Adaptive Fuzzy-PI Control for Speed’s Tracking in a Leisure Electrical Vehicle

This paper deals with a comparison study between different kinds of controllers based on a speed tracking problem inside an electrical vehicle using an axial-flux permanent-magnet traction motor. In fact, the framework of these works is the control of the powertrain of electric vehicles which is a great challenge to ensure driving comfort. To do so, the vehicle’s performances are carried out using an established simulation platform implemented on Matlab-Simulink environment. Then, obtained results are compared for the three synthesized controllers. It has been found that required speed levels are reached with a clear advantage of the adaptive Fuzzy-PI controller versus the PI and the fuzzy counterparts, from the points of view of rise and settling times.

Keywords: Electric Vehicle, Axial Flux Permanent Magnet Motor, PI Control, Fuzzy Control, Adaptive Fuzzy-PI control, Speed Tracking.

Article history: Received 14 March 2016, Accepted 9 August 2016

1. Introduction

Worldwide, transport sector represents nearly 60% of oil consumption and about 30% of carbon emissions [1,2]. According to the IEA, transport will be the major factor leading to an increase of 1% of oil demand by 2030 [1]. Identification of alternative fuel sources and reduction of gas emissions are more relevant than ever. Such a situation argues the growing interest shown in electrical vehicles where several countries set ambitious projects for their adoption. Moreover, electric motors are more efficient and less pollutant than conventional internal combustion engines [3,4].

In addition, Progress in permanent magnets technology makes the synchronous motors a serious candidate to insure vehicle traction [5]. In fact, Compared to asynchronous motors, synchronous ones are more efficient with a higher power density. Moreover, synchronous motors are characterized by an easy speed control and a high positioning precision [6,7]. Thus, bad control of these motors provides a highly corrugated torque leading to numerous electrical and mechanical problems [8].

In automotive traction, several types of controllers can be integrated to guaranty speed tracking, like the famous classic PID which definition needs an explicit mathematical model [8,9]. Such a model is not always possible to obtain, especially when treating a highly nonlinear elements as the vehicle powertrain ones. This can explain the fact that predictive controls like fuzzy ones are more convenient for such cases.

This paper focuses the comparison between a classical PI controller, a Fuzzy controller and an adaptive Fuzzy-PI controller, considering a speed tracking problem inside an electrical vehicle using an axial-flux traction motor. The originality of this work consists of the application of a mixed adaptive controller in traction domain. In first, the components of the traction chain will be described and modeled. Then, in a second time, the correctors
will be synthesized to ensure targeted function. Finally, a comparison is performed around the vehicle performances carried out with each synthesized controller.

2. Notation

The notation used throughout the paper is stated below.

\[ I_d \] : Direct component of the current
\[ I_q \] : In squaring components of the current
\[ u_d \] : Direct component of the voltage
\[ u_q \] : In squaring components of the voltage
\[ \Omega \] : The motor’s speed
\[ T_{em} \] : The electromagnetic torque of the motor
\[ U_{ph1}, U_{ph2}, U_{ph3} \] : the three phase voltage outputs of the inverter
\[ C_1, C_2, C_3 \] : the switches command constants
\[ T_{rB} \] : Friction torque
\[ T_{aero} \] : Aerodynamic torque
\[ T_{g} \] : Gravity torque
\[ T_{a} \] : Acceleration torque
\[ T_{w} \] : Developed wheel torque
\[ T_{r} \] : Load torque
\[ S \] : Laplace operator
\[ G_{e} \] : Input error’s gain
\[ G_{\Delta e} \] : Variation error’s gain
\[ G_{\Delta U} \] : Output’s gain

3. Motor: layout and description

The AFPM topologies are generally of two types. The first one is a motor with two external stator discs managed in one side and in the other of a single rotor disc as illustrated in Fig.1.a) and Fig.2.a). Contrary to the first structure, the second one is a motor with two rotor discs and a single stator disc as shown in Fig.1.b) and Fig.2.b. According to [3] and [6], the second structure is more stable and has less vibration problems and is mechanically more stable, compared to the first one. Moreover, depending on the manner that permanent magnets are placed on rotor discs, the flux linkage can flow through the parts of the magnetic circuit of the motor as shown in Fig.3.a), in the case of a Torus NS, and in Fig.3.b), in the case of a Torus NN which is preferred as it has the advantage of an easier construction of the stator [4-6]. Based on mentioned reasons, the Torus NN is considered in the present work.

