Comparison of Metaheuristic Optimization Algorithms for Electromechanical Actuator Fault Detection

Comparison of Metaheuristic Optimization Algorithms for Electromechanical Actuator Fault Detection

Matteo D. L. Dalla Vedova

DIMEAS, Politecnico di Torino, Italy. E-mail: matteo.dallavedova@polito.it

Pier Carlo Berri

DIMEAS, Politecnico di Torino, Italy. E-mail: matteo.dallavedova@polito.it

Javier Bravo Lendínez

Dept. of Bioengineering and Aerospace Engineering, UC3M, Madrid, Spain. E-mail: jblendinez@gmail.com

Model-based Fault Detection and Identification (FDI) for prognostics rely on the comparison between the response of the monitored system and that of a digital twin. Discrepancies among the behavior of the two systems are analyzed to filter out the effect of uncertainties of the model and identify failure precursors. A possible solution to identify faults is to leverage a model able to simulate faults: an optimization algorithm varies the faults magnitude parameters within the model to achieve the matching between the responses of the model and the actual system. When the algorithm converges, we can assume that the fault parameters that produce the best match between the system and its digital twin approximate the actual faults affecting the equipment. The choice of an optimization algorithm appropriate for the task is highly problem dependent. Algorithms for FDI are required to deal with multimodal objective functions characterized by poor regularity and a relatively high computational cost. Additionally, the derivatives of the objective function are not usually available and must be obtained numerically if needed. Then, we restrict our search for a suitable optimization algorithm to metaheuristic gradient-free ones, testing Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Grey Wolf Optimization, Dragonfly Algorithm, and Whale Optimization Algorithm. Their performances on the considered problem were assessed and compared, in terms of accuracy and computational time.

Keywords: EMA, Fault Detection and Identification, FDI, Metaheuristic Optimization Algorithms, PHM.

1. Introduction

In recent decades, brushless DC electric motors (BLDC) have found widespread use in aerospace due to new design philosophies aiming to install more onboard electrical power for aeronautical systems (as reported by Rosero et al. 2017). This technological objective is supported by various reasons, including the reduction of fuel consumption and air pollution and a reduction in the relative maintenance costs. Electric motors, often used in low power systems, drones and secondary flight controls, are hence slowly replacing the traditional hydraulic and pneumatic actuators also in the primary flight control systems. This solution, as shown by Berri et al. (2017), reduces the overall weight and energy budget of the entire flight control system and improves its logistical availability, thanks to more straightforward and faster maintenance (e.g. electric machines are not subject to fluid leaks or contamination problems which, instead, are common to hydraulic systems and which, in general, are challenging to detect and repair, due to the poor accessibility of these systems). On the contrary, as reported by Dalla Vedova et al. (2019c), electromechanical actuators (EMAs) are not yet able to operate as primary flight controls of large aircraft or in case of extreme working conditions (e.g. high values of external load, actuation speed or operating temperature). These shortcomings are due to significant issues related to their response/resilience to failures, as well as the motor thermal control and sensitivity to electromagnetic interferences (EMC and ESD problems). Therefore, according to Vachtsevanos et al. (2006), mainly due to the influence of these critical applications on the safety of the aircraft, the Prognosis and Health Management (PHM) disciplines can make safer EMA operating conditions. As shown by Byington et al. (2004), PHM is an emerging discipline aiming to exploit a continuous estimation of the Remaining Useful Life (RUL) of the system, optimizing the maintenance interventions. The RUL assessment, aiming to detect the actual state of health of the system, is performed in the Fault Detection and Identification (FDI) phase of the prognostic analysis. The FDI procedure can be conceived as an optimization problem, minimizing the error between an output measured by the real system and one calculated by a numerical model able to take into account selected failure modes.

As reported by Wolpert and Macready (1997), many different optimization algorithms are available in literature, but none is universally suitable for all problems; hence, the choice of an acceptable optimization strategy is strongly problem-dependent. In this paper, authors focus on the FDI task for a typical on-board Electromechanical Actuator (EMA), using bio-inspired meta-heuristic algorithms. These are characterized by good robustness, primarily when the objective function is little known, although the convergence is generally rather slow compared to deterministic algorithms based on the gradient (Dalla Vedova et al., 2019b). For this purpose, in this work, the authors compare six different algorithms, namely:

- Genetic Algorithm (GA), Holland (1973)
- Particle Swarm Optimization (PSO), as proposed by Kennedy and Eberhart (1995)
- Differential Evolution (DE), Storn and Price (1997), and Lones (2014)
- Grey Wolf Optimization (GWO), as shown by Mirjalili et al. (2014)
- Dragonfly Algorithm (DA), Mirjalili (2015)
- Whale Optimization Algorithm (WOA), according to Mirjalili and Lewis (2016)

Their applicability to the considered case study and the performance thus obtained, both in terms of accuracy and calculation time, are examined and critically compared with each other in the following sections of this work.

