Remaining Useful Battery Life Prediction for UAVs based on Machine Learning ⋆

Sina Sharif Mansouri * **Petros Karvelis** ** **George Georgoulas** * **George Nikolakopoulos** *

* *Robotics Group, Control Engineering Division of the Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden.*
** *Laboratory of Knowledge and Intelligent Computing, Department of Computer Engineering, Technological Educational Institute of Epirus, Arta, Greece*

Abstract: Unmanned Aerial Vehicles are becoming part of many industrial applications. The advancements in battery technologies played a crucial part for this trend. However, no matter what the advancements are, all batteries have a fixed capacity and after some time drain out. In order to extend the flying time window, the prediction of the time that the battery will no longer be able to support a flying condition is crucial. This in fact can be cast as a standard Remaining Useful Life prognostic problem, similarly encountered in many fields. In this article, the problem of Remaining Useful Life estimation of a battery, under different flight conditions, is tackled using four machine learning techniques: a linear sparse model, a variant of support vector regression, a multilayer perceptron and an advanced tree based algorithm. The efficiency of the overall proposed machine learning techniques, in the field of batteries prognostics, is evaluated based on multiple experimental data from different flight conditions.

Keywords: Battery, Remaining Useful Life, Machine Learning, UAVs, Prediction

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are gaining more and more attention, since they can be utilized in a wide range of applications, such as search and rescue missions, space exploration, security patrols, traffic control, agricultural vehicles, forest fire inspection, mapping and land mine removal. Moreover, recent developments have enabled miniaturized UAVs, such as quadrotors to become available for research and commercial applications (Wuthier et al., 2016; Mansouri et al., 2015). These UAVs could be used in outdoor and indoor experiments due to their mobility, safety and low cost (Alexis et al., 2012, 2010). A first fundamental step towards this vision is the development of reliable, online, methods for monitoring the state of the UAV's battery.

However, current batteries as a core part of the system depending on vehicle, payload, and wind conditions enable only flights up to 30 min for quadrotors (Ure et al., 2015), which can limit the usage of these UAVs for long time missions and experiments. Moreover, failure in battery can cause catastrophic events (Chen and Pecht, 2012) apart from affecting the overall functionality and performance of the system. Thus, it is essential that the batteries are operated safely and reliably under a wide range of payload and weather conditions, with the operator having a clear view of the time left before he needs to land the UAV.

⋆ This work has received partial funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS.

In general, there are different quantities that measure the state of a battery, with the State-of-Charge (SOC), which measures the amount of charge remaining in a battery, being one of the most common ones. Its estimation however is not an easy task. In any case the ultimate goal of predictive algorithms for batteries monitoring is to estimate the time till the End of Discharge (EOD) and End of Life (EOL). Such an estimation, which aims to find the EOD or EOL of a component is usually referred as the estimation of the Remaining Useful Life (RUL) (Si et al., 2011). In reality instead of the actual EOD, a threshold just before failure (discharge in our case) is used for the estimation of RUL, ensuring in this way the safe operation of the battery.

The most common taxonomy for describing the RUL prediction methods is the following: the Physics-of-Failure (PoF)-based approach and the data-driven approach. The PoF-based prognostic methods, as its name implies, relies on the expert knowledge of the underlying mechanisms and result into what is also known as model based methods. However, since the Li-ion battery is a dynamic, time-varying and nonlinear electrochemical system, the internal physical evolving mechanisms and failure modes are very complicated (Liu et al., 2015). Therefore, model based methods tend to be computationally complex making their applicability to real life settings questionable. On the other hand, data-driven approaches do not require extensive knowledge on battery chemical composition and failure mechanisms but they do require a representative set of data instead (Long et al., 2013). Machine Learning

(ML) (Eddahech et al., 2012), filtering (Orchard et al., 2013) and stochastic process (Liu et al., 2013) are the three main data-driven methodologies for RUL estimation, with the first one being under-explored.

