Just-in-time Point Prediction Using a Computationally-efficient Lebesgue-sampling-based Prognostic Method: Application to Battery End-of-discharge Prediction

Camilo Reyes[1], Francisco Jaramillo[2], Bin Zhang[3], Chetan Kulkarni[4], and Marcos Orchard[5]

[1,2,5] *Department of Electrical Engineering, University of Chile, Av. Tupper 2007, Santiago, Chile*
camiloreyes@ug.uchile.cl
francisco.jaramillo@ing.uchile.cl
morchard@u.uchile.cl

[3] *Departament of Electrical Engineering, University of South Carolina, SC, 29208, USA*
zhangbin@cec.sc.edu

[4] *NASA Ames Research Center, Moffett Field, California, 94035, USA*
chetan.s.kulkarni@nasa.gov

ABSTRACT

Battery energy systems are becoming increasingly popular in a variety of systems, such as electric vehicles. Accurate estimation of the total discharge of a battery is a key element for energy management. Problems such as path planning for drones or road choices in electric vehicles would benefit greatly knowing beforehand the end of discharge time. These tasks are generally performed online and require continuously quick estimations. We propose a novel prognostic method based on a combination of classic Riemann sampling (RS) and Lebesgue sampling (LS) applied to a discharge model of a battery. The method utilizes an early and inaccurate prediction using a RS-based method combined with a particle-filter based prognostic. Once a fault condition has been detected, subsequent Just-in-Time Point (JITP) estimations are updated using a novel LS-based method. The JITP prediction are triggered when the Kullback-Leibler divergence between the probability density functions (PDF) of the long-term-based prediction and the last filtered state reaches a threshold. The CPU time needed to execute a procedure is used as a measure of the computational resources. Results show that this combined approach is several orders of magnitude faster than the classical prognosis scheme. The combination of these two methods provides a robust JITP prognosis with less computational resources, a key factor to consider in real-time applications in embedded systems.

1. INTRODUCTION

1.1. Riemann and Lebesgue Sampling

Sampling and sensing signals constitute a fundamental task in our modern digital technology. From basic signal processing in control systems to digital communications, sampling is a key factor to represent signals from nature in an efficient way. In the digital world, traditional sensing consists of an equal time sampling period which is known as Riemann sampling (RS) (Åström & Bernhardsson, 2002). RS method is widely used in computer-based controlled systems due to the low complexity in its analysis, design, and implementation (Åström & Bernhardsson, 2002; Miskowicz, 2016).

Data acquisition technology has been focused on uniform sampling periods where a great deal of theory has been done, although there have been alternative sampling techniques proposed. Despite these proposed sampling techniques, there have been few technologies that can directly support them (Dorf, Farren, & Phillips, 1962; Tsividis et al., 2016). One of these nonuniform sampling techniques is based on the Lebesgue theory, where the sampling is dictated dynamically.

The Lebesgue sampling method is one of the most common event-triggered sampling mechanism. The main feature of this strategy is that the signal is sampled every time it crosses a preassigned level. This sampling method is a way to reduce the number of data points required to describe a signal. This data reduction is an important factor when dealing with low memory embedded systems or network communications. Figure 1 illustrates an example of the difference between Riemann and Lebesgue sampling methods.

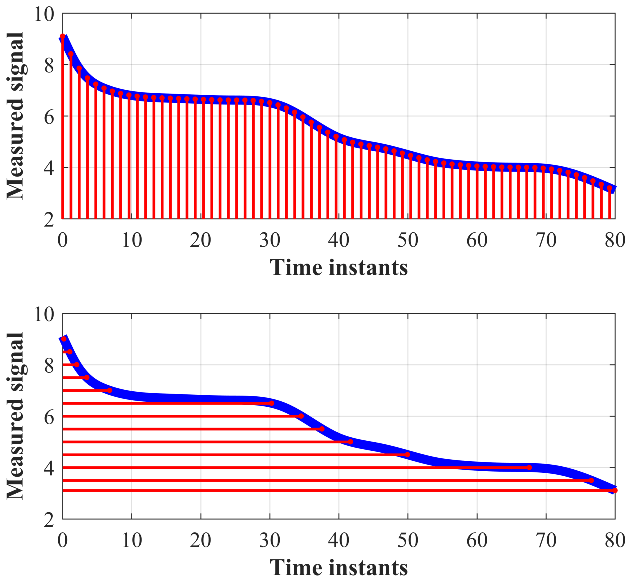

Figure 1. (A) Riemann Sampling (B) Lebesgue Sampling.