The stator of this motor, Fig.4.a), is laminated and enclosed six principle teeth separated with six inserted ones, leading so to twelve slots around which the motor armature is wound. Besides, the two rotor discs are made of massive iron. Each disc contains eight surface-mounted permanent magnets as shown in Fig.4.b). The detailed scheme of a Torus NN with 2 pole pairs is illustrated in Fig.5.
a) Single-rotor - two-stators structure b) Two-rotors - single-stator structure
Fig. 1: Torus configurations of Permanent-magnet axial flux machines.

a) Internal stator b) External stator
Fig. 2: Axial flux permanent magnet motor with internal/external stator. Legend: 1-stator, 2-coils, 3-rotor, 4-magnets, 5-frame, 6-rolling, 7-shaft

a) TORUS NS b) TORUS NN
Fig. 3: Axial-Flux Permanent-Magnet motor type TORUS-NS /NN.

a) Stator b) Rotor
Fig. 4: Parts of the motor.
The optimal design, the appropriate topology and the ability to withstand heavy conditions lead a motor to become reliable in the field of in-wheel applications. The Torus AFPM motor is a serious candidate to fill specific performances with a guarantee of a high power density gathered to a modular structure [6,7]. A Torus NN was dimensioned in [4] to satisfy a leisure vehicle’s power specifications. These latter are remembered with obtained motor sizes and parameters in the appendix.

4. Vehicle’s powertrain: description and modeling

Due to registered progress in calculation and soft tools, simulation can be today a very efficient mean for the test of engineering solutions. Nevertheless, established models have to be sufficiently detailed to take into account interaction between parameters on which the control strategy is based. In the present case, it is Matlab-Simulink numerical tools for power systems and dynamic modeling that are used.

In order to meet Road Traffic Code norms, proposed vehicle has to satisfy, among others, dynamic criteria to guarantee required level of stability. Indeed, if we recall that the vehicle speed is a direct consequence of the immediate effort applied on wheels, it is therefore important to stabilize this speed with the right torque and power levels.

As shown in Fig.6, traction chain going to be studied consists of: supplying batteries connected to the traction motor through a conventional six switch inverter and a mechanical gear intended for power transmission to the vehicle’s wheels. The bloc diagram showing the connection between the powertrain components and controllers is illustrated by Fig.7 and its established model implemented on Matlab-Simulink environment is shown in Fig.8. In what follows, each bloc of the vehicle model will be described.
4.1. Traction motor

Studied axial flux motor is synchronous with non-salient poles and the analytical model of its electrical and mechanical parts can be described in the (dq) frame as in equation (1), [6]:

\[
\begin{bmatrix}
    I'_d \\
    I'_q \\
    \Omega
\end{bmatrix}
= \begin{bmatrix}
    \frac{-R_s}{L_s} I_d + pI_q \Omega \\
    \frac{-R_s}{L_s} I_q - pI_d \Omega - \frac{p\Psi_a}{L_s} \Omega \\
    \frac{p\Psi_a}{J} I_q - \frac{f}{J} \Omega
\end{bmatrix}
+ \begin{bmatrix}
    1 & 0 & 0 \\
    0 & \frac{1}{L_s} & 0 \\
    0 & 0 & -\frac{1}{J}
\end{bmatrix}
\begin{bmatrix}
    u_d \\
    u_q \\
    T_{em}
\end{bmatrix}
\]

(1)

4.2. Inverter

It is a conventional three phases six switched inverter controlled using PWM. The expressions of its output voltages are given by equation (2).