In this article, the specific case of Li-ion batteries is considered, as they are used in wide range of applications, mainly due to some desirable properties, such as high energy density, high efficiency, slow material degradation, lack of memory effect, low self-discharge, and minimal maintenance requirements (Tulsyan et al., 2016). The main contributions of the article are the following. Firstly, the aim of the established scheme is to estimate the RUL of the battery during a flight mission. Unlike most studies encountered in the literature that focus on the prediction of SOC (Zhang and Lee, 2011; Charkhgard and Farrokhi, 2010; Burgos et al., 2015), the RUL prediction in our case is based on the measurement of the terminal voltage. As it was pointed out in (Saha et al., 2011), "most battery powered systems have a cutoff based on battery voltage and not SOC", which makes it a more natural choice to track the actual variable that defines the EOL. It should also be noted that the RUL refers to the time left for an individual cycle (estimating in other words the EOD) and not to the available remaining charging cycles.

Four different prediction methods from the field of ML are used for the RUL estimation: Least Absolute Shrinkage and Selection (LASSO) (Friedman et al., 2009), Multi-Layer Perceptron (MLP) (Haykin, 1994), Least Squares Support Vector machines for Regression (LS-SVR) (Suykens et al., 2002), and Gradient Boosted Trees (GBTs) (Ridgeway, 1999). Their selection was done in order to cover the most prominent methods of ML. LASSO gives rise to linear models, which however unlike conventional linear regression models are the outcome of an optimization process that produces sparse models leading generally to increased generalization performance and can be used as a baseline method for comparison with non-linear approaches. The LS-SVR was selected as a member of the kernel based methods family, which have been a very popular tool over the past twenty years, providing nonlinear mapping capabilities along with a guarantee of globally optimal solutions. The MLP is probably the most well known method from the field of ML and recently is living a renaissance with the deep learning paradigm. Finally the GBTs represent the family of ensemble learning methods, which perform remarkably well for a variety of problems (e.g. (Kaggle, 2016) competitions).

An additional contribution of this article is the fact that among the different methods, the GBTs are used for the first time in the related literature, achieving overall higher performance compared to the rest of the methods, suggesting that this method should be further explored. The final contribution stems from the utilization of information coming only from the voltage reading, contrary to other methods that use more than one measurements (Saha et al., 2011; Pola et al., 2015; Burgos et al., 2015).

The rest of the paper is structured as follows: Section 2 describes in brief the four ML methods. Section 3 presents the experimental test-bed as well as the process followed for the prediction of the RUL and the achieved results. Finally Section 4 concludes the paper offering also some guidelines for future research.

2. THE MACHINE LEARNING ALGORITHMS FOR RUL ESTIMATION

When it comes to RUL estimation two paths can be followed (Goebel et al., 2008): a) a model is built that tries to capture the evolution of the measured quantity (the quantity that quantifies the degradation of the system) and then it is recursively applied to propagate that quantity in time till it crosses a predefined threshold or b) a model is built that directly estimates the RUL based on measurements coming from different sources/sensors usually over a predefined time interval (Loutas et al., 2013). In both cases historical data is needed.

In this work the second path is followed, or in other words the RUL estimation is cast as a regression problem: a two minute sliding window with the present voltage measurement, as well as past voltage measurements are processed and used as an input to the ML model, which in the sequel tries to estimate the RUL at this point in time. In the rest of this section the theoretical background of all four methods is briefly presented.

Formally speaking, consider a training set of l pairs of samples $\{x_i, y_i | i = 1, ..., l\}$ with $i \in Z^+$, where each $x_i \in R^n$ are the input and $y_i \in R$ are the corresponding prediction values. Regression is the problem of finding a transformation f such that $X_{n\times m} \xrightarrow{f} Y_{n\times 1}$ in the best possible way. In our case the input is a set of processed voltage measurements and the output is the corresponding RUL.