Specifically, in applications oriented to failure prognosis the time between the fault detection and failure occurrence can be considerable. So a constant monitoring may be redundant and resource consuming. In cases where fault diagnosis and prognosis (FDP) is difficult to implement using a traditional Riemann sampling (RS-FDP), a Lebesgue based sampling methodology may be a novel response to the problem of scarce computational resources.

1.2. PHM methods based on Lebesgue Sampling

The Lebesgue sampling (LS) method is characterized by the philosophy "used when needed" which results in a significant reduction of computational resources, especially for FDP algorithms with long prognosis horizon. This LS FDP method is based on the division of the state axis into a number of predefined states (the Lebesgue States) and the FDP algorithm will only be triggered when there is a change from one Lebesgue state to another, details see (Yan, 2017).

Applications of a Lebesgue based sampling method can be seen in a great number of studies. In (Tsividis, 2010) techniques for event-based sampling were reviewed showing the potential of this method in reducing energy consumption. In (Åström & Bernhardsson, 2002) a comparison is made between the Riemann and the Lebesgue sampling methods which were tested on a linear first order system. A numerical efficiency analysis is done in (Åström & Bernhardsson, 2003) comparing both sampling methods. Here they show that the Lebesgue method has better performance than a Riemann sampling based method. A PID Lebesgue based controller is presented in (Åarzén, 1999) showing a significant reduction in the CPU resources.

Applications of Lebesgue sampling method to prognosis has been a relatively new area of research. However, the increasing number of articles published shows that it is beginning to become an active field. The applications in fault diagnosis and prognosis problems(FDP) can be seen in (Zhang & Wang, 2014) where the method has been tested in a gearbox model. In (Yan, Dou, Liu, Peng, & Zhang, 2015), (Yan, Zhang, Wang, Dou, & Wang, 2016), (Yan & Zhang, 2016), (Yan, Zhang, & Orchard, 2016) and (Yan, Zhang, Zhao, Weddington, & Niu, 2017) are examples of a Lebesgue sample method for failure diagnosis and prognosis applied to battery models. Other applications spread from parameter optimization to extended Kalman Filter (EKF) for prognosis problems.

In this study, our main goal is to combine these two sampling techniques and quantify the accuracy of this joint method. We are also interested in measuring the computational resources they use. This joint method will be applied to the battery discharge estimation problem.

2. PROPOSED METHODOLOGY

2.1. Estimation and Prognosis based on Riemann Sampling

In order to estimate a system's state of health, a model has to be established through a suitable state variable (Souibgui, BenHmida, & Chaari, 2011). The next step is to characterize the evolution of the chosen model in order to forecast the remaining useful life (RUL). Despite there are many techniques to address the forecasting problem many researchers use Bayesian filters because it allows to include the notion of uncertainty.

The Bayesian filter proposed in this study is the particle filter (PF). This filter has been well established throughout the PHM community because it allows to manage uncertainty (details see (Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008)).

Failure prognostics involves predicting the SoH in future time instants where no measurements are available. The long terms predictions of the SoH allows the estimation of the remaining useful life (RUL) or the time of failure (ToF). In our case, we use the concept of ToF as defined in (Acuña & Orchard, 2017). The ToF consists in the time instant when a failure takes place and its estimation is done using the corrected expression, defined in the aforementioned article, given by Eq. 1.

$$P(F_k) = P(F_k | H_{k_p:k-1}) P(H_{k_p:k}) \tag{1}$$

where F_k is a failure occurrence at instant k and $H_{k_p:k-1}$ is the system staying healthy from the instant k_p until the instant $(k-1)$.

Using the mathematical formulation shown in (Acuña & Orchard, 2017), we calculate the PDF of the ToF. This PDF allows us to estimate the just in time point (JITP) and other statistical figures that can be used to analyze the evolution of the state of health (SOH). We use the concept of the JITP as defined in (Pola et al., 2015).

2.2. JITP model based on Lebesgue Sampling

Using the concept of the Lebesgue sampling we propose to analyze the evolution of the SOH of a system. This evolution is modeled through the JITP$_\alpha$ as it reaches the predefined Lebesgue states. The model will allow us to predict the behavior of the JITP$_\alpha$ during the prognosis stage. In our article the JITP$_\alpha$ = JITP$_{2.5}$.

