Learning Aerial Vehicle Model Parameters for use in Prognostic Decision Making

Edward Balaban
Stanford University, Stanford, CA, 94305, USA
ebalaban@stanford.edu

***Abstract*—The field of Prognostic Health Management (PHM) has been undergoing rapid growth in recent years, with the development of increasingly sophisticated techniques (some based on machine learning methods) for diagnosing faults in system components and estimating fault progression trajectories. Research efforts on how to utilize prognostic health information (e.g. for extending the remaining useful life of the system, increasing its safety, or maximizing its operational effectiveness) are mostly in their early stages, however. The process of using prognostic information to determine a systems actions or its configuration is beginning to be referred to as Prognostic Decision Making (PDM).**

This project aimed to expand previous work in PDM conducted at NASA Ames Research Center and Stanford University. Its primary objective was to develop methods for automatically learning vehicle model parameters from experimental data. The vehicle model will then be utilized in a PDM system based on continuous Partially Observable Markov Decision Processes (POMDP). Stochastic algorithms (such as those based on Sequential Monte Carlo principles) will use the POMDP framework for performing decision-making in scenarios involving route replanning or system reconfiguration after an in-flight fault has been detected.

I. Introduction

As aircraft and spacecraft become more complex and their missions more demanding, it is becoming increasingly challenging for even the most experienced pilots, controllers, and maintenance personnel to analyze changes in vehicle behavior that can indicate a fault and accurately predict the short- and long-term effects that a fault can cause. Therefore, some of the latest vehicle designs begin to incorporate automated fault diagnostic and prognostic methods that can assist with these tasks. The next research frontier is in how to effectively utilize diagnostic and prognostic information about vehicle health in making autonomous or semi-autonomous decisions concerning operations and maintenance. This research area is beginning to be referred to as Prognostic Decision Making (PDM) and the goal of this project is to extend the work on PDM conducted at NASA Ames Research Center and Stanford University (Balaban & Alonso, 2013).

The main goal of the project described in this report was to identify a method or a set of methods that provide a robust way of deriving physics models needed for PDM reasoning from large, diverse experimental data sets, with minimal amount of hand-tuning required. The data used in the project has been collected on an electrically-powered Edge 540 unmanned aerial vehicle (UAV) flown at NASA Langley Research Center (Hogge, Quach, Vazquez, & Hill, 2011). The data sets contain aerodynamic (e.g. airspeed, angle of attack, altitude) as well electrical and thermal measurements (collected on the vehicle's batteries and motors). The data was collected during a series of flights lasting up to 30 minutes, under a variety of load and environmental conditions. While a significant effort has been put into ensuring data acquisition quality, the typical problems associated with experimental data collection are present in the data sets (e.g. noise, drop-outs, and inconsistencies) and need to be properly handled.

The UAV modeling framework consisted of linear, non-linear, and differential equations describing the overall aerodynamics, as well as the dynamics of the power and propulsion systems. Model parameters included thermal transfer coefficients, electrical loss coefficients, several internal resistance coefficients for the batteries, mechanical friction coefficients, etc. While it is possible to estimate some of parameters with direct measurements on the ground (e.g. the electrical loss coefficients), others appeared to be more suitable for derivation from in-flight data.

There were two main challenges to be contended with during the project. First, the model structure was dictated by physical processes being described, therefore terms could not be added or removed arbitrarily to achieve a better fit to the training data. Second, the objective functions constructed using the model's equations were non-linear and non-convex, requiring use of appropriate regression techniques.

Once developed, the model will be used as part of the reasoning architecture depicted on Figure 1. The architecture consists of the four main components: the diagnoser (DX, for diagnostics), the decision maker (DM), the vehicle simulator (VS), and the vehicle itself.

The PDM process will be initiated with an input route p_0 and an initial fault set F_0 supplied to the decision maker. This can be done at the beginning of a mission (in that case the fault set may be empty) or if a fault is diagnosed in flight. A fault set F is one or more fault descriptors. A fault descriptor f_i consists of a fault type d and a fault magnitude m.

DM will utilize VS to evaluate and, optionally, optimize the input route. An initial optimization can be performed if a successful completion of the input route is deemed unlikely. Alternative routes sent to VS will consist of an ordered set of waypoints (with vectors Γ of specific values for waypoint parameters). DM would also inform VS of the relevant fault modes. VS will simulate the candidate path, p_c, and return the reward and cost estimates for it. Once DM finalizes the route recommendation p^*, it will be sent to the vehicle for execution.

