A Hybrid TLBO-Particle Filter Algorithm Applied to Remaining Useful Life Prediction in the Presence of Multiple Degradation Factors

Leonardo R. Rodrigues
Electronics Engineering Division
Aeronautics Institute of Technology
São José dos Campos, Brazil
leonardolrr2@fab.mil.br

Daniel B. P. Coelho
Performance Software Department
Embraer S.A.
Belo Horizonte, Brazil
daniel.coelho@embraer.com.br

João Paulo P. Gomes
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
jpaulo@lia.ufc.br

***Abstract*—One of the end goals of a Prognostic and Health Monitoring (PHM) algorithm is to provide accurate Remaining Useful Life (RUL) predictions for the monitored component or system. Most of the PHM algorithms found in the literature are based on the assumption that the degradation process is governed by only one degradation factor. However, some components and systems may be subject to multiple degradation factors. In this paper, we propose a hybrid algorithm that incorporates a Teaching-Learning Based Optimization (TLBO) step into a Particle Filter (PF) framework. PF is an algorithm that can handle multiple degradation factors. However, it has some drawbacks such as sample degeneracy and sample impoverishment. The hybrid TLBO-PF algorithm proposed in this paper improves the performance of the standard PF algorithm by reducing the effects of sample degeneracy and sample impoverishment. A case study is presented to evaluate the performance of the proposed algorithm for estimating the degradation factors and predicting the RUL of a Lithium-ion battery, which is affected by two degradation factors. The results show that the proposed algorithm presented a better performance for both the tasks (degradation factor estimation and RUL prediction) when compared with the standard Particle Filter algorithm.**

***Index Terms*—Prognostics, health monitoring, remaining useful life, particle filter, teaching-learning based optimization, Lithium-ion battery.**

I. Introduction

Due to the increasing competitiveness in many industry sectors, high availability levels are expected from equipment and systems. To meet the high availability requirements, developing a robust maintenance plan that reduces the number of unscheduled maintenance activities plays a crucial role. In this scenario, the use of Prognostics and Health Monitoring (PHM) techniques can be seen as a powerful tool to monitor the health status of critical components and prevent unexpected failure events from happening [1].

A PHM system uses operational data obtained from sensors that provide information about the degradation level of components and systems. These data are used to assess the degradation level of the monitored system. Many PHM approaches have been proposed in the literature to monitor the degradation level of a wide range of components such as batteries [2], servo valves [3], wind turbines [4], among others.

Most of the proposed PHM methods assume that the monitored component operates under the influence of only one degradation factor [5], [6]. However, modern systems are often composed of many complex interacting components and, in many cases, multiple degradation mechanisms are present [7], [8]. In these situations, the mathematical techniques used in PHM methods must be capable of dealing with multiple degradation factors.

The Particle Filter (PF) algorithm can deal with multiple degradation factors and has been used in many PHM solutions [9]. However, PF suffers from sample impoverishment and sample degeneracy. Sample impoverishment occurs when distributions with wide variances are used. Sample degeneracy occurs when the particles are too concentrated. These problems reduce the accuracy of the solutions obtained with a PF [10].

Hybrid versions of PF have been proposed to overcome sample impoverishment and sample degeneracy. In [11], the authors proposed a hybrid method that uses a heuristic Kalman Filter along with PF to identify the number of degradation factors and predict the RUL (Remaining Useful Life) of a monitored system. In [12], the authors integrated unscented Kalman Filter and wavelet transform into the PF framework to solve a GPS multipath mitigation problem. The use of nature-inspired metaheuristics has also been investigated. In [13], the authors presented a hybrid method using a PF and a PSO (Particle Swarm Optimization) algorithm. They obtained good results. However, parameter tuning in PSO can be a challenging task.

In this paper, we propose a hybrid algorithm aiming at reducing the effects of sample impoverishment and sample degeneracy. We use the standard PF framework, and incorporate a step based on the Teaching-Learning Based Optimization (TLBO) algorithm. TLBO is a population-based metaheuristic algorithm inspired by the teaching-learning process observed in a classroom [14]. The two main phases of TLBO are the Teacher Phase and the Student Phase, which are responsible for the intensification and the diversification capabilities of the algorithm, respectively. Also, TLBO does not require any specific parameter to be defined.

The main contribution of this paper is proposing a hybrid method to reduce the effects of sample impoverishment and sample degeneracy in a PF framework. The performance of the proposed algorithm is illustrated in a case study that consists in estimating the Remaining Useful Life (RUL) of a Lithium-ion battery, which has a degradation process affected by two degradation factors. The performance of the standard PF is used as a reference baseline. The results show that the proposed TLBO-PF algorithm provided better performance in terms of degradation parameters estimation error and RUL prediction accuracy when compared with the standard PF.

The remaining sections of this paper are organized as follows. Section II presents a brief overview of Particle Filters, the Teaching-Learning Based Optimization algorithm, and the basic concepts of prognostic and health monitoring techniques and RUL predictions. Section III introduces the proposed hybrid TLBO-PF algorithm. Section IV presents the results obtained in a case study used to illustrate the application of the proposed algorithm to estimate the RUL of a lithium-ion battery. Concluding remarks are presented in Section V.