The first step is to divide the state space into two different sets of Lebesgue states as suggested in Eq. (2):

$$\begin{cases} \Delta_F = \{L_1, L_2 ... L_f\}(filtering) \\ \Delta_P = \{L_{f+1}, L_{f+2} ... L_k\}(prognosis) \end{cases} \tag{2}$$

where the first set of Lebesgue states Δ_F is defined for the filtering stage (measurements available). The second set is defined during the prognosis stage. The first set of the Lebesgue states (where measurements are available) is used when a model can be generated. This model explains the dynamics of the state transition. Moreover, this model can be evaluated since measurements are available until the current time instant t_k. Once we create a model that explains the state transition it is assumed that this model holds for the prognosis stage. This supposition, however, depends deeply on the nature of the system analyzed.

It is important to note that the filtering process continues after the t_k instant even though a prognosis has already been made. This means that Lebesgue states that were in the set Δ_P become part of the set Δ_F, which allows us to update the model continuously. Particularly, in our case, 50% of the last JITP$_\alpha$ data (as a moving average) from the filtering process is used to construct the Lebesgue model.

Figure 2 shows the process used to generate the transition model from one Lebesgue state to another, using the JITP$_\alpha$. First finding the PDF of the ToF in each Lebesgue state, afterward calculate the JITP$_\alpha$ and finally model the JITP$_\alpha$ as a function of the Lebesgue states. This process can be summarized in following algorithm.

Algorithm 1:

1. Divide the systems states under monitoring into Lebesgue state sets for each stage of the process: Δ_F and Δ_P.

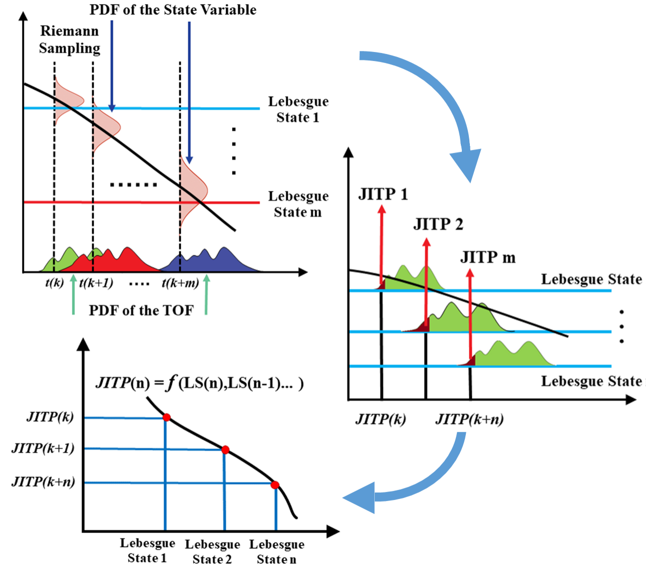

Figure 2. Three step algorithm employed to model the JITP$_\alpha$ as a function of the Lebesgue State.

2. Find the JITP$_\alpha$ of the ToF for each Lebesgue state L_i of the set Δ_F as done in the classical method.
3. Use the JITP$_\alpha$ calculated in previous step and find an empirical model between the JITP$_\alpha(L_i)$ and the L_i: $JITP_\alpha(L_i) = f(L_i)$.

2.3. Combination of RS and LS prognosis

This study proposes the combination of both forms of sampling to take advantage of the strengths each one has. As known the classical RS-FDP uses the information obtained until the instant t_k for diagnosis. The prognosis stage starts using the results obtained from the diagnosis as its initial conditions (Pola et al., 2015). Here, since we are using a Riemann sampling method, the classical prognosis has to propagate the state trajectories in time. This is done until it reaches a condition characterized by a particular failure state or threshold. Beyond this state the system is considered unable to perform normally. So, as before, the process is separated into two stages, the filtering stage and the prognosis stage. In our case, we use a particle filter for estimating the unknown fault state and to project this state estimation into future instants beyond t_k. When the system reaches the predefined failure state the time of failure (ToF) can be characterized by the JITP$_\alpha$.

After the the instant t_k it is important to keep in mind that the system can continue to evolve and therefore continue to produce measurements. Using a classical RS method it is possible to continue to update the ToF as more measurements are available. However this update has a high computational cost when using the RS method. Since we are using a particle filter to estimate the state we also have, through the weights, the PDF of the state estimation. As we obtain the new information we can compare the new states $posterior$ PDF with the PDF of the state in instant t_k (last PDF before the prognosis stage). Once the difference exceeds a certain threshold, a new prognosis process starts. However, this new prognostic is done using a model based on Lebesgue sampling, which was obtained during the filtering stage.

As known, once a fault is detected the prognostic system generates predictions updating them as new measures arrive. The prediction process runs continuously consuming computational resources and energy. Our proposed combined method decreases the use of computational resources and increases the prediction speed of the prognostic process.