2.1 Least Absolute Shrinkage and Selection Operator - LASSO

The LASSO is a method that performs both variable selection and regularization, which is also known as the basis pursuit in signal processing literature (Friedman et al., 2009; Chen et al., 2001). In general, the lasso estimate is defined by:

$$\hat{\beta}^{lasso} = \arg\min_{\beta}\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j) + \lambda\sum_{j=1}^{p}|\beta_j|\}$$
$$s.t.$$
$$\sum_{j=1}^{p}|\beta_j| \leq t \tag{1}$$

where β_j are unknown parameters or coefficients, y is the N-vector of outputs in the training set and $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage (the larger the value of λ, the greater the amount of shrinkage). The constant β_0 can be re-parametrized by standardizing the predictors and the solution for $\hat{\beta}^0$ is $\bar{y} = \frac{1}{N}\sum_1^N y_i$, therefore, the model is fitted without an intercept. The LASSO problem can be re-written in the following equivalent Lagrangian form as:

$$\hat{\beta}^{lasso} = \arg\min_{\beta}\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\} \quad (2)$$

where the penalty term of $\sum_{j=1}^{p}|\beta_j|$ is replaced in the minimization with the LASSO computation to be a quadratic programming problem.

2.2 Multi-Layer Perceptron - MLP

Multi-layer perceptron (MLP) neural networks are well known universal approximators (Bishop, 1995), while they are often utilized as estimation tools in place of the classical statistical methods. The MLP consists of a number of interconnected processing elements, which are called neurons. The neurons connect with each other through weighted connections and they are arranged in one or more layers. The data is fed to the MLP in an input layer and finally an output layer holds the response of the network to the input values. The most simpler case of a neural network is a 1-hidden layer with a number of neurons that maps a d-vector to an m-vector as:

$$y = b + W\tanh(s + Vx), \quad (3)$$

where x is the d-vector (input), V is a $k \times d$ matrix, s is a k dimensional vector, which is used to store the hidden units biases, b is an m dimensional vector that is used to store the output units biases, W is an $m \times h$ matrix and tanh is the inverse hyperbolic function. In order to train an MLP, we learn all parameters (biases and weights) of the model using the well-known back-propagation algorithm or some of its quicker variants (Duda et al., 2012).

2.3 Least Square Support Vector machines for Regression LSSVR

SVMs are a very popular method in ML. They can be used both for classification and regression. For the regression case, LS-SVR, a variant of SVR, can be described as a Quadratic Programming problem and can be cast as the following minimization problem: (Van Gestel et al., 2001):

$$\begin{aligned} \min J(w,b,\xi) &= \frac{1}{2}w^T w + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\ &s.t. \\ &y_i - [w^T\pi(x_i) + b] = \xi_i, \quad i = 1,...,l \end{aligned} \quad (4)$$

where w is the normal of the hyperplane, ξ_i is the error of the i^{th} training sample, $\pi(x_i)$ is a nonlinear function that maps x_i to a high dimensional space, C is a regularized parameter, provided by the user, that regulates the smoothness of the regression function. A large C reduces the training error but increases the model complexity, and therefore the training time and possibly generalization ability. A smaller C results in a less complex model with a higher mean square training error but possibly resulting in better generalization and b is a bias term. The Lagrangian of the previous constraint optimization problem is:

$$L(w,b,\xi,a) = J(w,b,\xi) - \sum_{i=1}^{l} a_i\{w^T\pi(x_i) + b + \xi_i - y_i\}, \quad (5)$$

where a_i are the Lagrangian multipliers.
Thus one could try to find the solution to the following system of linear equations:

$$\begin{pmatrix} 0 & e^T \\ e & \Omega + \frac{I}{C} \end{pmatrix}\begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} 0 \\ Y \end{pmatrix} \quad (6)$$

where $I \in R^{l\times l}$ is an identity matrix,

$$\begin{aligned} Y &= (y_1, y_2, ..., y_l)^T, \\ a &= (a_1, a_2, ..., a_l)^T, \\ e &= (1, 1, ..., 1)^T, \\ \Omega &= (\Omega_{ij}^T) = (k(x_i, x_j)) \\ k(x_i, x_j) &= \langle\pi(x_i), \pi(x_j)\rangle \end{aligned} \quad (7)$$

2.4 Gradient Boosted Trees - GBT

Among the various ML algorithms tree based methods, which partition the feature space into a set of disjoint regions R_j, $j = 1, 2, ..., K$ and then fit a model in each one them, are among the most popular methods in data mining (Wu et al., 2008). For the case of regression, the simplest and most common model assigns a constant value to each region as:

$$x \in R_j \Rightarrow f(X) = c_j \quad (8)$$

Therefore, formally a tree can be expressed as

$$T(x) = \sum_{j=1}^{J} c_j I(x \in R_j) \quad (9)$$

where $I(.)$ is the indicator function.