2. Considered Case Study

According to Dalla Vedova et al. (2019c), Figure 1 shows the layout of the EM actuation system for secondary flight commands used as a testbed in this work; it consists of four main subsystems.
The **Actuator Control Electronics** (ACE) implements the control law aimed at minimizing the instantaneous value of the error calculated between the EMA actual position and the commanded one (by closing the EMA position feedback). This corrective action produces a control input that drives the **Power Drive Electronics** (PDE). In this work, the PDE is composed of a three-phase inverter bridge regulating the electrical power provided to the EM motor (that, in this case, is a three-phase brushless DC motor). The **BLDC motor** converts electrical energy into mechanics; its output shaft is usually connected to the aerodynamic control surface through mechanical transmission (often consisting of a gearbox which drives a ball screw or a roller screw).

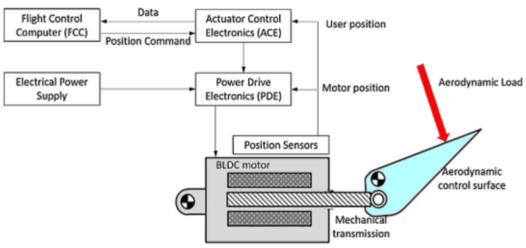

Fig. 1. Schematic of the considered flight-by-wire actuation system equipped with an EMA (Dalla Vedova et al., 2019b).

The position feedback is provided by a dedicated network of **Position Sensors** (PS) which detects the actual position of the final actuator (for control purposes), as well as phase currents, rotation speed and angular position of the BLDC motor (to close the control loops and command the switching of the three phases of the device). It should be noted that this EMA architecture is widely described by Dalla Vedova et al. (2015).

3. EMA Reference and Monitoring Models

In this paper, novel FDI methods useful for perform reliable EMA prognostics are proposed. Failures are evaluated using a High-Fidelity (or Reference Model, RM) Matlab-Simulink model implementing the considered faults, logging relevant data such as user position, motor angular speed and phase currents. Given the detailed system modeling, the measurements obtained are assumed to be those of a real system. Optimization is then performed using a lighter, approximated Monitor Model (MM), in order to best fit the reference current signal. After convergence is achieved, an estimation of the faults imposed in the reference model is then obtained. The MM is in fact a simplified version of the RM and is very useful to quickly perform the optimization based on genetic algorithms.

Figure 2 shows a schematic of the RM; it is composed by four main functional subsystems.

Com block represents the command given to the motor in terms of radial position.

ACE block (i.e. Actuator Control Electronics) implements the electronics system control laws.

BLDC Motor EM Model block simulates the electromagnetic (EM) model on the BLDC motor, evaluating mechanical torque as function of the reference current.

BLDC Motor Dynamic Model evaluates the instantaneous values of the positions and the actuation speeds of the BLDC motor and the moving surface, using them as feedback input for the EMA controller.

The RM, formerly proposed by Dalla Vedova et al. (2015), simulates the dynamic response of a typical on-board EMA equipped with a three-phase BLDC trapezoidal motor.

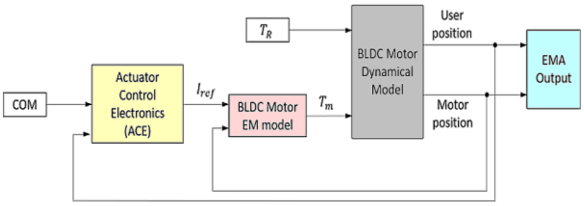

Fig. 2. EMA reference model (RM) block diagram.

The MM block diagram is shown in Figure 3; a detailed description can be found in Berri et al. (2016). It must be noted that the three phase BLDC motor of the RM, in MM becomes an equivalent single-phase DC motor with a single feedback loop. To account for the same fault modes of the RM dealing with this simplified representation, Berri et al. (2018) introduced a shape function-based model for the simulation of the electrical faults. Although not strictly related to the physics of the system, the approximation allows reproducing faults effects with suitable accuracy and reduced computational cost.