Using this combined methodology we use the advantage of the precision of a prognostic based on the RS-FDP method and the speed of the methodology based on the LS-FDP method that allows a quick update of the ToF estimation. Using the model proposed in Section 3, we are able to quantify the accuracy of the proposed combined methodology and the CPU time usage.

Once the system reaches the t_k instant, our proposed method starts which can be summarized through the following algorithm:

Algorithm 2:

1. Generate a prognostic using the traditional method (RS methodology) after the instant t_k calculating the JITP at each Lebesgue State Δ_P previously defined.
2. Store the posterior PDF of the predicted state at each time instant.
3. Calculate the posterior PDF of the filtered state at each time instant as new PDF information arrives.
4. Compare the posterior PDFs of the prognostic and the filtered state at each time instant using the KL divergence, if the KL divergence surpasses a threshold go to 5, if not go to 3.
5. Execute Algorithm 1 updating the model parameters with all the information obtained from 3.
6. Calculate the JITP of the Lebesgue states using the model generated in 5.

Figure 3 shows a general representation of the proposed method summarized by *Algorithm* 1 and 2.

The Kullback-Leibler (KL) divergence indicated in *Algorithm* 2, this is used to measure the differences between two PDFs. Basically this measure shows the divergence from one PDF to another. If we have two distributions P_{t_k} and $P_{t_{k+n}}$ the KL divergence is defined as:

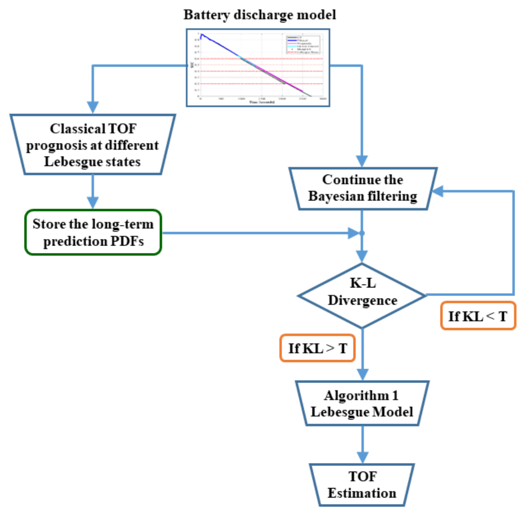

Figure 3. Proposed Algorithm.

$$D = \sum_{t_k} P_{t_k} log(\frac{P_{t_{k+n}}}{P_{t_k}}) \quad (3)$$

In our case we measure the state *posterior* PDF (after t_k) and the PDF of the last state before the prognosis stage (at t_k). Figure 4 shows an example of Kullback-Leibler divergence of the PDF at instant t_k) and the incoming PDF for $t > t_k$ with a case study of batter state of charge.

It is important to mention that the KL divergence is defined for continuous domains, so in order to use this divergence Gaussian kernels are used to estimate the continuous P_{t_k}.

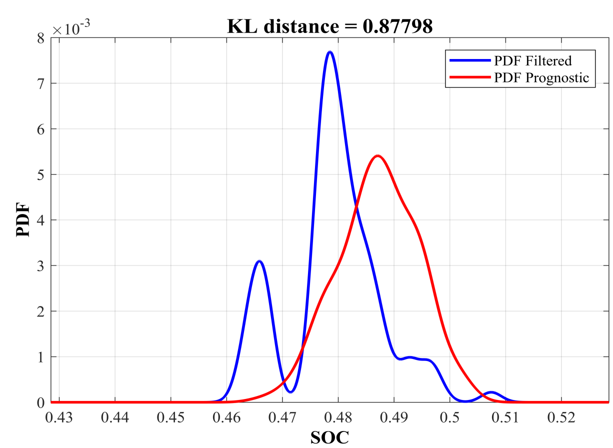

Figure 4. Kullback-Leibler divergence.

3. CASE STUDY: BATTERY STATE OF CHARGE ESTIMATION AND PROGNOSIS

3.1. State of Charge model

One of the main difficulties of estimating the state-of-charge (SOC) is the inability to measure this parameter directly. SOC can only be estimated through measurements of other parameters (Pattipati, Sankavaram, & Pattipati, 2011). Other issues that have to be taken into account when estimating these parameters are the dependency of factors such as temperature, age of the battery or hysteresis, etc (Tampier et al., 2015). More complex models such as multiphysics or chemical require many precise measurements (Charkhgard & Farrokhi, 2010) which make them less preferred. In contrast, the most popular models currently used are the ones based on the Ampere-hour counting or the open circuit voltage (OCV) measurements (Tampier et al., 2015). This last model, OCV model, also has the advantage that it does not need prior measurement information and is directly related with the SOC (Tang, Mao, Lin, & Koch, 2011).