\[
\begin{align*}
U_{ph1} &= \frac{U_{dc}}{3} (2C_1 - C_2 - C_3) \\
U_{ph2} &= \frac{U_{dc}}{3} (2C_2 - C_1 - C_3) \\
U_{ph3} &= \frac{U_{dc}}{3} (2C_3 - C_2 - C_1)
\end{align*}
\]

(2)
4.3. Dynamic model of the vehicle

Fig. 9 illustrates resistant forces applied on a vehicle. Thus, for motion, the automotive is required to develop a useful wheel torque to overcome these resistant components. The dynamics fundamental law applied on the vehicle gives the motion equation (3), [3,8,9]:

\[ T_w = T_{rB} + T_{aero} + T_g + T_a = T_r + T_a \]  

Where:
- \( T_{rB} \) is the resistant torque corresponding to the friction at bearings (equation (4)).
- \( T_{aero} \) is the resistant torque corresponding to aerodynamic effect (equation (5)).
- \( T_g \) is the resistant torque corresponding to the gravity effect (equation (6)).
- \( T_a \) is the resistant torque corresponding to the acceleration (equation (7)).

\[ T_{rB} = R_r f_r M_V g \]  

\[ T_{aero} = R_r V^2 \frac{M_{va} C_x S_f}{2} \]  

\[ T_g = R_r M_V g \sin(\lambda) \]  

\[ T_a = \sigma M_V R_{roue}^2 \frac{dV}{dt} \]  

Fig. 9: Forces on vehicle during motion. Legend: \( F_g \): gravity force, \( F_b \): bearing force, \( F_{aero} \): aerodynamic force.

Now, after modeling separately the components of powertrain, it seems imperative to synthesize the controller going to be used in order to ensure desired performances. In the present work, we are interested in a speed tracking case where the vehicle is required to fulfill a specified speed profile. To do so, a comparative study based on vehicle’s performances is achieved between three controllers: a conventional PI, a fuzzy and an adaptive fuzzy-PI one.
5. Synthesis of a PI controller

Our first choice is focused on the PI corrector because it is the most popular and common used corrector in industrial application. To determine the controller parameters, the transfer function of the whole traction chain has to be carried out through an identification phase. Indeed, the presence of several non-linear blocks with interconnected parameters makes difficult to explicit a mathematical model.

In the present work, the identification is performed on a single bloc formed by all traction chain’s elements, where the in-quadrature component of the current $I_q$ is the system input and the vehicle’s speed is its output as shown in Fig.10.

For different values of $I_q$, the speed of the vehicle is recorded and represented in Fig.11. Besides and due to the linear evolution of the system’s output in zone 2, Recursive Least Squares ‘RLS’ is adopted to identify the traction chain transfer function given by equation (8).

$$G(s) = \frac{1.1}{s + 93}$$  \hspace{1cm} (8)

Considering the bloc diagram of Fig.7 and the usual transfer function of a PI corrector, equation (9), the transfer function of the closed loop system can be written as in equation (10). To calculate the parameters of the PI corrector, the pole placement method is adopted. Moreover, to guarantee driving comfort, the vehicle should be stable with non-noising vibrations levels due to over speed perturbations. Thus, closed-loop response is chosen with an aperiodic shape. To satisfy such a condition, the corrector’s parameters values are $K_c = 845, 39$ and $T_i = 99, 89$.

$$C(s) = K_c \times \left(1 + \frac{1}{T_i \times s}\right)$$  \hspace{1cm} (9)

$$G_{bf}(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}$$  \hspace{1cm} (10)

Fig.10: Principal of the identification of the traction chain’s transfer function.