4. Considered EMA Progressive Failures

Five typical EMA fault modes, all characterized by progressive growth, have been considered. Their selection was made considering the criticality of the different EMA failure modes (assessed simultaneously in terms of severity of consequences and frequency of failure events), as described by Chesley (2011). Dry friction, usually caused by contamination or progressive wear of bearings or mechanical joints, require an accurate modelling given the high intrinsic non-linearity, as developed by Borello and Dalla Vedova (2012). Wear phenomena can also determine the rise of mechanical backlashes (e.g. on gearboxes, transmissions, hinges and ball screws) that are simulated by lumped parameters models proposed by Borello et al. (2009, 2014). Short circuits (SC) affecting BLDC stator coils and rotor static eccentricities (RSE), i.e. misalignments between rotor and stator due to bearings wear or poor manufacturing tolerances, are modelled as shown by Berri et al. (2016). Control electronics faults, according to Ginart et al. (2007), are modelled in a simplified way simulating a PID proportional gain drift.

5. Fault Detection/Identification Method

The fault identification task is performed by an optimization algorithm that compares output signals measured from the actual system (or from the Reference Model, for testing purposes) to a simplified digital twin (i.e. the Monitoring Model). The algorithm iteratively changes a set of parameters encoding the faults simulated by the MM, in order to minimize an error between the two models. The output signal to analyze with FDI algorithms must be significantly dependent on the system health condition; for the particular application shown in this paper, the motor currents were chosen. Since the RM features a three-phase electric model, while the MM leverages a single-phase equivalent, the envelope of the three RM phase currents is compared to the single MM current signal. The choice of an optimization algorithm adequate for a given task is highly problem dependent (Binitha 2012). This problem is multi-modal and characterized by a computationally expensive objective function and by a high dimensionality; in such conditions, deterministic search algorithms are usually unsuitable, while more robust meta-heuristic strategies are preferred. In this work, we assess and compare six bio-inspired meta-heuristic methods, namely: Differential Evolution (DE), Dragonfly Algorithm (DA), Genetic Algorithm (GA), Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO) and Whale Optimization Algorithm (WOA). A brief description of each algorithm is provided in the following sections.

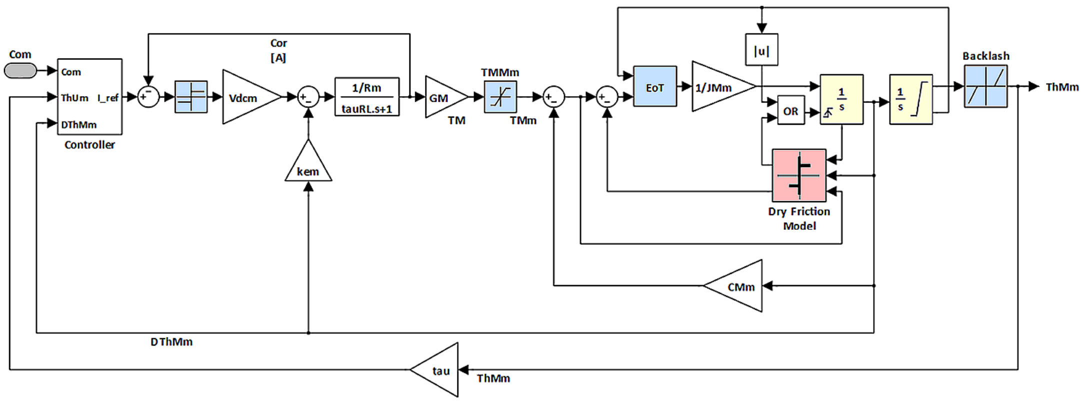

Fig. 3. EMA monitoring model (MM) block diagram (Berri et al., 2016).

5.1 *Genetic Algorithm*

Genetic Algorithm (GA), inspired by natural selection, was initially proposed by Holland in 1973 and is the most commonly employed bio-inspired optimization. At each iteration, the algorithm evaluates a population of solutions of the objective function. Each next population is made by the best individuals of the previous one (selection), their recombination (reproduction) and some new random solution (mutation). A proper balance between selection and mutation allows to trade-off the algorithm between search space exploration and convergence rate.