The model used for this study was developed by (Pola et al., 2015) in which an empirical scheme of the model was validated with experimental data. The model is defined in Eqs. (4) and (5).

$$\begin{cases} x_1(t+1) = x_1(t) + \omega_1(t) \\ x_2(t+1) = x_2(t) - v(t).i(t)\Delta t E_{crit}^{-1} + \omega_2(t) \end{cases} \quad (4)$$

$$v(t) = v_L + (v_0 - v_L)e^{\gamma(x_2(t)-1)} + \alpha v_L(x_2(t)-1)... \quad (5)$$
$$+ (1-\alpha)v_L(e^{-\beta} - e^{-\beta\sqrt{x_2(t)}}) - i(t)x_1(t) + \eta(t)$$

Using this model, a Monte Carlo simulation (20,000 realizations) is conducted in order to have a ground truth at each Lebesgue State of interest.

3.2. JITP model in Estimation Stage

The system defined in Eq.(4) and Eq.(5) will be used to predict the JITP and at each predefined Lebesgue state. Also at each Lebesgue state the CPU time will be measured. Five hundred Monte Carlo realizations are carried out in order to have sufficient statistics for the prognosis stage. For each one of these simulations, the algorithm for the Lebesgue modeling is done accordingly to the previous section. Figure 5 shows the result of one of the trials, which is representative of the dynamics behavior.

As we can see, the dynamics is linear, and the prognosis is straightforward using a particle filter of 200 particles. Figure 5 shows clearly the two stages of our methodology, the filtering part (blue line), which is done continuously at each time period (cyan line after the prognosis starts). The red lines show some of the predefined Lebesgue states. As explained in the algorithm, the first set defined as Δ_F is used to create an empirical model between the JITP$_\alpha$ and the Lebesgue states.

Figure 5 also shows the dynamic behavior of the JITP$_\alpha$ at each Lebesgue state ($\alpha = 2.5\%$) during the prognosis process. As we can see the suggested relationship is akin to a linear function. So the model proposed is given by Eq. (6):

$$J(n) = \alpha L(n) + \beta \quad (6)$$

where $J(n)$ is the $n-th$ just-in-time-point, and $L(n)$ is the $n-th$ Lebesgue state. Since our algorithms are continuously filtering as more data becomes available, the parameters α and β may vary.

This model is obtained in each one of the 500 Monte Carlo realizations. Using the results from the Monte Carlo process we are able to estimate an average of errors and the computational resources used in the prognostic stage.

3.3. Prognosis results

Once the 500 Monte Carlo simulations are done, the next step is to analyze the efficiency of our proposed methodology. Two figures of merit are proposed as follows:

- The CPU time used to estimate the JITP of each selected Lebesgue state (Δ_P).
- The error of each sampling method (Lebesgue vs Riemann) at each Lebesgue state defined in Δ_P.

It is important to remark that the results shown from hereafter are averages obtained from the results of the 500 Monte Carlo simulations.

First, let us investigate how the model behaves when the JITP

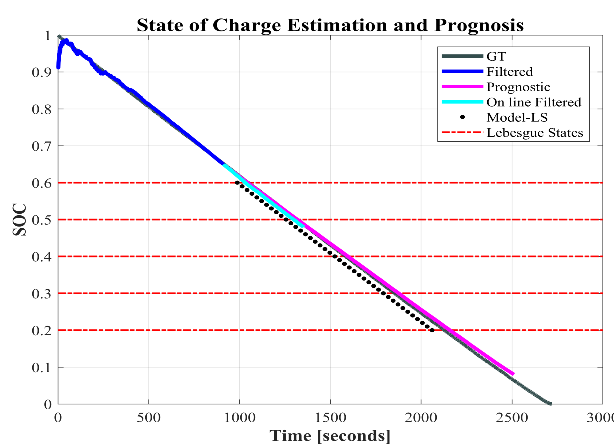

Figure 5. State dynamics of the model proposed.

of Lebesgue states are calculated for the prognosis state. Afterwards we will compare the results with the JITP obtained using the classical filtering method. Figure 6, shows the result (for the 500 Monte Carlo simulations) using our proposed method for the JITP estimation and a classical prognosis method.