Fig.11: Static study curve.
6. Synthesis of a fuzzy controller

In the case of classical control, where PI is used, the analytic expression is needed to settle on the controller parameters. To do so, as it was presented in the previous paragraph, we must resort to alternative solutions as “identification” which covers only a limited zone. Nevertheless, a predictive control is favored since the reaction is based on predefined rules related directly to the behavior of the considered system. In the present section we are interested on fuzzy logic control which basic structure is illustrated in Fig.12, [10,11], where:

- The “fuzzification interface” converts the range of input's values (non-fuzzy) into the corresponding Universe of Discourse (fuzzy).
- The “knowledge base” is formed by two parts:
  - Rule base: determines the goals and the desired behavior of the process.
  - Data base: contains the discretization definition, normalization of discourse universes, and Membership Functions definition.
- The “inference procedure”: using the rules of inference, it processes input data to infer fuzzy control actions.
- The “defuzzification interface”: converts the range of values of universes into their corresponding output variables which means the transformation of a fuzzy action into a 'non-fuzzy' (crisp) control action.

Fig.12: Basic structure of fuzzy logic control.

6.1. Application of Mamdani fuzzy controller

This phase is devoted to the definition of functions and rules of the fuzzy controller. Adopted strategy is based on both: the error between predetermined speed reference and vehicle speed ‘e’, and the evolution of such error ‘Δe’. Thus, the controller’s inputs are: the vehicle speed (process output), the desired speed sequence (reference), the error value “\(e_1\)” at time \(t_1\) and the error value “\(e_2\)” at time \(t_2\).
A) Membership Function

Triangular shapes are assigned to the Membership Functions (MFs) as shown in Fig. 13. In addition, seven fuzzy subsets are chosen for controller’s input variables \( e \) and \( \Delta e \) and for its output \( I_{q_{ref}} \): Positive Big (PB), Positive Medium (PM), Positive Small (PS), Zero (ZE), Negative Big (NB), Negative Medium (NM), Negative Small (NS).

B) Decision Rules

These rules consist of the fuzzy conditional statements, acting according to the subsequent structure: “If \( e \) is \( A \) AND \( \Delta e \) is \( B \), THEN \( \Delta U \) is \( C \), where \( U \) is the input command “\( I_q \)”. Defined rules should bring the system’s situation corresponding to the values of “\( e \)” and “\( \Delta e \)” and their matching “\( \Delta U \)” value.

In the present application, the adjustment of the vehicle speed depends largely on fixed inferences corresponding to both static and dynamic behavior of the system. Thus, considered decision rules are based on Mac-Vicar Whelan’s table where Max-Min inference method is used, [12]. Besides and for more flexibility of the controller, the universe of discourse for each input crisp is “normalized” to the interval \([-1, +1]\).

C) Defuzzification method

For the present application, we choose the widely used defuzzification method known as “centre of gravity weighted by heights” which is easy to implement and does not require much calculation.

6.2. Fuzzy controller bloc model

Adopted strategy is simply based on the variation of different gains enclosed in fuzzy controller bloc of Fig. 14 until obtaining the accurate desired speed. Several tests have been realized and the best performances were obtained by the following gains values:

- Input error’s gain: \( G_e = 1/90 \).
- Variation error’s gain: \( G_{\Delta e} = 1/23 \).
- Output’s gain: \( G_{\Delta U} = 10000 \).
7. Synthesis of an adaptive Fuzzy-PI controller

The present paragraph represents the originality of these works. Our objective is to combine the previous techniques in order to benefit of their advantages and discard their limitations. To do so, the proposed controller has a structure based on a fuzzy supervisor acting on PI parameters as illustrated in Fig.15. Indeed, the proportional coefficient $K_p$ and the integral one $K_i$ are no longer constants but scalable: $K_p \in [K_{p\min}, K_{p\max}]$ and $K_i \in [K_{i\min}, K_{i\max}]$, [12]. In fact, the proportional action is intended to correct instantaneously the gap between the reference and the speed and the integral action is devoted to eliminate the static error, [12,13]. Nevertheless, when tuning the PI parameter, the following aspects should be considered:
- The increase of $K_p$ leads to ameliorate the rapidity of the system with a risk of large overshoot,
- The reduction of the integral action gives a low error but at the cost of a possible mismanagement of stability.