II. Theoretical Background

A. Particle Filter

A Particle Filter (PF) is a sequential Monte Carlo method that uses a point mass representation of the probability density function [8], [15]. A Particle Filter uses a Bayesian inference approach, in which measurements are used to estimate the values of unknown parameters as a probability density function (PDF). Parameter predictions are updated in each step, when new measurements become available. The process of a Particle Filter algorithm is based on the state transition function $f(\cdot)$ and the measurement function $g(\cdot)$. These functions are presented in (1) and (2), respectively.

$$\begin{aligned} x_k &= f\left(x_{k-1}, \theta_k, \upsilon_k\right) && (1) \\ y_k &= g\left(x_k, \eta_k\right) && (2) \end{aligned}$$

where $f(\cdot)$ is the nonlinear state transition function, k is the time step index, x_k is the state model at the k-th step, θ_k is a vector of model parameters, υ_k is the i.i.d. (identical independent distributed) process noise, y_k is the measurement observed at the k-th step, $g(\cdot)$ is the measurement function, and η_k is the i.i.d. measurement noise. The standard deviation of η_k is denoted by σ.

In PF, the initial distribution of the states, denoted by $p(x_0)$, must be defined. A set of N particles is sampled from the initial distribution and propagated through the state transition function f. The distribution obtained from the particles at this stage is known as the prior distribution. Then, a new measurement y_k becomes available and is used to compute a likelihood value for each particle n, with $n = \{1, \ldots, N\}$, according to (3).

$$L\left(y_k | x_k^{(n)}\right) = \frac{1}{\sigma\sqrt{2\pi}} \cdot exp\left(-\frac{\left(y_k - x_k^{(n)}\right)^2}{2\sigma^2}\right) \qquad (3)$$

The likelihood value computed for each particle n represents its relevance in the model and is used as a weight, $w_k^{(n)}$, according to (4). The weights are then normalized according to (5).

$$w_k^{(n)} = L\left(y_k | x_k^{(n)}\right) \qquad (4)$$

$$w_k^{(n)} = \frac{w_k^{(n)}}{\sum_{n=1}^{N} w_k^{(n)}} \qquad (5)$$

In the sampling step of PF, the weight assigned to each particle n is used as the probability of obtaining a sample with an index n. A new set of N samples is drawn from the discrete distribution, with replacement. Particles with high associated weights are likely to be drawn multiple times while particles with low associated weights are likely to be eliminated during the resampling step. This new set of particles is known as the posterior distribution and replaces the prior distribution. At the end of the resampling step, all particles receive the same weight, according to (6). The larger the number of particles N, the better the state estimation. However, more computational power is required. Fig. 1 illustrates the overall process of the PF algorithm.

$$w_k^{(n)} = \frac{1}{N} \qquad (6)$$

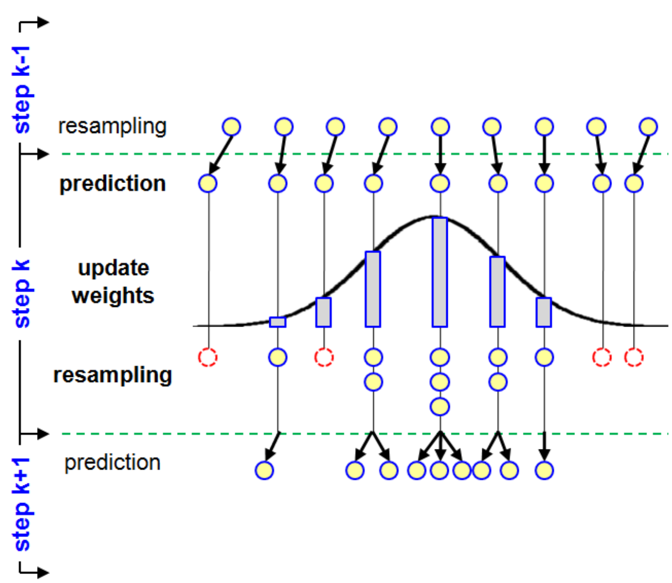

Fig. 1. PF algorithm process

B. Teaching-Learning Based Optimization

The Teaching-Learning Based Optimization (TLBO) algorithm is a population-based metaheuristic algorithm inspired by the teaching-learning process observed in a classroom [14]. This algorithm simulates the influence of a teacher on the output of a group of students in a class. The algorithm has two main phases: the Teacher Phase and the Student Phase [16]. During the Teacher Phase, students learn from the teacher, while in the Student Phase students learn through interactions among themselves.

Consider a group of N students. Each student X has an associated solution that corresponds to a candidate solution for the optimization problem. The quality of each solution is quantified by a fitness value $f(X)$ that is computed by evaluating the solution X using the objective function.

The student with the best solution in each iteration is called the Teacher. Fig. 2 shows the flowchart for implementing the TLBO algorithm [14]. The Teacher Phase and the Student Phase are described in the next sections.