It is clear that the classical prognosis method is inaccurate as the distance grows between the prognosis starting point and the Lebesgue states.

The error between both methods is used to compare and quantify the performance of each method. Figure 7 shows the increase of the relative error using the classical method as the Lebesgue state is further away from the initial prognosis point. Our proposed method apparently shows, for our case study, an upper bound and shows a more stable relative error than the classical method.

Figure 8 shows the comparison of PDF of the classical prognostic method, the PDF of the model, and the JITP from the Lebesgue model. Figure 9 shows the figure of merit in terms of the CPU time needed for each method in order to estimate the JITP after the 500 simulations.

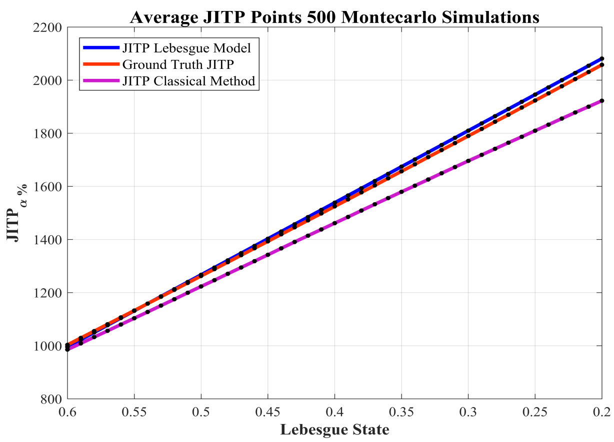

Figure 6. JITP estimation during prognosis stage for Lebesgue and Riemann Sampling.

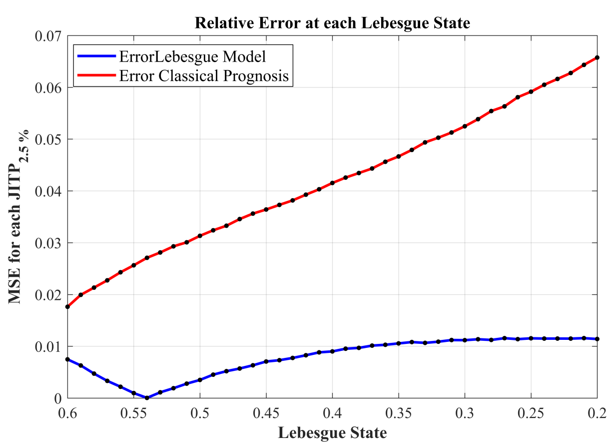

Figure 7. Relative MSE in the JITP estimation Lebesgue and Riemann Sampling methods.

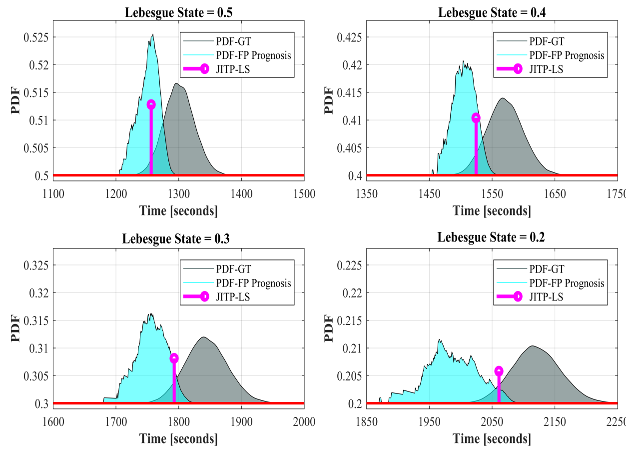

Figure 8. PDF and JITP at different Lebesgue states.

It is clear from Figure 9 that, as the distance increases between the Lebesgue state and the initial prognosis point (Figure 5), the computational time also increases for the classical prognostic approach. The Lebesgue model approach has the same CPU time. This is because the estimation of the model and the calculation of the JITP are calculated only once. The CPU time used in obtaining the model and the JITP is the one shown in Figure 9. We can see that despite the fact that two processes are done, the computational time of the proposed method is 2 orders of magnitude less than that of the classical method.