Their popularity stems from a number of properties that all tree based algorithms poses: they can handle data of mixed type (continues, categorical, etc.) as well as data with missing values, they are immune to the presence of outliers as well as to the presence of irrelevant inputs and on top of that they scale nicely (Friedman et al., 2009). On the other hand, their predictive power is usually inferior compared to other state of the art ML algorithms, such as neural networks and SVMs (Friedman et al., 2009).

However tree based methods can produce high accurate predictions if grouped together in the form of an ensemble. One such approach is to use of the boosting principle, to create gradient boosted machines (Friedman, 2001). The approach starts by building a simple model and then stagewise adds models that aim to explain observations that are modeled poorly by the existing trees of the ensemble ending up with a model of the form:

$$f_M(x) = \sum_{m=1}^{M} T_m(x) \quad (10)$$

where M is the total number trees comprising the ensemble and T_m is the m^{th} member of the ensemble. The GBTs algorithm is summarized in the following Algorithm 1 (adopted from (Friedman et al., 2009))

3. EXPERIMENTAL RESULTS

The proposed methods are evaluated with the utilization of the Ascending Technologies NEO hexacopter. This platform has an onboard Intel NUC computer utilizing a Core i7-5557U combined with 8 GB of RAM and is capable of providing a flight time of 26 min. The system is equipped with 4×4.2 V lithium polymer batteries and the voltage is measured by onboard sensor. The UAV flight in different

Algorithm 1 Gradient Boosted Trees Algorithm

(1) Initialize $f_0 = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$
(2) For m= 1 to M
  (a) For $i = 1, 2, ..., N$ compute

$$r_{im} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f=f_{m-1}}$$

  (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, ..., J_m$
  (c) For $j = 1, 2, ..., J_m$ compute

$$c_{jm} = \arg\min_c \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + c)$$

  (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} c_{jm} I(x \in R_{jm})$
(3) Output $\hat{f}(x) = f_M(x)$

indoor and outdoor experiments and the battery voltage is stored. The indoor experiment has the volume of $5 \times 5 \times 3$ m and the outdoor experiment has the volume of $3 \times 3 \times 11$ m.

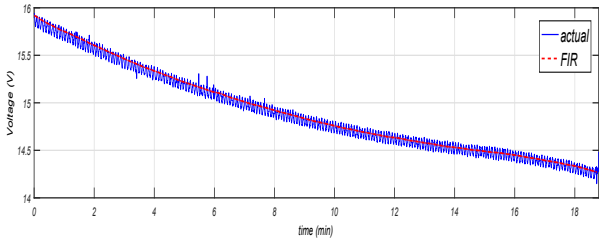

Fig. 1. The actual measured voltage compared to the processed quantity.

Six data sets were collected, all of them coming from different batteries, with the recordings of the output voltage as a function of time. As it can be observed in Figure 1, the voltage fluctuates constantly around an underlying exponential like trend. This fluctuation on one hand can pose an extra challenge during the modeling process while on the other hand could mask the underlying trend which is in fact the quantity that is of more interest. To eliminate this fluctuations the raw measurements are averaged over non overlapping three second windows and they are filtered using a Finite Impulse Response (FIR) filter. As it can be seen in Figure 1 the proposed procedure, captures the underlying discharge trend eliminating the non-essential fluctuations.The processed signal is used as input to each one of the aforementioned ML models. Therefore the overall procedure can be summarized as follows :

- The raw voltage measurement contained in a sliding two minute window is replaced (one could say "down-sampled" without however the strict definition) by the average values over non overlapping three second windows.
- The new signal is passed through a 10th order low-pass FIR filter.
- The output of the FIR filter forms the input vector (x_i) (of dimension 40 - two minutes divided into three second windows) for the ML methods, which provide as output a single $(\hat{y}_i)$ estimate of the RUL