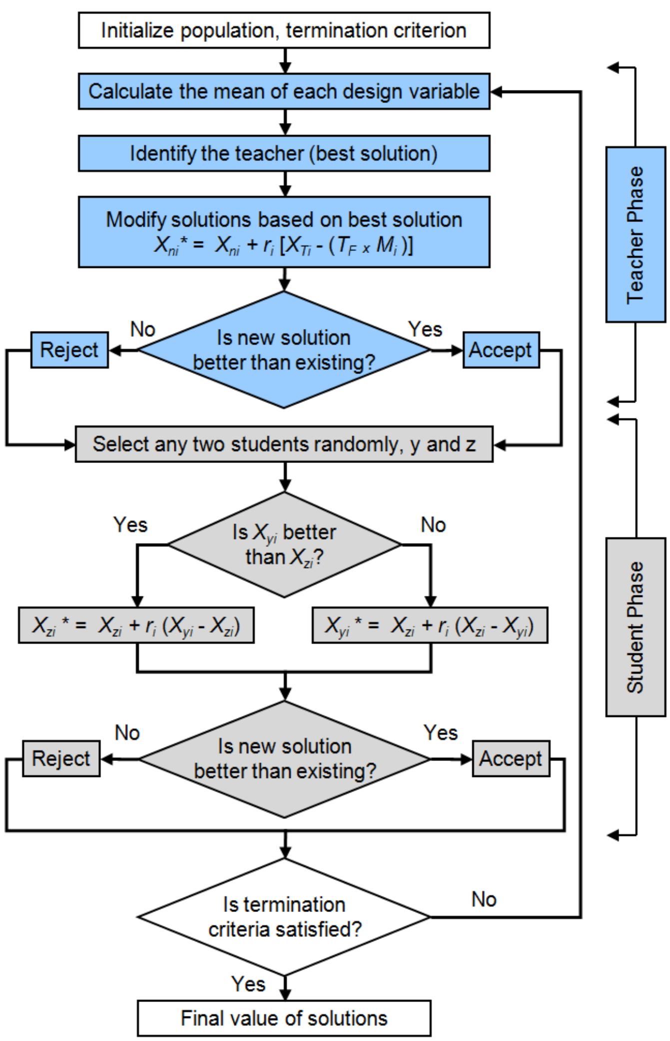

Fig. 2. TLBO flowchart

1) ***Teacher Phase:*** During the Teacher Phase, the algorithm simulates the learning of the students from the teacher (best solution). During this phase, the teacher makes an effort to increase the mean result of the class. Let M_i be the mean solution of all the students and T_i be the teacher in the i-th iteration. The teacher T_i will try to move M_i to its level. Knowledge is obtained based on the quality of the teacher and the quality of students. The difference D_i between the solution of the teacher, denoted by X_{Ti}, and the mean solution of the students, M_i, is expressed according to (7).

$$D_i = r_i(X_{Ti} - T_F \cdot M_i) \tag{7}$$

where r_i is a random number chosen from a standard uniform distribution, and T_F is the teaching factor for iteration i, which is randomly set to either 1 or 2 according to (8).

$$T_F = round(1 + rand(0, 1)) \tag{8}$$

Based on the difference D_i, the current solution associated with each student n in iteration i, denoted by X_{ni}, with $n \in \{1, 2, \ldots, N\}$, is updated during the teacher phase according to (9).

$$X^\star_{ni} = X_{ni} + D_i \tag{9}$$

where $X^\star_{ni}$ is the updated value of X_{ni}.

If $f(X^\star_{ni})$ is better than $f(X_{ni})$, then $X^\star_{ni}$ is accepted and replaces X_{ni} for the next iteration. Otherwise, $X^\star_{ni}$ is discarded.

2) ***Student Phase:*** During the Student Phase, TLBO simulates the learning of the students through interactions among themselves. During this phase, students gain knowledge by discussing with other students who have more knowledge [16].

Consider a pair of students y and z. Let X_{yi} and X_{zi} be the solutions of students y and z in iteration i, respectively. If $f(X_{yi})$ is better than $f(X_{zi})$, the solution of student z is updated according to (10). If $f(X^\star_{zi})$ is better than $f(X_{zi})$, $X^\star_{zi}$ is accepted and replaces X_{zi} for the next iteration. Otherwise, $X^\star_{zi}$ is discarded. Similarly, if $f(X_{zi})$ is better than $f(X_{yi})$, the solution of student y is updated according to (11). If $f(X^\star_{yi})$ is better than $f(X_{yi})$, $X^\star_{yi}$ is accepted and replaces X_{yi} for the next iteration. Otherwise, $X^\star_{yi}$ is discarded.

$$X^\star_{zi} = X_{zi} + r_i(X_{yi} - X_{zi}) \tag{10}$$
$$X^\star_{yi} = X_{yi} + r_i(X_{zi} - X_{yi}) \tag{11}$$

At the end of each iteration, the stop criteria are checked. Different stop criteria may be adopted. Some commonly used stop criteria are the maximum number of iterations, the maximum number of successive iterations without any improvement, the maximum simulation time, and the maximum number of objective function evaluations.

C. PHM Systems and RUL Prediction

A Prognostics and Health Monitoring (PHM) system evaluates operational data from components and systems to quantify their degradation level and to predict when a failure event is likely to occur. One way to estimate the degradation level of a monitored component is by comparing its current performance with the nominal performance. The RUL (Remaining Useful Life) is estimated based on the expected evolution of the degradation level and a failure threshold value, which is assumed to be known.