4. DISCUSSION

The results in Section 3show that, for the system analyzed, the use of a combined approach has advantages over classic

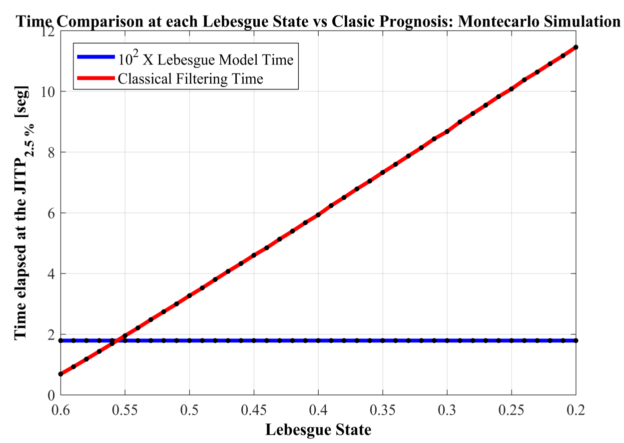

Figure 9. CPU time comparison between both approaches.

method. In our case, we use the classical method (based on Riemann sampling) to create a database of JITP later used for the development of the Lebesgue Model. The comparison of the relative error clearly demonstrates the advantages of the proposed method. Figure 9 show a different advantage of the Lebesgue based model in terms of CPU time, which is 2 orders of magnitude less than the classical estimation method.

For systems where computational resources are limited, the proposed combined methodology can be a solution. It is also important to note that the JITP estimation of our method shows a bounded accuracy loss.

Despite the usefulness of the proposed method, it is necessary to test this scheme on a nonlinear system where the Lebesgue model generated in the filtering phase can be a difficult issue to address and maybe more ad-hoc models will need to be developed. Another possible area of interest is nonlinear system where the relationship between fault and the Lebesgue State will need a closer analysis depending on the systems dynamics.

5. CONCLUSION

The proposed method shows a significant reduction of the CPU time. This indicates that this proposed combined method has the possibility to be used in embedded systems or where the communication between sensor-controller is scarce. In systems that have a CPU controlling and monitoring a process, estimating the ToF of the battery that energizes the sensor becomes essential. Our proposed method allows to continuously update the prognosis at different states without accuracy loss and with less computational resources used.

ACKNOWLEDGMENT

Thanks the partial support from CSIRO-CHILE 10CEII-9007 International Center of Excellence (Innova-Chile CORFO), Program 3-Project 1. The work of Francisco Jaramillo has been supported by CONICYT-PCHA/Doctorado Nacional/2014-21140201. We also want to acknowledge the support from the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008 and FONDECYT 1170044. The work of B. Zhang was partially supported by the ASPIRE grant program at the University of South Carolina.

REFERENCES

Åarzén, K.-E. (1999). A simple event-based pid controller. *IFAC Proceedings Volumes*, *32*, 8687–8692.

Acuña, D. E., & Orchard, M. E. (2017). Particle-filtering-based failure prognosis via sigma-points: Application to Lithium-Ion battery State-of-Charge monitoring. *Mechanical Systems and Signal Processing*, *85*, 827–848.

Åström, K. J., & Bernhardsson, B. (2003). Systems with Lebesgue Sampling. In A. Rantzer & C. I. Byrnes (Eds.), *Directions in mathematical systems theory and optimization* (pp. 1–13). Berlin, Heidelberg: Springer Berlin Heidelberg.

Åström, K. J., & Bernhardsson, B. M. (2002). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st ieee conference on decision and control* (Vol. 2, pp. 2011–2016 vol.2).

Charkhgard, M., & Farrokhi, M. (2010). State-of-charge estimation for lithium-ion batteries using neural networks and ekf. *IEEE Transactions on Industrial Electronics*, *57*(12), 4178-4187.

Dorf, R., Farren, M., & Phillips, C. (1962). Adaptive sampling frequency for sampled-data control systems. *IRE Transactions on Automatic Control*, *7*(1), 38-47.

Miskowicz, M. (2016). *Event-Based Control and Signal Processing*. CRC Press.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *2008 international conference on prognostics and health management* (pp. 1–6).

Pattipati, B., Sankavaram, C., & Pattipati, K. (2011, Nov). System identification and estimation framework for pivotal automotive battery management system characteristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *41*(6), 869-884.

Pola, D. A., Navarrete, H. F., Orchard, M. E., Rabié, R. S., Cerda, M. A., Olivares, B. E., ... Pérez, A. (2015). Particle-Filtering-Based Discharge Time Prognosis for Lithium-Ion Batteries With a Statistical Characterization of Use Profiles. *IEEE Transactions on Reliability*, *64*(2), 710–720.

Souibgui, F., BenHmida, F., & Chaari, A. (2011, March). Particle filter approach to fault detection and isolation in nonlinear systems. In *Eighth international multi-conference on systems, signals devices* (p. 1-6).