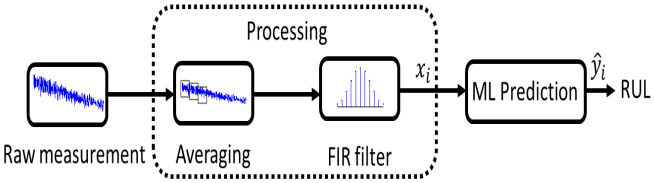

Fig. 2. Schematic of the proposed method.

Since none of the aforementioned methods is parameter-free, each time five of the recordings were used for tuning and training the models and then the trained model was applied for the prediction of the RUL of the recording which was left out. In other words the standard leave one out method was applied for the estimation of the performance of each model. The tuning was done using only training data completely decoupling the tuning from the estimation process to eliminate potential bias (Japkowicz and Shah, 2011).

The minimum and the maximum voltages are $14\,V$ and $16.8\,V$ respectively, which means that the effective voltage range is 2.8. In all experiments the UAV was grounded before the battery was completely depleted. In the failure prognosis literature, the prognostic algorithm is triggered after the diagnosis of a fault. In our case the discharging process is a natural occurring phenomenon and therefore no fault is taking place. Instead in order to initiate the prediction process, a threshold set equal to the minimum voltage plus 50% of the voltage range is used to trigger the prediction process. Accordingly to define the RUL and in order to make sure that a catastrophic event is avoided (crash of the UAV) a threshold set equal to the minimum voltage plus 15% of the voltage range is treated as the minimum acceptable voltage value.

The performance of the four algorithms is graphically depicted in Figure 3 and also quantified using Precision (11) and Mean Absolute Percentage Error (MAPER) (12), which are two popular, yet not the only available, measures (Saxena et al., 2010). The results indicate that GBTs are performing the best for this study, with the LASSO performing the worse. In fact LASSO continuously overestimates the RUL which can lead to catastrophic events. In the one case that LASSO seems to accurately predict the RUL at the beginning of the prediction horizon when it gets closer to the actual EOD it starts making unrealistic optimistic predictions. From the other methods the MLP manifests relatively good prediction performance, while the LS-SVR shows signs of over fitting, creating unstable predictions.

$$\text{Precision} = \sqrt{\frac{\sum_{t=1}^{N}(y(t) - \hat{y}(t))^2}{N-1}} \tag{11}$$

$$\text{MAPER} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{100\hat{y}(t)}{RUL_{actual}(t)}\right| \tag{12}$$

4. CONCLUSION

This work presented ML approaches for the online, during flight, prediction of the RUL of a Li-ion battery using only the voltage measurements. The results indicate that