RUL predictions are often represented in the form of a probability distribution. Different parametric probability distributions such as Gaussian and Weibull can be used [17], [18]. Non-parametric distributions can also be adopted [19]. Fig. 3 illustrates the degradation evolution and the RUL prediction process [20]. Each symbol "+" in Fig. 3 represents the degradation level measured (or computed) at a specific time. During its operation, the component degradation level increases due to the evolution of degradation factors such as wear, slackness, corrosion, vibration, overheating, leakage, etc. The time interval in which the degradation level reaches the failure threshold defines the RUL.

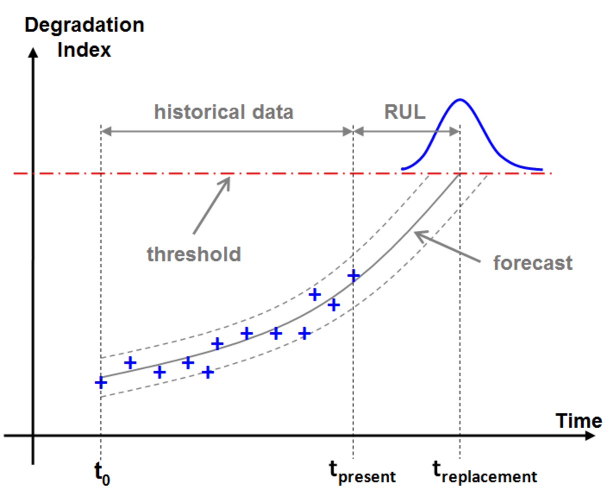

Fig. 3. Degradation evolution and RUL prediction process

Most PHM models proposed in the literature consider the existence of a single degradation factor in the component or system under consideration. However, due to the increasing complexity of modern systems, there are many cases in which multiple degradation factors are present. In these cases, PHM models that consider only one failure mechanism will not provide accurate RUL predictions [11]. In the past few years, some PHM models that consider multiple degradation mechanisms have been proposed in the literature [21], [22], [1].

The model proposed in this paper is a hybrid algorithm that uses a PF framework with a TLBO step. In PF, a d-dimensional state vector x can be used to estimate multiple degradation factors. Similarly, a d-dimensional candidate solution can be associated with each student in TLBO. Thus, the proposed algorithm can handle problems in which multiple degradation factors are present.

III. Proposed Hybrid TLBO-PF Algorithm

In this section, we present the proposed hybrid TLBO-PF algorithm. As mentioned earlier, our goal is to use the TLBO algorithm after the particle weight update step of PF aiming at reducing the effects of sample degeneracy and sample impoverishment, which are two well-known drawbacks of the PF algorithm [23], [24].

In the original PF algorithm, sample impoverishment may occur when the initial distributions with wide variance are adopted. It may occur, for example, when there is no information on the degradation mechanism. Due to the wide variance of initial distributions, the weights assigned to both important and unimportant particles are similar, and important particles can be removed while unimportant particles can be drawn during the resampling step. Sample degeneracy may occur when the particles are too concentrated. In this situation, the number of particles with high weights can be small. Consequently, the number of different particles selected to form the posterior distribution will be small, leading to a posterior distribution with small diversity. In both situations, the Monte Carlo approximations of the posterior distributions obtained with the PF tend to be inaccurate.

In the proposed TLBO-PF algorithm, we incorporate one iteration of the TLBO algorithm before the resampling step in each iteration of PF. A good balance between the diversification and the intensification capabilities is a desired characteristic for metaheuristic algorithms [25], [26]. In TLBO, the Teacher Phase and the Student Phase are responsible for the intensification and the diversification capability of the algorithm, respectively. Fig. 4 shows a flowchart of the proposed TLBO-PF algorithm.

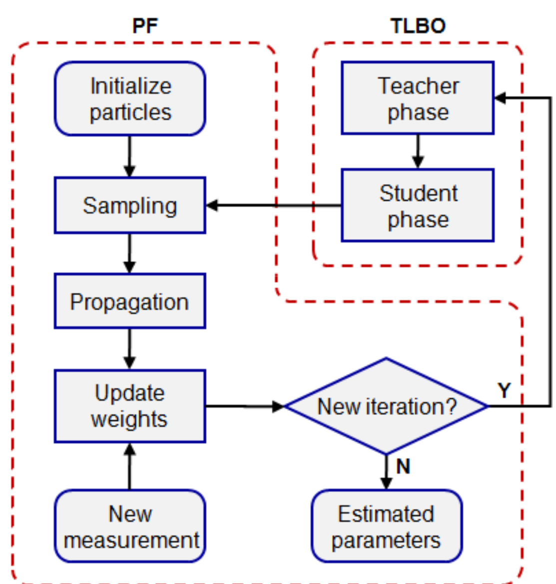

Fig. 4. Flowchart of proposed hybrid TLBO-PF algorithm

During the TLBO step of TLBO-PF, both the Teacher Phase and the Student Phase are implemented. Each particle of PF is considered as a student in TLBO-PF. Also, the weight w_n assigned to each particle n is considered as its fitness value. In the Teacher Phase, particles that are far away from the true value move towards the Teacher (the particle with the highest weight) reducing the sample impoverishment problem. In the Student Phase, if particles are too concentrated, they can spread and reduce the sample degeneracy problem.