As the mission proceeds, DX will continue to monitor observations Z (e.g. sensor readings) generated by the vehicle to detect any new fault conditions. A new fault condition will trigger a reevaluation of the vehicle route by DM. Another

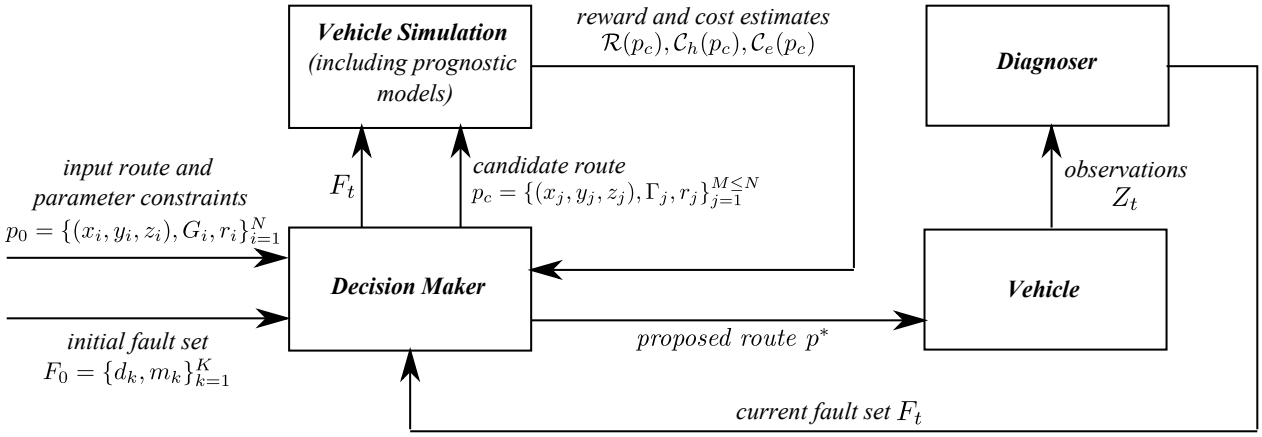

Figure 1. PDM reasoning architecture

event type that will trigger a reevaluation is a significant deviation of the predicted component degradation rates from the rates observed as the flight progresses further.

The modeling framework is described in Section II. Section III provides more details on the experimental data used. The approach taken for learning model parameters and the results obtained are described in Section IV. Integrated tests of the new model and the decision-making algorithms are discussed in Section V. Section VI summarizes the work performed and outlines directions for future work.

II. Vehicle Model

Given an estimate of the current vehicle state and the desired action, the vehicle simulator uses the model to generate an estimate of the next state. The state vector includes the aircraft position, velocity, acceleration, orientation, lift, drag, thrust, battery voltage, battery charge remaining, component temperatures, and other data. When an entire path p is provided, the simulator will be able to generate path reward $\mathcal{R}(p)$ and cost $\mathcal{C}(p)$ estimates (Figure 1).

Since the focus of this project was primarily on electrical and thermal models of the vehicle propulsion system, the aerodynamic components of the model are omitted here and can be found in (Balaban & Alonso, 2013). In the propulsion portion of the model there are currently three elements with non-linear, action-dependent degradation aspects: battery voltage, battery temperature, and motor temperature. The following subsections describe them in more detail.

A. Electrical model of the batteries

The battery model used in this work is based on an electrical circuit equivalent shown in Figure 2 and is adapted from the work by Daigle, Saxena, and Goebel (2012). Three main processes are captured in the model: the *ohmic drop* (also known as the *I-R drop*), the *parasitic resistance* (accounting for self-discharge), and the *concentration polarization*. Out of the three, the concentration polarization resistance is the primary contributor to the non-linearity of the battery output voltage as a function of its state of charge. The large capacitance C_b holds the charge q_b of the battery. The R_{CP}-C_{CP} pair captures the major nonlinear voltage drop due to concentration polarization, R_s-C_s pair captures the I-R drop, and R_p models the parasitic resistance that accounts for self-discharge.