To realize the desired aim of the supervisor, [13], the FLC-PI is composed of:
- Same 7 subsets for inputs and membership functions as used in paragraph 5,
- 2 additional subsets are defined for the two outputs $K'_p$ and $K'_{i}$. These ones have a singleton form as shown in Fig.16. Their linguistic levels are: 'H' (referring to High) and 'L' (referring to Low).
To normalize the PI parameters into [0, 1], we simply use the transformations given by equations (11) and (12).

\[
K'_p = \frac{K_p - K_{p\text{ min}}}{K_{p\text{ max}} - K_{p\text{ min}}} \\
K'_i = \frac{K_i - K_{i\text{ min}}}{K_{i\text{ max}} - K_{i\text{ min}}}
\] (11) (12)

The decision mechanism established for fuzzy supervisor's outputs is detailed in Tables 1 and 2.

**Table 1: Decision table for K'_p**

<table>
<thead>
<tr>
<th>e(\Delta e)</th>
<th>NB</th>
<th>NM</th>
<th>NP</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>NM</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>NP</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Z</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>PS</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>PM</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>PB</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
</tbody>
</table>

**Table 2: Decision table for K'_i**

<table>
<thead>
<tr>
<th>e(\Delta e)</th>
<th>NB</th>
<th>NM</th>
<th>NP</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>NM</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>NP</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>Z</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>PS</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>PM</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>PB</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>
The proposed Fuzzy-PI controller performs as the following:

- The fuzzy supervisor with its 'expertise' treats the two input crisps, specifies the membership degrees of each one into the fuzzy sets, identifies premises, and locates the condition in the inference table in order to assign the right decision to apply for $K'_p$ and $K'_i$, as shown in Fig.17.

- The PI control is based on the reverse transformations of equations (13) and (14) to have the real time gain (proportional and integral actions):

$$K_p = (K_{p\text{max}} - K_{p\text{min}})K'_p + K_{p\text{min}}$$  \hspace{1cm} (13)

$$K_i = (K_{i\text{max}} - K_{i\text{min}})K'_i + K_{i\text{min}}$$  \hspace{1cm} (14)

7. Simulations tests of speed tracking

The present paragraph is intended for a comparison between the synthesized controllers, based on vehicle performances in the case of speed tracking. Used simulation parameters are given in Tables 3, 4 and 5 (Appendix).

7.1 Test under a scaled speed sequence

The first test is performed under a scaled speed sequence and the vehicle speed is illustrated in Fig.18. Analyzing obtained results, we can remark that desired speed levels are reached in less than 4 seconds as required in the vehicle specification for the three controllers. Nevertheless, the following differences are noted:

- An overshoot of 0.7 % is registered and can’t be avoided by the PI controller.
- An overtaking of 2.3 % appears when the fuzzy controller is used. However, this latter recovers it progressively.
- No overshoot with the FLC-PI controller: the speed reaches the steady state without exceeding the reference and the static error was very low (about $10^{-4}$).
- A prominent amelioration of the 'rise time' is registered through the use of the PI, the fuzzy then the FLC-PI controller, respectively.
- The settling time is refined by the use of the FLC-PI.

To conclude about this first test, we can confirm that, compared to the PI and the fuzzy controllers, the FLC-PI has a clear advantage from the point of view of 'precision' and 'rapidity'.
7.2 Test under a driving cycle sequence

In the case of the second test, the required speed sequence is an INRETS driving cycle [14]. It is a normalized sequence composed by several speed levels, used by the French institute of science and technology for transport, development and networks (IFSTTAR) for use in the measurement of road vehicle emissions before their commercialization [14, 15].

Referring to obtained results, shown in Fig.19 and Fig.20, we can say that:

- In reference sequence, the maximum required speed does not exceed 76km/h,
- The vehicle follows desired speed reference and reaches demanded levels.
- Sudden changes in required speed level cannot be fulfilled with the same precision by the PI and the fuzzy controllers, unlike the case of a tracking with the FLC-PI where the speed follows fluently its reference.
8. Conclusion

To promote electric vehicles the control of the powertrain is a raised challenge to ensure driving comfort. It is the framework of this paper devoted to a comparison study between three controllers, considering a speed tracking problem inside an electrical vehicle using an axial-flux permanent-magnet traction motor. Such motor encloses a single stator disc and two rotor ones leading to the so called Torus-NN motor with Inner Stator.