IV. Numerical Experiments

This section presents a case study conducted to evaluate the performance of the proposed model. We consider a case study that consists in predicting the RUL of a Lithium-ion battery. It has been proven that the internal resistance of a battery is an accurate parameter to predict its degradation level and, consequently, its RUL [27], [28], [29]. Also, the Lithium-ion battery internal resistance can be divided into two different parts: the electrolyte resistance and the charge transfer resistance. Each part may present a different degradation mechanism [2], [11].

In this case study, the electrolyte resistance and the charge transfer resistance are denoted by R_E and R_C, respectively. Also, we assume that only measurements of the total resistance $R = R_E + R_C$ are available. The initial value of R_E and R_C are known. However, the degradation rates associated with R_E and R_C, denoted by α_E and α_C, respectively, are unknown. An exponential degradation model is assumed for both R_E and R_C according to (12) and (13), respectively.

$$R_E(k) = R_{E0} \cdot \exp(\alpha_E k) \tag{12}$$

$$R_C(k) = R_{C0} \cdot \exp(\alpha_C k) \tag{13}$$

Table I shows the numerical values used in the simulations. Resistance R_F defines the failure threshold, i.e. the battery fails whenever the total internal resistance $R = R_E + R_C$ reaches R_F.

TABLE I
Battery simulation data

parameter	value	unit
R_{E0}	0.10	Ω
R_{C0}	0.03	Ω
R_F	1.0	Ω
α_E	0.012	N/A
α_C	0.026	N/A

Fig. 5 shows the evolution of the total internal resistance over the operation cycles. A Gaussian measurement noise v with zero mean and standard deviation $\sigma = 0.06$ is considered. The battery fails in $k = 116$, when total internal resistance R is higher than R_F.

To predict the battery failure instant, two algorithms are used: the standard Particle Filter (PF) and the proposed hybrid TLBO Particle Filter algorithm (TLBO-PF). Table II presents the parameters used in the simulations.

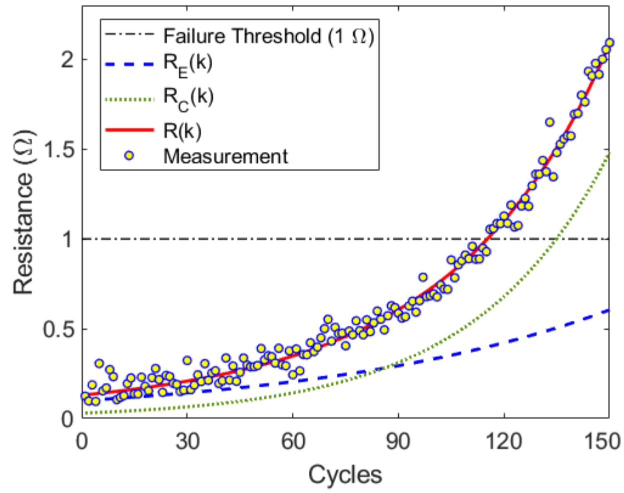

Fig. 5. Evolution of total internal resistance

TABLE II
Parameter values

parameter	description	value
N	number of particles	$3,000$
K	maximum number of iterations	150
pop	population size (TLBO)	$3,000$
$\alpha_E(0)$	initial distribution of α_E	$\mathcal{U}[0.002; 0.040]$
$\alpha_C(0)$	initial distribution of α_C	$\mathcal{U}[0.005; 0.090]$
$\sigma(0)$	initial distribution of σ	$\mathcal{U}[0.02; 0.10]$

A. Simulation Results

This section presents the results obtained during the simulations. A Monte Carlo simulation approach with 200 repetitions for each algorithm was adopted. Our goal is to compare the performances of PF and TLBO-PF in terms of degradation rate estimation and RUL prediction accuracy. All the experiments reported in this paper were carried out on a personal computer with Intel® Core™ i3, 1.9 GHz processor and 4GB RAM. The algorithm was coded in Matlab®.

*1) **Degradation Rate Prediction:*** Figs. 6 and 7 illustrate the degradation rate predictions obtained in one Monte Carlo repetition with PF and TLBO-PF, respectively. It can be seen from Figs. 6 and 7 that the proposed algorithm provided a better prediction error. To conduct a quantitative comparison, the root mean square error (RMSE) is computed for each algorithm. RMSE is computed according to (14). The RMSE values computed for the PF and TLBO-PF algorithms were 0.0114 and 0.0069, respectively. It shows that the proposed algorithm provided better performance in terms of prediction accuracy. We also computed the simulation time for each algorithm. The average simulation time, in seconds, for each repetition of PF and TLBO-PF were 1.837 and 2.299, respectively. It shows that the TLBO step incorporated into the PF framework reduces the estimation error but increases the required computational power by about 25%.

$$RMSE = \sqrt{\frac{1}{MC}\sum_{j=1}^{MC}\left[\left(\alpha_E^{(j)} - \alpha_E\right)^2 + \left(\alpha_C^{(j)} - \alpha_C\right)^2\right]} \tag{14}$$

where $\alpha_E^{(j)}$ and $\alpha_C^{(j)}$ are the degradation rate predictions obtained in the j-th Monte Carlo repetition, and MC is the number of Monte Carlo repetitions.