5.2 *Differential Evolution*

Differential Evolution (DE) was proposed by Storn and Price in 1995. It is another population-based, derivative-free algorithm, similar to GAs and applicable to continuous domains. The main difference with respect to GA lies in the recombination function: while GA leverages a random exchange of bits between the binary representations of two solutions, DE employs an arithmetical combination of the two points, e.g. a weighted average. This usually results in a better convergence rate than a traditional GA, but requires the search space to be continuous, and the performance may be significantly degraded when dealing with a noisy objective function.

5.3 *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) is a swarm-based search algorithm proposed by Kennedy and Eberhart in 1995. It evaluates the objective function in a collection of points, referred to as particles, that have an associated velocity in the search space. At each iteration, the position of the particles is updated by integrating their velocity, while velocity changes depending on the best-known solution, the values of neighbour particles, and the best solution found by that particular particle.

5.4 *Grey Wolf Optimization*

Grey Wolf Optimization (GWO) is an algorithm proposed by Mirjalili et al. in 2014 and inspired by the hunting technique of grey wolves. At first, a population of solutions is initialized and ranked hierarchically: the three best solutions are assigned as alpha, beta and delta, all the rest are classified as omegas, similarly to the hierarchy of a grey wolves herd. At each iteration, the omegas move toward alpha, beta and delta; then the hierarchy is updated. The process repeats until a stopping criterion is satisfied.

5.5 *Dragonfly Algorithm*

Dragonfly Algorithm (DA), as show by Mirjalili in 2016, simulates the swarming behaviour of dragonflies. The algorithm is similar to PSO, but employs a different set of rules to update the velocity of each solution in the search domain. At each iteration, the velocity of each solution is corrected in order to fulfil multiple requirements: (1) maintain separation between the solutions, (2) align the velocity with neighbour solutions, (3) reduce distances from neighbour solutions, (4) approach food, i.e. the best-known position, and (5) get away from the enemy, i.e. the worst known position.

5.6 *Whale Optimization Algorithm*

Introduced in 2016 by Mirjalili and Lewis, the Whale Optimization Algorithm (WOA) is a swarm-based algorithm that mimics the behaviour of humpback whales. The algorithm simulates the hunting strategy of these animals, that can be schematically summarized into three phases: (1) encircling the prey, i.e. evaluating the fitness function near the best-known location; (2) bubble-net attacking, i.e. progressively reducing the search radius around the best location; (3) search for prey, i.e. the exploration phase in which the objective function is evaluated at a progressively increasing distance from known solutions.

6. Fitness Function

To detect the occurrence (and evolution) of progressive faults as accurately as possible, the six optimization algorithms require an adequate fitness function: it expresses how the monitor model approximates in a satisfying way the high fidelity one. The definition of the fitness function in an optimization problem is a critical problem because most of the characteristics and behaviors of the optimization algorithms are based on this. As proposed by Dalla Vedova et al. (2019b), the fitness function is a cumulative error in terms of equivalent single-phase current between Monitor (MM) and Reference (RM) models. Operatively speaking, the main goal of the optimization process is to identify a suitable eight elements normalized vector ***k***, in which these coefficients are related to the actual magnitudes of the fault parameters already introduced in chapter 4.

Every element is normalized between 0 and 1 (as shown by Padmaraja, 2003):

$$\boldsymbol{k} = [k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8]\#(1)$$

$k_1 \in [0,1]$ represents the normalized friction fault: as shown by Borello and Dalla Vedova (2012). $k_1 = 0$ describes nominal conditions, $k_1 = 1$ introduces friction fault equal to three times the corresponding nominal conditions.

$k_2 \in [0,1]$ introduces the normalized backlash fault: according to Dalla Vedova et al. (2018) k_2=0 means nominal backlash, equal to 0.005 rad, k_2=1 represents a backlash equal to 100 times nominal conditions.

k_3, k_4, $k_5 \in [0,1]$, according to Berri et al. (2017), are the short circuit indexes for phase A, phase B and phase C, respectively. $k_i = 0$ means that the *i-th* phase is working correctly, $k_i = 1$ means that the *i-th* phase is in a complete short-circuit condition.

k_6, $k_7 \in [0,1]$ represents the normalized eccentricity fault in terms of amplitude and phase. In particular, k_6 is the rotor eccentricity ratio $\zeta = x_0/g_0$: if set to 0, it means that there is no static eccentricity; if it is equal to 1 the rotor touches the stator, decreasing the air gap to zero. k_7 describes the phase of the rotor eccentricity: $k_7 = 0$ implies a $\phi = -\pi$, $k_7 = 1$ is the same of an angle $\phi = \pi$. This last parameter, during the evaluation of the error, is suitably treated, because if the eccentricity is null, the eccentricity phase can assume all values between $-\pi$ and π.