The state of charge (SoC) is defined as

$$SoC = 1 - \frac{q_{max} - q_b}{C_{max}}, \tag{1}$$

where q_b is the current charge in the battery (assumed to be held by capacitance C_b), q_{max} is the maximum possible charge, and C_{max} is the maximum possible capacity. The concentration polarization resistance is expressed as:

$$R_{CP} = R_{CP0} + R_{CP1} \exp\left(R_{CP2}(1 - SoC)\right), \tag{2}$$

where R_{CP0}, R_{CP1}, and R_{CP2} are model parameters. The resistance, and, therefore, the voltage drop, increases exponentially as SoC decreases.

Voltage drops across the individual circuit elements are given by

$$V_b = q_b/C_b; V_{CP} = q_{CP}/C_{CP}; V_s = q_s/C_s; V_p = V_b - V_{CP}, \tag{3}$$

where q_{CP} is the charge associated with the capacitance C_{CP}. The terminal voltage of the battery is described as

$$V = V_b - V_{CP} - R_s i, \tag{4}$$

with i being the current at the terminals. Currents associated with the individual circuit elements are:

$$i_p = V_p/R_p; i_b = i_p + i; \text{ and } i_{CP} = i_b - V_{CP}/R_{CP}. \tag{5}$$

The charges are then expressed as

$$\dot{q}_b = -i_b \text{ and } \dot{q}_{CP} = i_{CP}. \tag{6}$$

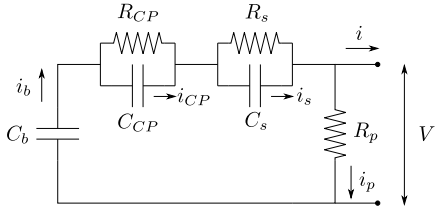

Figure 2. Battery equivalent circuit

The work conducted during this project has been focused on learning the best values for Rs, R_p, R_{cp0}, R_{cp1}, R_{cp2}, C_s, C_b, and C_{cp}.

B. Thermal model of the motors and the batteries

Being able to predict battery temperatures is important since excessive temperatures can lead to premature capacity degradation and, beyond a certain point, to thermal runaway and battery failure (Chen, Song, & Evans, 1996). In an electric motor overheating can also result in failure due to stator winding insulation damage or rotor magnet delamination (Milanfar & Lang, 1996). Battery and motor temperatures are estimated with the following model:

$$dT = \frac{1}{C_t}(RI^2 + h(T_a - T))dt, \tag{7}$$

where R is the electrical resistance of the component, C_t is the thermal inertia coefficient, h is the thermal transfer coefficient, I is the current, T is the component temperature, and T_a is the ambient temperature.

III. Experimental Data

Data from 20 UAV flights was used in this work. There are four batteries on-board the aircraft (Thunder Power G6 Pro Lite, 7800mAh, 5-cell, 20V nominal terminal voltage). Two serially-connected batteries supply current to each of the motors. In the data sets the batteries are labeled LLF20, ULA20, LRF40, and URA40 (lower-left-front-20V, upper-left-aft-20V, lower-right-front-40V, and upper-right-aft-40V, respectively). Current draw, terminal voltage, and surface temperature for each battery were recorded.

The motors powering the aircraft are AXI 5330/F3A outrunner motors. The two motors are serially connected through the propeller shaft and drive a single propeller. The motors are labeled as AFTMTR and FWDMTR in the data sets. Current going into each of the motors, rpm, and temperature on the support bracket between the motors are recorded.

IV. Parameter Learning Approach and Results

Since a non-linear hypothesis function (i.e. our system model) is utilized, the linear regression methods are not directly applicable. Instead, Levenberg-Marquardt algorithm, or LMA (also known as the damped least-squares method), was first utilized (Marquardt, 1963). LMA algorithm adaptively switches the parameter updates between the gradient descent update and the Gauss-Newton update. Gradient descent update is used when the algorithm is assumed to be far from a function minimum. As the solution approaches the minimum, the update used by LMA starts to approach that of the Gauss-Newton method. LMA will only find a local minimum, however, if the objective function is not convex. This was the case for the optimization problems posed in this project, therefore the initial guess at the parameter values became important.

As an alternative to LMA, a global optimization method, Simulated Annealing (SA), was investigated as well. SA (Bertsimas & Tsitsiklis, 1993) is inspired by metallurgical process of heating a material and then slowly cooling it down, thus decreasing the occurrence of defects. In the algorithms an *annealing schedule* is specified which is used to systematically decrease the *temperature* parameter. The *temperature* is used to control the extent of the search (i.e. how far from the current point to look for the next possible solution) and the probability of accepting a new point that increases the objective instead of lowering it. As the *temperature* decreases, exploration of the search space gets more and more restricted. While this procedure does not guarantee optimality, it does help to avoid getting stuck in a local minimum.