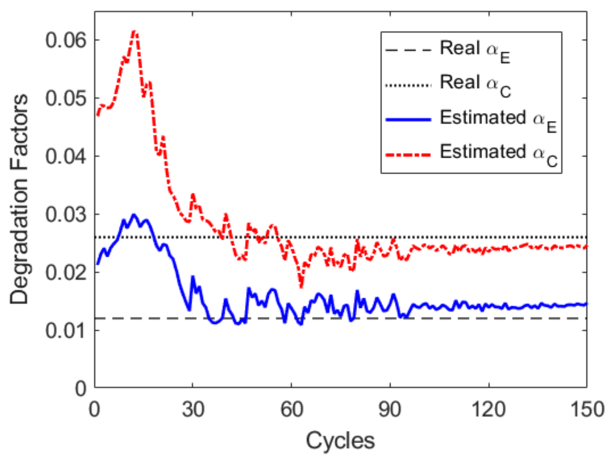

Fig. 6. Degradation rate predictions using the Particle Filter algorithm

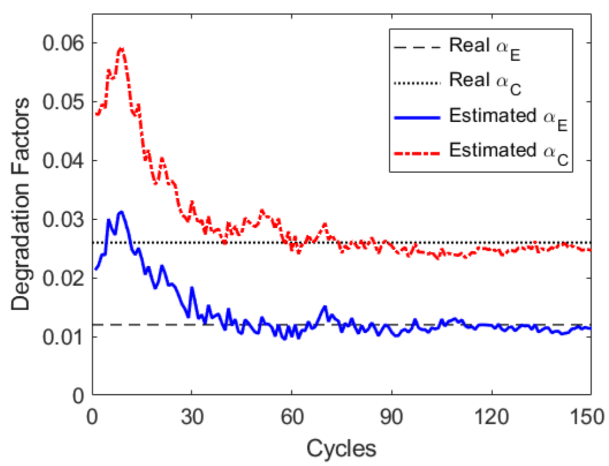

Fig. 7. Degradation rate predictions using the proposed hybrid TLBO-PF algorithm

*2) **RUL Prediction**:* The ultimate goal of a prognostic algorithm is to predict when a failure event of the system under consideration is likely to occur. In [30], the authors proposed some performance metrics to quantify the performance of prognostics algorithms. In this paper, we compare the performance of the algorithms in terms of RUL prediction accuracy using a modified Prognostics Horizon (MPH) indicator. The Modified Prognostic Horizon (MPH) used in this paper is computed according to (15).

$$MPH = t_{EoL} - t_{EB} \tag{15}$$

where t_{EoL} is the time index in which a failure event of the system under consideration occurs and represents its End of Life (EoL), and t_{EB} is the time index in which the RUL prediction meets the performance requirements and continues to meet them until the End of Life of the system. Higher values of MPH are associated with better prognostic performances.

In this paper, the performance requirement for RUL prediction purposes is defined as the maximum admissible error bound around the true RUL. The evolution of RUL predictions for the same system obtained from two different algorithms are presented in Fig. 8 to illustrate the MPH concept. The End of Life is the same for the two algorithms since there is only one system. The black continuous line represents the true RUL and the gray area represents the admissible Error Bound (EB). For algorithm 1 (yellow squares), the first RUL prediction inside the gray area occurs when $k = 3$. All RUL predictions obtained from algorithm 1 for $k \geq 3$ are inside the gray area. So, $t_{EB} = 3$ for algorithm 1. Considering algorithm 2 (red circles), although the first RUL prediction inside the gray area occurs when $k = 2$, for $k = 3$ the RUL prediction is outside the gray area. When $k = 4$, the prediction is inside the gray area and all RUL predictions for $k \geq 4$ are also inside the gray area. So, $t_{EB} = 4$ for algorithm 2. Based on the MPH indicators, in this example, algorithm 1 presented a better performance.

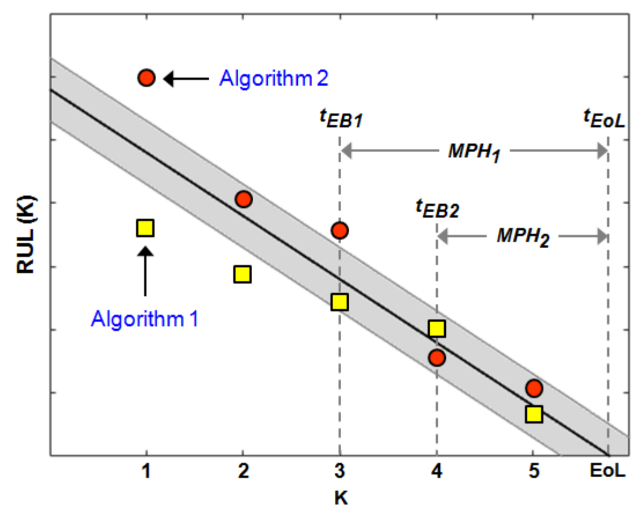

Fig. 8. Modified prognostics horizon for two different algorithms.