$k_8 \in [0,1]$ gives a proportional gain drift. Nominal conditions are equal to $k_8 = 0.5$; $k_8 = 0$ means that the gain is decreased to 50%, k_8 =1 means that it is increased to 150%.

Hence, in nominal conditions the fault vector is $\boldsymbol{k} = [0,0,0,0,0,0,0.5,0.5]$.

The fitness function output is evaluated with a Total Least Squares Error (Markovsky and Van Huffel, 2007). This is preferred over a more classical Least Squares method to avoid the overestimation of the fitness function when the two EMA models gather a small phase lag when a current signal abruptly changes. As shown by Berri et al. (2018), the objective function evaluates the signal error ***err***, calculated between the reference enveloped current I_{3equiv} and the obtained monitor current I_m:

$$err = \sum_t \frac{(I_{3equiv}(t_0) - I_m(t_0))^2}{\sqrt{\frac{dI_{3equiv}(t_0)}{dt}^2 + 1}}\#(2)$$

7. Results Analysis

This chapter focuses on the evaluation of the obtained results; the six optimization algorithms have been tested to assess their performance. Simulations have been run in Matlab-Simulink environment (R2019a); calculations are managed by an Intel®Core™ i5-7200U with CPU @2.5 GHz, 8 GB of RAM, Window 10 Pro 64 bit.

Once the algorithms have been chosen, to compare them objectively, they must all be simulated with the same operating conditions. Then, the following settings have been selected:

Population: number of individuals of each generation. Increasing this parameter allows you to get better accuracy and reliable data, but raises the calculation time. In this work, it is set to 50.

Iterations: a higher number of iterations gives a better quality of the solutions but also a longer calculation time. In this work, it is set to 200.

Parallelization: to reduce the computational time, the algorithm splits the computational efforts over different cores (i.e. the algorithm can call multiple individuals at the same time).

7.1 *Optimization Percentage Error*

As anticipated in chapter 5, the reference current is generated by injecting the fault vector ***k*** into the high-fidelity model. Subsequently, using the monitoring model, the algorithms estimate the faults by calculating the trend of the equivalent current and suitably modifying the fault coefficients to approximate as better as possible the real reference current. In each optimization, the percentage error is computed as:

$$Err_{\%} = 100 \cdot \sqrt{\sum_{i=1}^{6} \left(k_i - \widehat{k_i}\right)^2 + \widehat{k_6} \cdot \left(k_7 - \widehat{k_7}\right)^2 + \left(k_8 - \widehat{k_8}\right)^2}$$

where $\widehat{\boldsymbol{k}} = [\widehat{k_1}, \widehat{k_2}, \widehat{k_3}, \widehat{k_4}, \widehat{k_5}, \widehat{k_6}, \widehat{k_7}, \widehat{k_8}]$ are the values of the reference model's fault. For every fault introduced into the RM, two different objective functions are investigated: the low fault detection (with $\widehat{k_i} \leq 0.25$)) and high fault detection ($\widehat{k_i} \geq 0.7$), to evaluate the effectiveness of the considered algorithms in different operating conditions.

7.2 *Performance Coefficient*

To effectively compare the overall performance of algorithms applied to the same fault condition, taking into account both their accuracy and the calculation time used by each method, authors introduced a new parameter named Performance Coefficient (PC), Dalla Vedova et al. (2019b).

Operatively speaking, it is the percentage value, calculated according to Eq. 3, indicating the effectiveness of a given algorithm, expressed in terms of calculation time and percentage error, for each type of fault considered:

$$PC_i(\%) = 100 \cdot \left(1 - \frac{t_i \cdot err_i(\%)}{\sum_{i=1}^{4} t_i \cdot err_i(\%)}\right) \#(3)$$

where PC_i is the said Performance Coefficient of the *i-th* optimization algorithm, t_i is its average computational time and $err_i(\%)$ is its average percentual error (note that PC_i is dimensionless and expressed in terms of percentage).

The denominator has been introduced to have as output a non-dimensional value; subtracting the resulting value to 1 and the next multiplication for 100 transforms it in a percentage value. In this way, choosing the highest PC means choosing the best algorithm for that problem. In table below the PC has been calculated for all the faults studied in this work.