In first, all parts of the vehicle powertrain were described and analytically modeled to establish a simulation platform on Matlab-Simulink environment. Then, in order to insure speed tracking, three controllers were synthesized and integrated in established model to carry out the vehicle performances under two different speed sequences: a scaled sequence in first, then a driving cycle. Comparing obtained results we can conclude that desired speed levels are reached in less than 4 seconds as required by the vehicle specification for the three controllers. However, sudden speed changes cannot be fulfilled with the same precision by the PI and the fuzzy controllers, unlike the case of the tracking with the FLC-PI where the speed follows fluently its reference. In fact, a prominent amelioration is registered through the use of the PI, the fuzzy then the FLC-PI controller, respectively, from the points of view of rise and selling times.

Finally, authors are aware that other controllers could be used to ensure the speed tracking in an electrical vehicle like sliding mode, robust control …. Thus, other comparisons should be conducted to ensure the best control without driving inconveniencies.

References


## Appendix

### Table 3: Vehicle and motor specifications

<table>
<thead>
<tr>
<th>Parameter (Vehicle)</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>Parameter (Motor)</th>
<th>SYMBOL</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal power</td>
<td>$P_n$</td>
<td>15 kW</td>
<td>Resistance per phase</td>
<td>$R$</td>
<td>0.0076 $\Omega$</td>
</tr>
<tr>
<td>Starting torque</td>
<td>$C_d$</td>
<td>58.8 N.m</td>
<td>Cyclic inductance</td>
<td>$L$</td>
<td>6.7E-004 H</td>
</tr>
<tr>
<td>Basic velocity</td>
<td>$V_b$</td>
<td>30 km/h</td>
<td>Direct inductance</td>
<td>$L_d$</td>
<td>0.157 mH</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>$V_{max}$</td>
<td>80 km/h</td>
<td>In squiring inductance</td>
<td>$L_q$</td>
<td>0.157 mH</td>
</tr>
<tr>
<td>Vehicle mass</td>
<td>$M_v$</td>
<td>800 kg</td>
<td>Pole pair number</td>
<td>$p$</td>
<td>4</td>
</tr>
<tr>
<td>Wheel ray</td>
<td>$R_w$</td>
<td>0.26 m</td>
<td>Direct voltage</td>
<td>$U$</td>
<td>300 V</td>
</tr>
<tr>
<td>Inertia Coefficient</td>
<td>$\sigma$</td>
<td>1</td>
<td>Viscous friction coefficient</td>
<td>$f$</td>
<td>0.043 N.m.s/rad</td>
</tr>
<tr>
<td>Switched frequency</td>
<td>$f_c$</td>
<td>0.15 m</td>
<td>Rotor internal diameter</td>
<td>$D_{int}$</td>
<td>0.15 m</td>
</tr>
<tr>
<td>Reduction report</td>
<td>$r_d$</td>
<td>4</td>
<td>Rotor external diameter</td>
<td>$D_{ext}$</td>
<td>0.35 m</td>
</tr>
<tr>
<td>Stating time</td>
<td>$t_d$</td>
<td>4 s</td>
<td>Rotor mass</td>
<td>$M_r$</td>
<td>10.3262 kg</td>
</tr>
</tbody>
</table>

### Table 4: Electric and dynamic parameters used in simulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SYMBOL</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drag coefficient</td>
<td>$C_d$</td>
<td>0.55</td>
</tr>
<tr>
<td>Frontal surface</td>
<td>$S_f$</td>
<td>1.8 m²</td>
</tr>
<tr>
<td>Coefficient to bearing pneumatic</td>
<td>$f_r$</td>
<td>0.01</td>
</tr>
<tr>
<td>Electric constant</td>
<td>$k_e$</td>
<td>0.3</td>
</tr>
</tbody>
</table>