Figs. 9 and 10 show the evolution of RUL predictions in one Monte Carlo repetition for PF and TLBO-PF, respectively. The End of Life for this case study occurs for $k = 116$. A maximum admissible error bound of 10 is adopted. The values of t_{EB} for PF and TLBO-PF in Figs. 9 and 10 are 95 and 72, respectively. The MPH values for PF and TLBO-PF for the repetitions presented in Figs. 9 and 10 are 21 and 44, respectively. Considered all Monte Carlo repetitions, the MPH computed for the standard PF was 15.922 ± 6.502, while the MPH computed for TLBO-PF was 25.465 ± 5.398. It can be seen that the proposed TLBO-PF presented a better performance in comparison with PF in terms of RUL prediction accuracy.

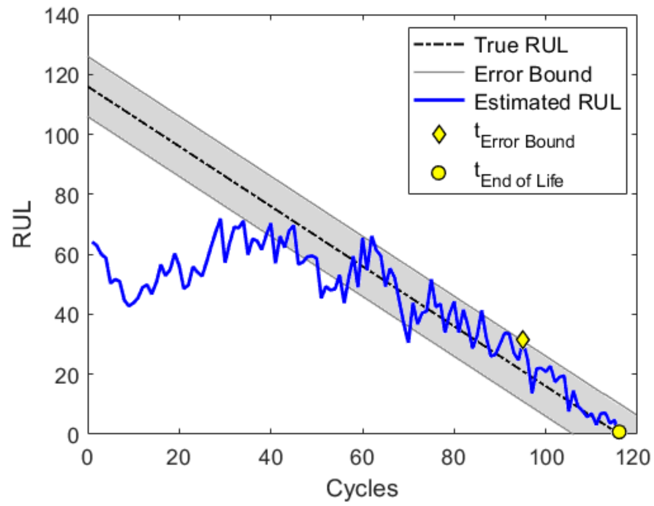

Fig. 9. RUL predictions using the Particle Filter algorithm

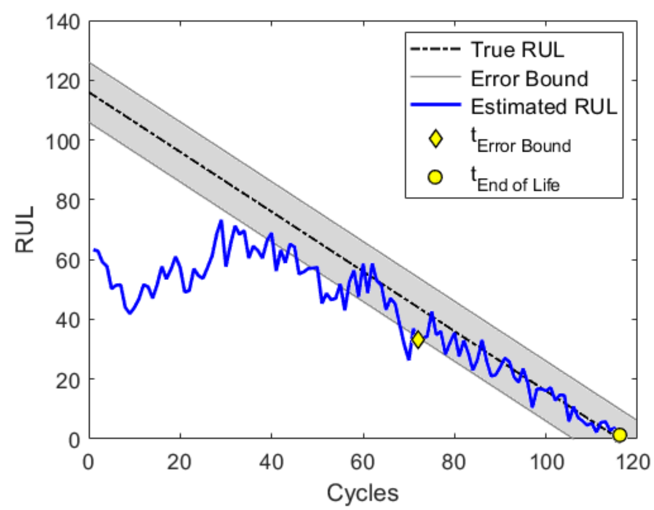

Fig. 10. RUL predictions using the proposed hybrid TLBO-PF algorithm

V. Conclusions

In this paper, we proposed a hybrid TLBO-PF metaheuristic algorithm. The main idea behind the proposed algorithm is to use the TLBO (Teaching-Learning Based Optimization) algorithm after the particle weight update step in the Particle Filter framework to reduce the effects of sample degeneracy and sample impoverishment, which are two known drawbacks of the original PF algorithm.

We compared the performance of the proposed algorithm with the performance of the original Particle Filter algorithm for estimating the degradation factors and predicting the Remaining Useful Life of a Lithium-ion battery. The degradation level of this battery type is affected by two degradation factors, related to the electrolyte resistance and the charge transfer resistance.

The root mean square error (RMSE) indicator was used to evaluate the performance of the algorithms for estimating the battery degradation level. To evaluate the performance for predicting the battery RUL, we used a modified prognostic horizon (MPH), which is a modified version of a prognostic-related performance indicator. The results show that the proposed algorithm presented a better performance for both tasks (degradation factor estimation and RUL prediction). However, the proposed TLBO-PF algorithm required a higher computational power in comparison with the standard PF algorithm. It is worth pointing out that the MPH indicator is sensitive to the error bound.

Future research may extend the scope of this paper by investigating new methods to incorporate the TLBO step into PF. For instance, an adaptive approach could be used to define, in each iteration of PF, which phases of TLBO should be used based on the position of particles.

Acknowledgment

The authors acknowledge the support of the Brazilian National Council for Scientific and Technological Development - CNPq (grant 423023/2018-7).

References

[1] W. O. L. Vianna and T. Yoneyama, “Predictive maintenance optimization for aircraft redundant systems subjected to multiple wear profiles,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 1170–1181, 2018.

[2] B. Saha, K. Goebel, S. Poll, and J. Christophersen, “Prognostics methods for battery health monitoring using a bayesian framework,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 2, pp. 291–296, 2009.