7.2 *Single Fault Analysis*

In this section, a comparative study of the different algorithms has been developed for each of the considered failures. The values that have been used for this comparison are time and error. The time refers to the time that each simulation has lasted, while the error, as already seen in the previous section, takes into account how close are the fault parameters obtained to those that are introduced in the reference model.

Table 1 shows the ten failure modes with which the FDI algorithms have been tested:

Table 1. Fault vectors used to test FDI methods.

k	F	B	N_a	N_b	N_c	Z	phi	Gain
# 1	25%	0%	0%	0%	0%	0%	50%	50%
# 2	75%	0%	0%	0%	0%	0%	50%	50%
# 3	0%	10%	0%	0%	0%	0%	50%	50%
# 4	0%	75%	0%	0%	0%	0%	50%	50%
# 5	0%	0%	80%	0%	0%	0%	50%	50%
# 6	0%	0%	30%	0%	0%	0%	50%	50%
# 7	0%	0%	0%	0%	0%	25%	50%	50%
# 8	0%	0%	0%	0%	0%	75%	50%	50%
# 9	0%	0%	0%	0%	0%	0%	50%	25%
# 10	0%	0%	0%	0%	0%	0%	50%	75%

where **F** represents the friction fault, **B** is the backlash fault, $\mathbf{N_a}$, $\mathbf{N_b}$, $\mathbf{N_c}$ represent the short circuit fault of each phase, **Z** is the eccentricity fault, **phi** is the angular fault, and finally, **Gain** represents the gain fault. It should be noted that for each of these ten cases, ten different optimizations have been made for each of the algorithms considered. Therefore, the results shown in the following histograms represent the arithmetic mean of the ten optimizations carried out for each algorithm and failure mode. Figures 4 and 5 highlight that, at least in terms of percentage error, some metaheuristic algorithms (i.e. PSO, DE, and GWO) provide performances better than the classic GA. However, as shown in Figure 4, the PSO offers the best accuracy in terms of average percentage error, particularly for the detection of low failures. This property is particularly clear in backlash fault and gain fault, where the percentage error is less than 0.5%; on the other hand, for a high rotor eccentricity Z, the GWO obtained its worst performances, with an average total percentage error higher than 4%.

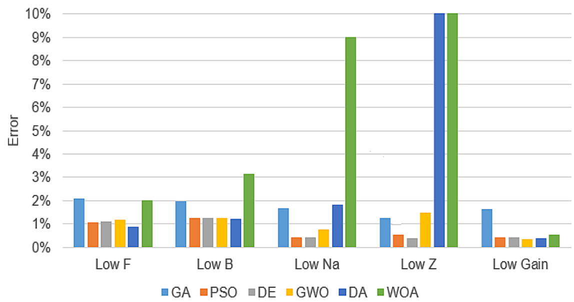

Fig. 4. Average error obtained running the considered six optimization algorithms in case of Low Fault conditions.

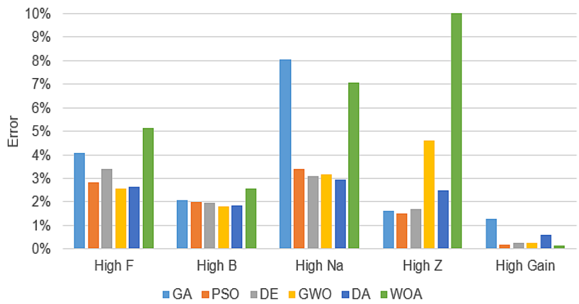

Fig. 5. Average error obtained running the considered six optimization algorithms in case of High Fault conditions.

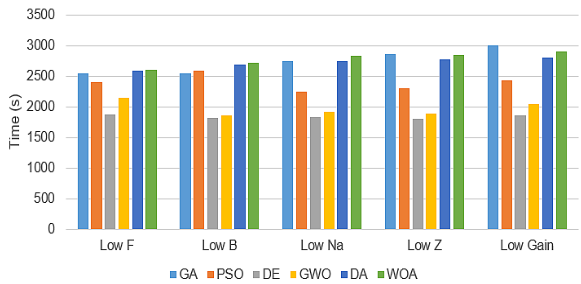

Fig. 6. Average computational time obtained running the six considered optimization algorithms in Low Fault conditions.