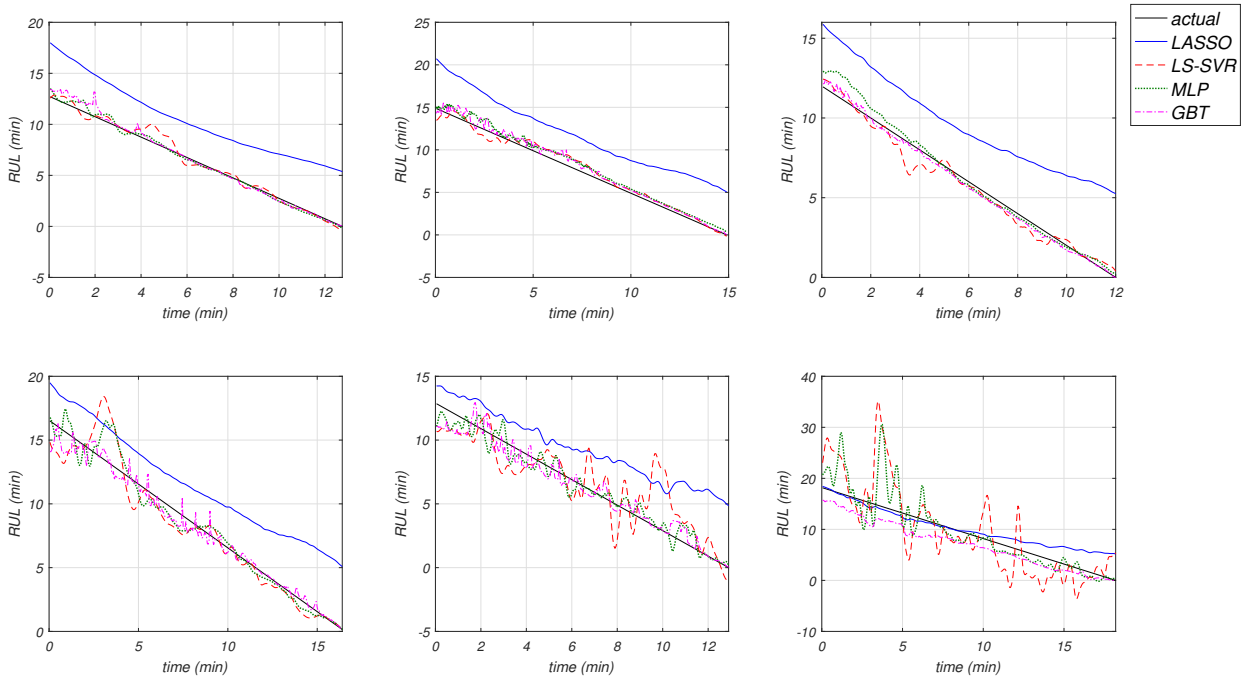

Fig. 3. The results of the RUL prediction for the six data sets using the four ML methods. The closer the line to the ground truth RUL (black line) the better. The linear model in all but one cases overestimates heavily the RUL.

Table 1. Precision of the methods.

#	GBT	LS-SVR	MLP	LASSO
1	0.48	0.54	0.31	4.11
2	0.62	0.64	0.68	4.34
3	0.29	0.58	0.55	3.67
4	0.65	1.44	0.95	3.38
5	0.77	1.86	0.71	3.13
6	2.26	5.49	3.47	2.09
Average	0.85	1.76	1.11	3.45

Table 2. Mean Absolute Percentage Error

#	GBTs	LS-SVR	MLP	LASSO
1	99.59	101.80	96.63	332.92
2	109.66	104.92	120.30	304.39
3	96.33	108.95	107.22	334.94
4	99.75	95.52	97.87	271.87
5	103.76	136.45	109.53	306.79
6	70.33	177.22	98.18	233.66
Average	96.57	120.81	104.95	297.43

non-linear methods usually outperform linear ones, even if the latter comes from an optimization procedure such as LASSO. Among the nonlinear models the GBTs performed the best. GBTs have recently become very popular due to their winning in various data mining competitions (Kaggle, 2016). The results confirm that GBTs are very powerful nonlinear approximators that can be used for the prediction of the RUL of batteries.

On the other hand we must note that this was a preliminary investigation. In future work the time length of the involved window will be optimized and more elaborate schemes for the selection of inputs to ML will be explored for finding the "optimal" setting for RUL prediction. Moreover since other parameters, like current, ambient temperature and UAV load, convey useful information regarding the RUL, a multi parametric input will also be considered. Moreover the required processing power and power consumption aspects of each algorithm will be evaluated to assess its applicability on the UAV's embedded processor. Finally as it was stressed in Section 2 the ML methods were trained to automatically predict the RUL. However there is a second path that involves the prediction of the quantity that it is actually measured. This approach is more aligned to engineering reasoning and can be integrated within an Model Predictive Control (MPC) framework. Therefore in future work the two approaches will be compared using also more experimental data.

REFERENCES

Alexis, K., Nikolakopoulos, G., and Tzes, A. (2010). Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts. In *Control & Automation (MED), 2010 18th Mediterranean Conference on*, 1461–1466. IEEE.