[3] S. Shahkar, Y. Shen, and K. Khorasani, “Diagnosis, prognosis and health monitoring of electro hydraulic servo valves (EHSV) using particle filters,” in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 385–390.

[4] B. Valeti and S. N. Pakzad, “Remaining useful life estimation of wind turbine blades under variable wind speed conditions using particle filters,” in *Proceedings of the Annual Conference of the PHM Society*, 2015, pp. 1–10.

[5] A. Qin, Q. Zhang, Q. Hu, G. Sun, J. He, and S. Lin, “Remaining useful life prediction for rotating machinery based on optimal degradation,” *Shock and Vibration*, vol. 2017, pp. 1–12, 2017.

[6] D. An, J. H. Choi, and N. H. Kim, “Prediction of remaining useful life under different conditions using accelerated life testing data,” *Journal of Mechanical Science and Technology*, vol. 32, no. 6, pp. 2497–2507, 2018.

[7] N. Raghavan and D. D. Frey, “Remaining useful life estimation for systems subject to multiple degradation mechanisms,” in *2015 IEEE Conference on Prognostics and Health Management (PHM)*, 2015, pp. 1–8.

[8] D. An, J. H. Choi, and N. H. Kim, “Prognostics 101: A tutorial for particle filter-based prognostics algorithm using matlab,” *Reliability Engineering and System Safety*, vol. 115, pp. 161–169, 2013.

[9] C. Wan, R. Huang, L. Huang, B. Wen, and T. Sato, “Damage identification using particle filters,” *Procedia Engineering*, vol. 188, pp. 41–47, 2017, structural Health Monitoring - From Sensing to Diagnosis and Prognosis.

[10] P. C. P. M. Pardal, H. K. Kuga, and R. V. de Moraes, “The particle filter sample impoverishment problem in the orbit determination application,” *Mathematical Problems in Engineering*, pp. 1–9, 2015.

[11] P. L. T. Duong and N. Raghavan, “A metaheuristic approach to remaining useful life estimation of systems subject to multiple degradation mechanisms,” in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2017, pp. 227–233.

[12] N. shan Zheng, L. shi Cai, H. fang Bian, and C. Lin, “Hybrid particle filtering algorithm for GPS multipath mitigation,” *Transactions of Nonferrous Metals Society of China*, vol. 24, no. 5, pp. 1554–1561, 2014.

[13] K. Schmal and H. Cheng, "Numerical study of a hybrid particle filter," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2015, pp. 1–4.

[14] R. V. Rao, D. P. Vakharia, and V. J. Savsani, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, pp. 303–315, 2011.

[15] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[16] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, pp. 710–720, 2013.

[17] C. Louen, S. X. Ding, and C. Kandler, "A new framework for remaining useful life estimation using support vector machine classifier," in *Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems*. Nice, France: Curran Associates, October 2013, pp. 228–233.

[18] P. Boškoski and D. Juričić, "Modeling localized bearing faults using inverse gaussian mixtures," in *Proceedings of the 2013 Annual conference of the prognostics and health management society*. New Orleans, Louisiana: PHM society, October 2013, pp. 107–113.

[19] J. Gu and M. Pecht, "Prognostics and health management using physics-of-failure," in *Proceedings of the 2008 Annual Reliability and Maintenance Symposium*. Las Vegas, Nevada: IEEE, January 2008.

[20] L. R. Rodrigues, "Remaining useful life prediction for multiple-component systems based on a system-level performance indicator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 141–150, 2018.

[21] A. Ragab, S. Yacout, M. Ouali, and H. Osman, "Multiple failure modes prognostics using logical analysis of data," in *2015 Annual Reliability and Maintainability Symposium (RAMS)*, 2015, pp. 1–7.

[22] O. Blancke, A. Tahan, D. Komljenovic, N. Amyot, M. L. evesque, and C. Hudon, "A holistic multi-failure mode prognosis approach for complex equipment," *Reliability Engineering & System Safety*, vol. 180, pp. 136–151, 2018.

[23] T. Fetzer, F. Ebner, F. Deinzer, and M. Grzegorzek, "Recovering from sample impoverishment in context of indoor localisation," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–8.

[24] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado, "Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3944–3954, 2014.

[25] X.-S. Yang, S. Deb, and S. Fong, "Metaheuristic algorithms: Optimal balance of intensification and diversification," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, pp. 977–983, 2014.

[26] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "Unified strategy for intensification and diversification balance in ACO metaheuristic," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 139–143.

[27] L. Gao, S. Liu, and R. A. Dougal, "Dynamic lithium-ion battery model for system simulation," *IEEE Transactions on Components and Packaging Technologies*, vol. 25, no. 3, pp. 495–505, 2002.

[28] C. Hu, G. Jain, P. Tamirisa, and T. Gorka, "Method for estimating capacity and predicting remaining useful life of lithium-ion battery," *Applied Energy*, vol. 126, pp. 182–189, 2014.

[29] Y. Chen, X. Liu, G. Yang, and H. Geng, "An internal resistance estimation method of lithium-ion batteries with constant current tests considering thermal effect," in *IECON 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 7629–7634.

[30] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, pp. 1–20, 2008.