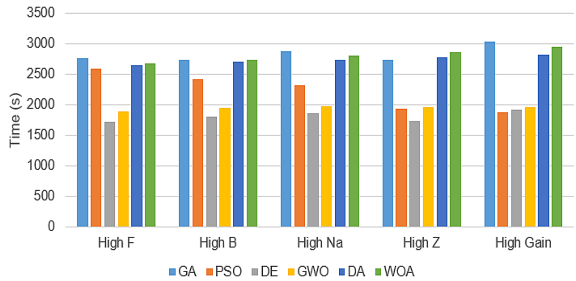

Fig. 7. Average computational time obtained running the six considered optimization algorithms in High Fault conditions.

This unsatisfying behaviour can be justified considering the search strategy of GWO algorithm: it encourages the exploration of the search agents instead exploitation and so the accuracy is badly affected. It should be noted that the DA algorithm also has very low and competitive errors, but it behaves very badly in the high eccentricity fault and this penalizes it. However, DA provides the better results in almost 30% of cases considered. Something similar happens to WOA, which offers the best performance in the event of a high gain error (but it is also satisfactory for low gain errors). However, given that WOA provides the worst performance in case of eccentricity failure, it is not an acceptable option. Referring to Figs. 6 and 7, we can state that, as regards the performance of algorithms in terms of average computational time, GWO and DE are the fastest ones, especially to carry out the optimization in case of single low fault conditions. It can be explained taking into account that these two codes have been implemented in Matlab by the authors and, therefore, allow to better the chosen and setting of stop criteria. The correct calibration of these stop criteria allows better performances (greater precision and convergence speed) for these two optimization algorithms. Concerning the average calculation time, it was also observed that DA and WOA are not able to provide superior performance to the other algorithms but, on the contrary, are comparable with GA. Table 2 summarizes the overall performance provided by the six optimization algorithms, quantifying them in terms of performance coefficient PC_i. (described in Eq. 3). Concerning the overall values, it is possible to affirm that, as regards the single faults considered in this work, PSO, DE and GWO provide similar performances and, therefore, are generally the most effective optimization algorithms.

Table 2. Performance Coefficient PC_i

	GA	PSO	DE	GWO	DA	WOA
Low F	74%	87%	90%	87%	89%	74%
High F	77%	85%	88%	90%	86%	73%
Low B	80%	87%	91%	90%	87%	66%
High B	81%	84%	88%	88%	83%	76%
Low Na	88%	97%	98%	96%	87%	33%
High Na	67%	89%	92%	91%	89%	72%
Low Z	97%	93%	99%	98%	69%	43%
High Z	97%	98%	98%	94%	96%	17%
Low Gain	51%	90%	92%	93%	90%	85%
High Gain	47%	95%	93%	93%	78%	94%
Average	76%	91%	93%	92%	85%	63%

8. Conclusions and Future Works

Referring to the proposed case study (an EMA for secondary flight controls), it is possible to state that PSO, DE, and GWO are the best optimization algorithms to perform the FDI of the (single) faults considered, reaching the highest PC value (all over 90 %). On contrary, DA and WOA algorithms have not achieved satisfactory results. However, the authors believe that the evaluation of new algorithms, despite the negative results gained, is still useful. Indeed, it must be noted that, as shown by Wolpert and Macready (1997), the suitable algorithm for a given problem must be chosen case by case. However, to better evaluate the effectiveness of these algorithms, it would be advisable to evaluate them even in case of multiple failures. The future research on this topic will be focused on the comparison of other types of optimization algorithms such as the swarm intelligence ones. In addition, several multiple faults should be evaluated, in order to assess a wider applicability of these methods

References

Byington, C., W. Watson, D. Edwards, and P. Stoelting (2004). A Model-Based Approach to Prognostics and Health Management for Flight Control Actuators. *IEEE Aerospace Conf. Proceedings*.

Berri, P.C., M.D.L Dalla Vedova, and P. Maggiore (2016). A smart electromechanical actuator monitor for new model-based prognostic algorithms. *International Journal of Mechanics and Control 17(2)*, 59-66.

Berri, P.C., M.D.L Dalla Vedova, and P. Maggiore (2017). On-board electromechanical servo mechanisms affected by progressive faults: proposal of a smart GA model-based prognostic approach. *Safety and Reliability - Theory and Applications - Proc. of ESREL 2017*, 839-845.