For the purposes of this project, numerical LMA and SA implementations from MATLAB's Optimization and Global Optimization toolboxes, respectively, were used (*MATLAB version 7.11.0.584 (2010b)*, 2010).

Numerous training experiments were conducted with LMA and SA (on both thermal and electrical parameter sets and with various algorithm settings) and the expected strengths and weaknesses of each indeed demonstrated themselves. LMA proved to be sensitive to the initial choice of parameters and SA would sometimes take many iterations to converge or hit the set limit on iterations without converging.

In the end, the following procedure was settled on for selecting the parameter values:

1) For each component 8 data sets were selected for training, 2 for validation, and 2 for testing (some of the original data had to be excluded due to artifacts that could not be sufficiently corrected with preprocessing).
2) The expected range of parameter values was established and constraints defined accordingly.
3) SA was executed using the constraints (and the default parameters) 10 times. Random values from within the expected ranges were picked as the starting points.
4) LMA was then run 10 times with each of the SA solutions as the starting point (occasionally some of the solutions were the same).
5) Solutions found by LMA were then tested on the validation sets and the one with the lowest validation error value retained.
6) The parameter vector found in the previous step was then used with the test data set in order to obtain the test error value.

Tables I and II contains the best set of parameters found using this procedure for the batteries and the motors, respectively (the term 'best' is used here to avoid the implication of optimality, since no guarantee of even near-optimality can be provided). Note that a different set of parameters was found for each battery, but only one for both of the motors. This is because currently each battery is instrumented with its own temperature sensor, while only one temperature sensor is available for the two motors (mounted on the support bracket between them). In order to obtain the thermal coefficients for the motors, the input current to them was averaged. The previously used parameter values are also provided for comparison (the previous set of model parameters was obtained through extrapolation from similar components described in the literature and/or by trial-and-error).

Figure 3 shows the output of the model (predicted terminal voltage) vs. the actual terminal voltage measurements for one

of the test scenarios and Figure 4 shows predicted vs. actual battery temperature. As this example illustrates, the new model is capable of capturing the general behavior of the system fairly well, although its accuracy still needs some improvement (towards the end of missions errors tend to accumulate to substantial values).

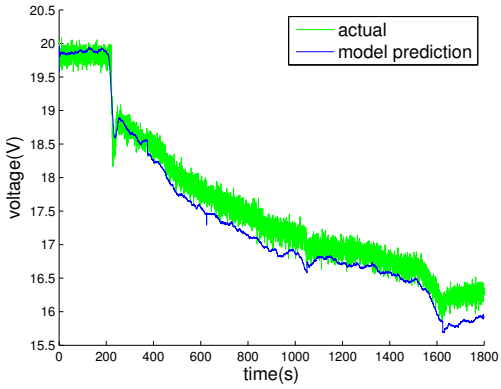

Figure 3. Actual vs. predicted voltage, URA40 (Flt 23)

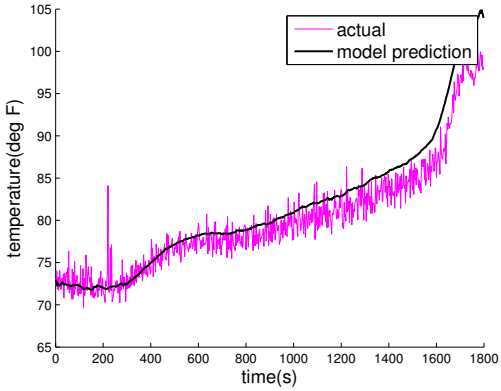

Figure 4. Actual vs. predicted temperature, URA40 (Flt 23)

Table II. Best values for the thermal parameters

	$C_t(\times 10^{-4})$	h
LLF20	20.76	11.68
ULA20	19.52	11.94
LRF40	21.12	11.38
URA40	18.27	10.41
battery previous	15.00	12.00
motor	10.32	11.17
motor previous	12.00	10.00

The overall mean test errors were $0.84V(0.90V)$ for battery terminal voltage (across all four batteries), $3.3°F(3.9°F)$ for battery temperature (also across all four batteries), and $4.7°F(5.2°F)$ for the motor temperatures. The numbers inside the parenthesis refer to the values obtained with the previous set of model parameters.

