project, which only requires the user to connect the plant events with the function that handles the events.

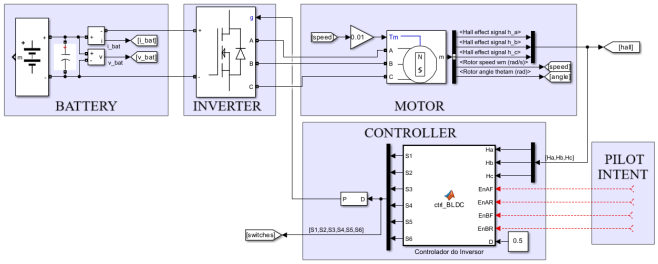



Figure 7. Simulation Blocks of System.

The simulation block settings were generic, maintaining standard characteristics, except for the modifications:

- Battery: nominal voltage of 60V, initially fully charged;
- Inverter: three arms (6 MOSFETs) with characteristics close to ideal;
- Motor: three phases and position sensors.
- Controller: a block that calls functions in $C$ upon the occurrence of events. The outputs are the commands of the inverter switches, and the inputs are the Hall effect sensors and operation commands of the inverter (uncontrollable events). An input for the PWM duty cycle was also added to multiply the switches' output.

The chosen simulation period was 1 second, as it is sufficient, and longer times lead to higher processing demands due to the motor and inverter models. In the simulation, the vehicle is considered stationary until 0.1 s and then accelerated forward from 0.1 s to 0.4 s. Between 0.4 and 0.6 s, no command is sent to the inverter and, subsequently, from 0.6 s to 1 s, regenerative braking is activated.

Fig. 8 illustrates the effects on the vehicle's motor: until acceleration at 0.1 s, the position remains constant, and the speed is null. When acceleration begins, between 0.1 s and 0.4 s, there is an increase in the motor's speed, leading to a gradual increase in its position. Between 0.4 s and 0.6 s, the acceleration ceases, but due to the inertia of the vehicle, there is still positive velocity, although slowing down. After 0.6 s, when regenerative braking starts, the speed decreases at a higher rate, tending towards zero while retaining the motor's position.

Fig. 9 depicts the activation of the Hall effect sensors. Before the acceleration (up to 0.1 s), there is a constant combination. When acceleration begins, the combination starts changing in a frequency proportional to the motor's speed, remaining constant at the end of the simulation when the speed is zero. This image also confirms that always one or two sensors are activated simultaneously.

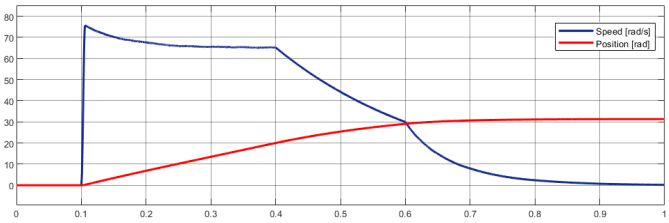



Figure 8. Position and speed during acceleration.

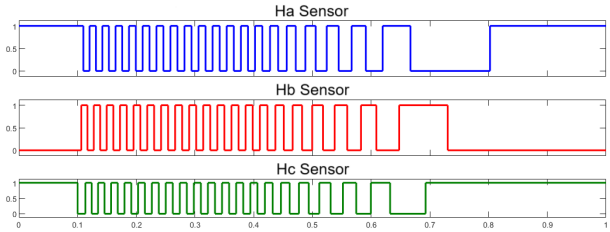



Figure 9. Hall effect sensors during acceleration.

Based on the combination of sensors in Fig. 9, the DES controller activates the switches of the inverter, as shows Fig. 10. The change of combination occurs at a frequency proportional to the motor speed, with one upper and one lower switch simultaneously (from distinct arms) during acceleration and only one lower switch during braking.

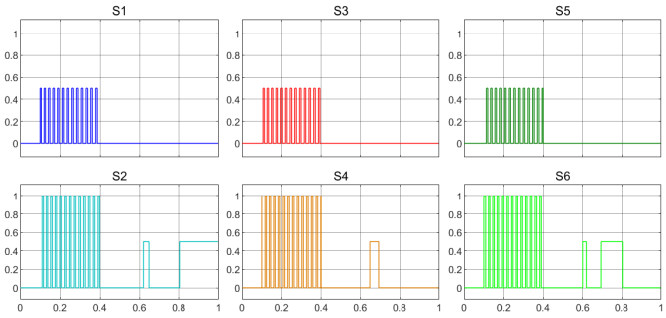



Figure 10. Switch activation during acceleration.