Berri, P.C., M.D.L Dalla Vedova, and P. Maggiore (2018). Enhanced hybrid prognostic approach applied to aircraft on-board electromechanical actuators affected by progressive faults. *Safety and Reliability - Safe Societies in a Changing World - Proc. of ESREL 2018*, 1077-1084.

Binitha, S., and S. Siva Sathya (2012). A Survey of Bio inspired Optimization Algorithms. *International Journal of Soft Computing and Engineering (IJSCE) 2*, 137-151.

Borello, L., G. Villero, and M.D.L. Dalla Vedova (2009). New asymmetry monitoring techniques: Effects on attitude control. *Aerospace Science and Technology 13(8)*, 475-487.

Borello L. and M.D.L. Dalla Vedova (2012). A dry friction model and robust computational algorithm for reversible or irreversible motion transmission. *International Journal of Mechanics and Control 13(2)*, 37-48.

Borello, L., G. Villero, and M.D.L. Dalla Vedova (2014). Flap failure and aircraft controllability: developments in asymmetry monitoring techniques. *Journal of Mechanical Science and Technology (JMST) 28(11)*, 4593-4603.

Chesley, J.C. (2011). Handbook of Reliability Prediction Procedures for Mechanical Equipment. USA Navy, Naval Surface Warfare Center, NSWC Carderock Division.

Dalla Vedova, M.D.L., P. Maggiore, L. Pace, and A. Desando (2015). Evaluation of the correlation coefficient as a prognostic indicator for electromechanical servomechanism failures. *International Journal of Prognostics and Health Management 6(1)*.

Dalla Vedova, M.D.L., P.C. Berri, P. Maggiore (2018). Novel Metaheuristic Bio-inspired Algorithms for Prognostics of Onboard Electromechanical Actuators. *International Journal of Mechanics and Control 19(2)*, 95-101.

Dalla Vedova, M.D.L., and P.C. Berri (2019a). Optimization techniques for prognostics of on-board electromechanical servomechanisms affected by progressive faults. *International Review of Aerospace Engineering 12(4)*, 160-170.

Dalla Vedova, M.D.L., P.C. Berri, and S. Re (2019b). A Comparison of Bio-Inspired Meta-Heuristic Algorithms for Aircraft Actuator Prognostics. *Proc. of the 29th European Safety and Reliability Conference (ESREL 2019)*, 1064-1071.

Dalla Vedova, M.D.L., A. Germanà, P.C. Berri, and P. Maggiore (2019c). Model-Based Fault Detection and Identification for Prognostics of Electromechanical Actuators Using Genetic Algorithms. *Aerospace 6(9)*, 01-15.

Ginart, A., D. Brown, P. Kalgren, and M. Roemer (2007). On-line Ringing Characterization as a PHM Technique for Power Drives and Electrical Machinery. *Autotestcon*, 17-20.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing 2(2)*, 88-105.

Kennedy, J. and R. Eberhart (1995). Particle Swarm Optimization. *Proc. of the IEEE International Conference on Neural Networks 4*, 1942-1948.

Lones, M.A. (2014). Metaheuristics in Nature-Inspired Algorithms. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'14)*.

Markovsky I. and S. Van Huffel (2007). Overview of total least-squares methods. *Journal of Signal Processing 10*, 2283-2302.

Mirjalili, S., S.M. Mirjalili and A. Lewis (2014). Grey Wolf Optimizer. *Advances in Engineering 69*.

Mirjalili, S. (2015). Dragonfly Algorithm: a New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems. *Neural Computing and Applications 27(4)*, 1053-1073.

Mirjalili, S., and A. Lewis (2016). The Whale Optimization Algorithm. *Advances in Engineering Software 95*, 51-67.

Padmaraja, Y. (2003). Brushless DC (BLDC) motor fundamentals, Application Note AN885, Microchip Technology Inc.

Rosero, J.A., J.A. Ortega, E. Aldabas, and L. Romeral (2017). Moving Towards a More Electric Aircraft. *IEEE Aerospace and Electronic Systems Magazine 22(3*), 3-9.

Storn, R. and K. Price (1997). Differential Evolution - a simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization 11*.

Vachtsevanos, F., F. Lewis, M. Roemer, A. Hess, and B. Wu (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Wiley.

Wolpert, D.H. and W.G. Macready (1997). No Free Lunch Theorems for Optimization. *IEEE Trans. on Evolutionary Computation 1*, 67-82.