Alexis, K., Nikolakopoulos, G., and Tzes, A. (2012). Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications*, 6(12), 1812–1827.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Burgos, C., Sáez, D., Orchard, M.E., and Cárdenas, R. (2015). Fuzzy modelling for the state-of-charge estimation of lead-acid batteries. *Journal of Power Sources*, 274, 355–366.

Charkhgard, M. and 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.

Chen, C. and Pecht, M. (2012). Prognostics of lithium-ion batteries using model-based and data-driven methods. In *Prognostics and System Health Management (PHM), 2012 IEEE Conference on*, 1–6. IEEE.

Chen, S.S., Donoho, D.L., and Saunders, M.A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1), 129–159.

Duda, R.O., Hart, P.E., and Stork, D.G. (2012). *Pattern classification*. John Wiley & Sons.

Eddahech, A., Briat, O., Bertrand, N., Delétage, J.Y., and Vinassa, J.M. (2012). Behavior and state-of-health monitoring of li-ion batteries using impedance spectroscopy and recurrent neural networks. *International Journal of Electrical Power & Energy Systems*, 42(1), 487–494.

Friedman, J., Hastie, T., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 1. second edition.

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Goebel, K., Saha, B., and Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. In *62nd meeting of the society for machinery failure prevention technology (mfpt)*, 119–131.

Haykin, S. (1994). Neural networks, a comprehensive foundation.

Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

Kaggle (2016). `https://www.kaggle.com/`.

Liu, D., Pang, J., Zhou, J., Peng, Y., and Pecht, M. (2013). Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectronics Reliability*, 53(6), 832–839.

Liu, D., Zhou, J., Pan, D., Peng, Y., and Peng, X. (2015). Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning. *Measurement*, 63, 143–151.

Long, B., Xian, W., Jiang, L., and Liu, Z. (2013). An improved autoregressive model by particle swarm optimization for prognostics of lithium-ion batteries. *Microelectronics Reliability*, 53(6), 821–831.

Loutas, T.H., Roulias, D., and Georgoulas, G. (2013). Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. *IEEE Transactions on Reliability*, 62(4), 821–832.

Mansouri, S.S., Nikolakopoulos, G., and Gustafsson, T. (2015). Distributed model predictive control for unmanned aerial vehicles. In *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on*, 152–161. IEEE.

Orchard, M.E., Hevia-Koch, P., Zhang, B., and Tang, L. (2013). Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, 60(11), 5260–5269.

Pola, D.A., Navarrete, H.F., Orchard, M.E., Rabié, R.S., Cerda, M.A., Olivares, B.E., Silva, J.F., Espinoza, P.A., and 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.

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics*, 172–181.

Saha, B., Koshimoto, E., Quach, C.C., Hogge, E.F., Strom, T.H., Hill, B.L., Vazquez, S.L., and Goebel, K. (2011). Battery health management system for electric uavs. In *Aerospace Conference, 2011 IEEE*, 1–9. IEEE.

Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, 1(1), 4–23.

Si, X.S., Wang, W., Hu, C.H., and Zhou, D.H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1–14.

Suykens, J., Van Gestel, T., and De Brabanter, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Tulsyan, A., Tsai, Y., Gopaluni, R.B., and Braatz, R.D. (2016). State-of-charge estimation in lithium-ion batteries: A particle filter approach. *Journal of Power Sources*, 331, 208–223.

Ure, N.K., Chowdhary, G., Toksoz, T., How, J.P., Vavrina, M.A., and Vian, J. (2015). An automated battery management system to enable persistent missions with multiple aerial vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(1), 275–286.

Van Gestel, T., Suykens, J.A., Baestaens, D.E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., and Vandewalle, J. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on neural networks*, 12(4), 809–821.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1), 1–37.

Wuthier, D., Kominiak, D., Kanellakis, C., Andrikopoulos, G., Fumagalli, M., Schipper, G., and Nikolakopoulos, G. (2016). On the design, modeling and control of a novel compact aerial manipulator. In *Control and Automation (MED), 2016 24th Mediterranean Conference on*, 665–670. IEEE.

Zhang, J. and Lee, J. (2011). A review on prognostics and health monitoring of li-ion battery. *Journal of Power Sources*, 196(15), 6007–6014.