V. Tests with a Decision-Making Algorithm

A stochastic decision-making algorithm, based on particle filtering (and prototyped previously) was matured during this project and tested with the new model. The tests were intended to serve as more of a 'sanity' check rather than a comprehensive validation of the new model's performance (a more thorough evaluation will be done in the near future). Still, the tests conducted so far appear to suggest that the new parameters result in same or better performance compared to those used previously.

In the experiments a set W of ten sequentially numbered waypoints was first created, with each waypoint associated with a specific reward value. Test scenarios with progressively increasing numbers of waypoints were then defined (the 7-waypoint scenario contains waypoints 1 through 7, and so on). As the UAV transitions between waypoints 2 and 3 (in the original, unoptimized order), a fault is injected into one of the motors. The motor loses power, however its rotor can still spin. The fault leads to greater loads on the remaining motor, resulting in increased current draw and heat accumulation.

The waypoints are selected in such a way as to make it impossible for the UAV to visit all of them in the original order before either energy depletion or vehicle health deterioration beyond the point of failure. DM is then expected to rearrange and/or reduce the set of waypoints to maximize the mission payoff (i.e. find an optimal path), while remaining within the constraints on energy consumption and health degradation.

The particle filtering (PF) algorithm (Algorithm 1) is initialized with a set of k particles, each particle p_i containing the starting waypoint wp_1 and assigned the weight of $w_i = 1/k$. During each of the iterations of the algorithm (and for each particle), the path associated with a particle is sampled randomly out of the set of unvisited waypoints up to the maximum length of N. Each sample is tested in the simulator and the particle weight is updated proportionally to the objective function value. Unless system failure is believed to be likely for even the shortest path extensions, the particle path is extended by one waypoint (the first one in the randomized remaining waypoints set τ). The highest weight particle is identified and stored after each iteration, to enable interruptibility. Particle weights are then normalized and the particles are resampled. When the algorithm terminates, the particle with the highest weight is retained as the solution.

Table III provides the results of experiments with $k = 30$ particles. The columns contain the following information: (1) the number of waypoints in a scenario; (2) the benchmark reward values found by a deterministic algorithm (described in (Balaban & Alonso, 2013)); (3) the mean and the standard deviation of reward obtained over 30 trials with the previous set of model parameters; (4) the mean and the standard deviation of reward obtained over 30 trials with the new set of model parameters; (5) percent change in mean and standard deviaton with new vs. previous parameters. While the improvement in the last column may appear slight, it does indicate that in some of the trails, when faced with a close call between two potential paths, the algorithm was able to correctly select one with a higher reward based on more accurate model predictions.

VI. Conclusions and Future Work

The main objective of this project was finding methods for automated learning of model parameters for an aerial vehicle.

Table I. Best values for the battery electrical parameters

	$R_s(\times 10^{-3})$	$R_p(\times 10^{-4})$	$R_{cp0}(\times 10^{-2})$	$R_{cp1}(\times 10^{-16})$	R_{cp2}	C_s	C_b	C_{cp}
LLF20	7.4	0.9	3.0	1.2	34.41	112.3	18.3	316.1
ULA20	7.3	1.0	2.9	1.1	34.42	111.2	17.9	317.7
LRF40	7.3	1.1	3.1	1.1	34.42	112.8	18.0	317.2
URA40	7.1	1.1	3.4	1.0	34.46	113.6	18.2	316.9
previous	8.0	1.0	2.0	1.1	36.00	115.0	20.0	315.0