Finally, the effect of the control system on batteries is shown in Fig. 11. The graph confirms that energy is drawn from the battery during acceleration (positive current) and recovered during regenerative braking (negative current).

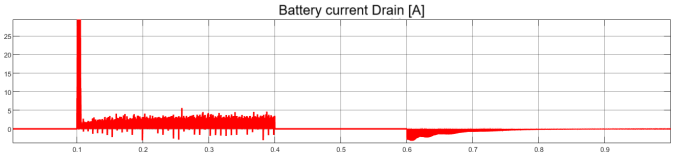



Figure 11. Battery current draining under acceleration.

## 4. CONCLUSION

This paper presents a DES strategy that helps harness energy from regenerative braking in electric vehicles. The approach avoids the need for manually programming switching logic for the combination of motor positions and movements. While manual programming could achieve the same practical result (motor operation) if well-designed and considering the same events, the programmer would need to anticipate a high number of possibilities, as evidenced by the 1080 states of the controller found by the composition of the plants and specifications on Supremica. Additionally, the modularity provided by the DES method makes it easier to modify and understand the logic since the diagram of events and states is graphic and intuitive. This approach has practical appeal, and its potential gains have been proven by simulation.

Future research directions will focus on implementing the generated solution on a microcontroller connected to a real inverter and motor as a replacement for simulations. Integrating our strategy with other algorithms controlling a BLDC motor, such as braking and acceleration, is also possible. In this case, the real system could be really powered with the energy savings extracted from regenerative braking using our approach. This could also exploit topologies for battery systems and electronic circuit options.

## REFERENCES

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, 2 edition.

Dadashnialehi, A., Bab-Hadiashar, A., Cao, Z., and Kapoor, A. (2015). Intelligent sensorless antilock braking system for brushless in-wheel electric vehicles. *IEEE Trans. on Industrial Electronics*, 62(3), 1629–1638.

Godfrey, A.J. and Sankaranarayanan, V. (2018). A new electric braking system with energy regeneration for a BLDC motor driven electric vehicle. *Engineering Science and Technology*, 21.

Kvijayan, P. and Beevi, M.W. (2018). An efficient regenerative braking system for BLDCM driven electric vehicles. In *Int. Conf. for Convergence in Technology*, 1–5. Mangalore, India.

Malik, R., Åkesson, K., Flordal, H., and Fabian, M. (2017). Supremica–an efficient tool for large-scale discrete event systems. *IFAC-PapersOnLine*, 50(1), 5794–5799.

Mishra, A.K., Singh, A.K., and Gururaj, M.V. (2022). A fuel-efficient BLDC motor-driven light electric vehicle with single stage on-board charging system. *IEEE Trans. on Transportation Electrification*.

Pasupuleti, S.S., Tummuru, N.R., and Misra, H. (2022). Power management of hybrid energy storage system based wireless charging system with regenerative braking capability. *IEEE Trans. on Industry Applications*.

Possato, T., Valentini, J.H., Southier, L.F.P., and Teixeira, M. (2024). Automated code generation for DES controllers modeled as finite state machines. In *Formal Methods: Foundations and Applications*, 113–130. Springer Nature Switzerland, Cham.

Ramadge, P.J. and Wonham, W.M. (1989). The control of discrete event systems. *Proc. of the IEEE*, 77(1), 81–98.

Riyadi, S. and Setianto, Y.B.D. (2019). Analysis and design of BLDC motor control in regenerative braking. In *Int. Symp. on Electrical and Electronics Engineering*, 211–215. Ho Chi Minh City, Vietnam.

Tang, J. (2021). Technical manual series: Brushless motor structure and rotation principles. Accessed 15-10-2023.

Teixeira, M., Malik, R., Cury, J.E.R., and de Queiroz, M.H. (2013). Variable abstraction and approximations in supervisory control synthesis. In *2013 American Control Conference, ACC'13*, 120–125. Washington, USA.