Tampier, C., Pérez, A., Jaramillo, F., Quintero, V., Orchard, M., & Silva, J. (2015). Lithium-Ion Battery End-of-Discharge Time Estimation and Prognosis based on Bayesian Algorithms and Outer Feedback Correction Loops: A Comparative Analysis. In *Annual conference of the prognostics and health management society 2015* (p. 14).

Tang, X., Mao, X., Lin, J., & Koch, B. (2011, June). Li-ion battery parameter estimation for state of charge. In *Proceedings of the 2011 american control conference* (p. 941-946).

Tsividis, Y. (2010). Event-driven data acquisition and digital signal processing- a tutorial. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *57*(8), 577-

581.

Tsividis, Y., Kurchuk, M., Martinez-Nuevo, P., Nowick, S. M., Patil, S., Schell, B., & Vezyrtzis, C. (2016). Event-based data acquisition and digital signal processing in continuous time. In *Event-based control and signal processing* (p. 353-378). CRC press.

Yan, W. (2017). *A lebesgue sampling based diagnosis and prognosis methodology with application to lithium-ion batteries* (Unpublished doctoral dissertation). University of South Carolina.

Yan, W., Dou, W., Liu, D., Peng, Y., & Zhang, B. (2015). Parameters adaption of lebesgue sampling based diagnosis and prognosis for lithium ion batteries. In *Annual conference of the prognostics and health management society 2015* (p. 9).

Yan, W., & Zhang, B. (2016). Extended kalman lter development in lebesgue sampling framework with an application to li-ion battery diagnosis and prognosis. In (p. 7).

Yan, W., Zhang, B., & Orchard, M. (2016). Parameters optimization of lebesgue sampling-based fault diagnosis and prognosis with application to li-ion batteries. In *Annual conference of the prognostics and health management society 2016* (p. 8).

Yan, W., Zhang, B., Wang, X., Dou, W., & Wang, J. (2016, March). Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, *63*(3), 1804-1812.

Yan, W., Zhang, B., Zhao, G., Weddington, J., & Niu, G. (2017). Uncertainty management in lebesgue-sampling-based diagnosis and prognosis for lithium-ion battery. *IEEE Transactions on Industrial Electronics*, *64*(10), 8158-8166.

Zhang, B., & Wang, X. (2014). Fault diagnosis and prognosis based on lebesgue sampling. In *Annual conference of the prognostics and health management society 2014* (p. 11).

BIOGRAPHIES

Camilo Reyes Has a Physics Engineer degree and researcher CSIRO-CHILE in predictive maintenance. He has contributed to generate and supervise multidisciplinary work teams focused on the area of heavy equipment, automatic control, and remote sensing systems in Ejército de Chile. Currently collaborates with professor Marcos Orchard as PhD student in Universidad de Chile.

Francisco Jaramillo received the B.Sc. degree in Electronics Engineering from Universidad de La Frontera, Temuco, Chile, in 2009. Currently he is a doctorate student at the Department of Electrical Engineering at the University of Chile under Dr. Marcos E. Orchard supervision. His research interests include machine learning, control systems, and estimation and prognosis based on Bayesian algorithms with applications to nitrogen removal in pilot-scale Sequencing Batch Reactors for Wastewater Treatment Plants.

Dr. Bin Zhang Received the B.E. and M.E. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. Before that, he was with R&D, General Motors, Detroit, MI, USA, with Impact Technologies, Rochester, NY, USA, and with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests are prognostics and health management and intelligent systems.

Dr. Chetan Kulkarni Is a staff researcher at the Prognostics Center of Excellence and the Diagnostics and Prognostics Group in the Intelligent Systems Division at NASA Ames Research Center. His current research interests are in Systems Prognostics and Health Management and specifically focused in the area of developing component level physics-based models, prognostics of electronic systems, battery prognostics and cryogenic fueling system prognostics and hybrid systems.

Dr. Marcos Orchard is Associate Professor with the Department of Electrical Engineering at Universidad de Chile and was part of the Intelligent Control Systems Laboratory at The Georgia Institute of Technology. His current research interest is the design, implementation and testing of real-time frameworks for fault diagnosis and failure prognosis, with applications to battery management systems, mining industry, and finance. His fields of expertise include statistical process monitoring, parametric/non-parametric modeling, and system identification. His research work at the Georgia Institute of Technology was the foundation of novel real-time fault diagnosis and failure prognosis approaches based on particle filtering algorithms. He received his Ph.D. and M.S. degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He received his B.S. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile. Dr. Orchard has published more than 100 papers in his areas of expertise.