Algorithm 1 PF

inputs: $\{wp_i\}_{i=1}^{N}, K$
outputs: p^*
$p_1, \ldots, p_K \leftarrow \{wp_1\}$
$w_1, \ldots, w_K \leftarrow 1/k$
for $d \leftarrow 1, D$ **do**
  for $k \leftarrow 1, K$ **do**
    $\tau \leftarrow$ *permute*$(\{wp_i\}_{i=1}^{N} - p_k)$
    $l \leftarrow -1$
    repeat
      $l \leftarrow l + 1$
      $p_{test} = \{p_k, \{wp_1, \ldots, wp_l\}\}$
      $\{b, \mathcal{R}, \mathcal{C}_h, \mathcal{C}_p\} \leftarrow$ *simulate*(p_{test})
      $w_k \leftarrow \Theta^T \cdot \{\mathcal{R}, -\mathcal{C}_h, -\mathcal{C}_p\}$,
    until $\mathcal{F}(b) = true$
    if $l \geq 1$ **then**
      $p_k \leftarrow \{p_k, \{wp_1\}_\tau\}$
    end if
  end for
  $j \leftarrow \arg\max_j w_j$
  $p^* \leftarrow p_j$
  $\{w_1, ..., w_K\} \leftarrow \{w_1, ..., w_K\} / \sum_{i=1}^{K} w_i$
  $\{p_1, ..., p_K\} \leftarrow resample(\{p_1, ..., p_K\}, \{w_1, ..., w_K\})$
end for

Table III. Results of the simulated test scenarios

scenario	$\mathcal{R}$	$\mu_{\mathcal{R}}(\sigma_{\mathcal{R}})$	$\mu^*_{\mathcal{R}}(\sigma^*_{\mathcal{R}})$	*change*
7 *wpts*	236	234.0(1.1)	234.9(1.0)	0.3%(10%)
8 *wpts*	298	293.8(5.7)	295.3(4.8)	0.5%(18%)
9 *wpts*	383	375.2(10.2)	378.3(10.0)	0.8%(02%)
10 *wpts*	413	399.3(15.6)	402.4(13.5)	0.8%(15%)

This goal was, to a large extent, accomplished, although some of the parameters (related to mechanical efficiency and thermal coefficients for individual motors) were not derived due to the limitations of the currently available data sets. One of the outcomes of the project, however, was a set of ideas on how to improve the sensor suite on the test UAV before the next series of flight tests, so that these gaps can be addressed.

Due to the objective functions being based on non-linear equations describing physical processes, linear regression methods could not be used. Instead two general regression algorithms were explored, Levenberg-Marquardt and Simulated Annealing. With each having some strengths and weaknesses, a learning procedure was developed which used Simulated Annealing to zero in on a promising search space region and Levenberg-Marquardt to find the local optimum in that region.

The secondary objective was to test the best set of parameters with a decision-making algorithm based on particle filtering. Preliminary results suggest that the new set of parameters performs as well or, perhaps, even better as the previously used set (extrapolated from values reported in the related literature and through trial-and-error).

Future work will include the previously mentioned improvements to data collection and further refinements to the model parameter learning process, hopefully resulting in the ability to robustly estimate the current level of wear/degradation in a component. On the decision-making side, a better incorporation of uncertainty estimates into state updates and policy selection is planned.

ACKNOWLEDGEMENT

The author would like to extend his gratitude to Prof. Juan J. Alonso of the Department of Aeronautics and Astronautics (Stanford University) and to his colleagues at NASA Ames Research Center for their advice on this work.

REFERENCES

Balaban, E., & Alonso, J. J. (2013). A Modeling Framework for Prognostic Decision Making and its Application to UAV Mission Planning. In *Annual Conference of the Prognostics and Health Management Society* (pp. 1–12). New Orleans, LA.

Bertsimas, D., & Tsitsiklis, J. (1993). Simulated Annealing. *Statistical Science*, *8*(1), 10–15.

Chen, Y., Song, L., & Evans, J. W. (1996). Modeling Studies On Battery Thermal Behaviour, Thermal Runaway, Thermal Management, and Energy Efficiency. In *Proceedings of the 31st Intersociety Energy Conversion Engineering Conference (IECEC).*

Daigle, M. J., Saxena, A., & Goebel, K. (2012). An Efficient Deterministic Approach to Model-based Prediction Uncertainty Estimation. *Annual Conference of the Prognostics and Health Management Society*.

Hogge, E., Quach, C., Vazquez, S., & Hill, B. (2011). *A Data System for a Rapid Evaluation Class of Subscale Aerial Vehicle, NASA/TM2011-217145* (Tech. Rep.).

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441.

MATLAB version 7.11.0.584 (2010b). (2010). Natick, Massachusetts: The MathWorks Inc.

Milanfar, P., & Lang, J. (1996). Monitoring the Thermal Condition of Permanent-Magnet Synchronous Motors. *IEEE Transactions On Aerospace And Electronic Systems*, *32*(4), 1421–1429.