

Developing Health Management Strategies Using Power Constrained Hardware

Kevin M. Farinholt¹, Ali Chaudhry², Mark Kim³, Ethan Thompson⁴, Nathan Hipwell⁵, Ryan Meekins⁶, Stephen Adams⁷, Peter Beling⁸, Sherwood Polter⁹

^{1,2,3,4,5,6}*Luna Innovations, Inc., 706 Forest Street, Suite A, Charlottesville, VA, 22903, United States*

farinholtk@lunainc.com, chaudhrya@lunainc.com, kimm@lunainc.com, thompsons@lunainc.com, hipwellh@lunainc.com

^{7,8}*Systems and Information Engineering Department, University of Virginia, Charlottesville, VA 22903, United States*

rmm6ey@virginia.edu, sca2c@eservices.virginia.edu, pb3a@virginia.edu

⁹*Naval Surface Warfare Center, Philadelphia Division, Philadelphia, PA 19112, United States*

ABSTRACT

Condition based maintenance with prognostics (CBM+) is an area of research that interests many in the industrial, energy, and defense sectors. Interest in this concept is focused on lowering overall cost of operations, while also increasing equipment availability and mission readiness. Many applications, however, include power constraints and extensive lifecycle requirements that pose a challenge for existing embedded sensing systems. In some cases these systems can be expected to operate for years to decades without access to wired electricity or reliable energy harvesting sources. In this study a battery powered sensor node is presented that collects operational (pressure, acceleration, position) and environmental (temperature) information to identify and track faults seeded into an instrumented hydraulic test stand. The experimental setup is described in this paper, along with the range of baseline, damage cases, and severities imposed upon the system. Machine learning algorithms are developed specifically to leverage features that can be processed at the sensor node, then applied using low-power, computationally-limited microcontrollers. Several classifiers are considered in this analysis, including random forest and classification trees. The results discussed include prediction accuracies, training and testing requirements, as well as physical power consumption measured using actual hardware. Findings indicate that small sized random forest algorithms (up to 5 trees) can be implemented at the node and provide lower

error rates; however they operate with the higher computing times and power requirements when compared to other machine learning techniques. Conversely, classification trees provide a good trade-off in accuracy and computing time, prolonging the operational life of the sensor node given a finite capacity battery as the power source.

1. INTRODUCTION

Prognostics and health management (PHM) is an engineering discipline that uses health information about a system to support maintenance and operational decisions (Kalgren, Byington, & Roemer, 2006), (Uckun, Goebel, & Lucas, 2008). Embedded systems are often used to collect and process data with the intention of estimating the current health state of the system and predicting the future health state. PHM is being incorporated into numerous industries including aerospace (Batzel & Swanson 2009), automotive (Holland, Barajas, Salman, & Zhang, 2010), and manufacturing (Vogl, Weiss, & Helu, 2016). These industries all have large-scale systems-of-systems operations, and monitoring can be required for hundreds or thousands of components. The embedded monitoring system for each component could require multiple sensors and processors. As the number of monitored components grows, the energy efficiency of PHM systems becomes a paramount concern. In this paper, we outline our approach to creating energy efficient stand-alone CBM nodes to enable the use of PHM principles. The systems described are designed to run for several years on a limited power supply such as a battery. This is a goal because there are numerous applications where health monitoring hardware must operate in constrained, isolated, or harsh environments where external power is inaccessible. In other applications,

Kevin Farinholt et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

power may be nearby, but installation constraints introduce constraints that make it unfeasible to connect the monitoring systems to line-powered supplies. For example, one of our applications is monitoring hydraulic actuators on naval vessels. Ships may rely on hundreds of actuators to perform normal operating tasks and each could require several sensors for collecting data. While these ships have an abundant power supply, there are space and access concerns in some locations that limit wiring access to the hardware.

From research across several projects, we have identified three specific areas in the system design process where the tradeoff between performance and energy consumption should be evaluated: (1) data collection; (2) data processing and feature extraction; and (3) model training and implementation. We characterize the overall problem of constructing energy efficient PHM systems as a combined objective function where we minimize energy consumption and maximize performance. This multi-objective optimization problem often does not have a single solution but a Pareto optimal frontier on which any solution is feasible given stakeholder preference of energy consumption or performance. We consider both hardware-specific and software-specific solutions to this problem.

2. OVERVIEW OF METHOD

In each of our research projects targeting energy efficient machine learning techniques, the goal is to solve a multi-objective optimization problem that minimizes energy consumption E , while maximizing performance P . Mathematically, we can write this problem as,

$$\max(P, -E). \quad (1)$$

Constraints can be added to this optimization problem if necessary. We define three areas where energy consumption must be considered for a distributed system using a machine learning algorithm.

Data collection encompasses all of the energy consumed while acquiring the data set used to make a prediction. Let E_1 represent this quantity. There are numerous aspects of data collection that can affect energy consumption and overall performance of the system. Sensor selection is of primary concern as different sensor types can have wildly varying consumption costs. Signal conditioning and sampling rate also impact energy requirements throughout the collection process, particularly in applications where analog sensors are used. These three sources comprise the instances considered in this research; however future work could extend to channel resolution and other parameters for a more exhaustive investigation of methods to minimize energy consumption in the data collection process.

Data processing and feature extraction is the next area to consider. We represent the energy consumed during this stage as E_2 . This area includes all functions that must be executed to format the data and extract features used as

inputs into a machine learning algorithm. The processing part could include removing outliers and aligning data from various sensors. Feature extraction is the process of deriving values from collected data that are useful for training a machine learning algorithm or prediction. A feature can be extracted from a single data stream, or from multiple data streams. Further, features can be extracted from a single point in time or a time series of data. Often, when evaluating this stage of the machine learning process, we also evaluate the storage cost of features and data. This includes *what* information we choose to store, and *how* to store the information for easy, energy-efficient access.

Often, we combine the evaluation of these first two areas into a single process, and frame this task as a feature selection with cost problem. Feature selection is the process of selecting a relevant subset of features from the larger group of collected features (Dash & Liu, 1997), (Guyon & Elisseeff, 2003). Generally, feature selection methods only select features based on performance or intrinsic traits of the data, and they basically assume that each feature has the same cost. Cost-based feature selection methods can account for features with varying costs and weigh the tradeoffs between the cost of each feature and its performance benefit (Bolon-Canedo, Porto-Diaz, Sanchez-Marono, & Alfonso-Betanzos, 2014), (Iswandy & Koenig, 2006). We recommend utilizing a cost-based feature selection method to help make decisions about data collection, data processing, and feature extraction.

The energy consumption can be directly used as the cost in these methods. When performing sensor selection, features from the same sensor can be linked and treated as grouped features, i.e. if you want a feature from a sensor you can get the rest of the features from that sensor for no cost or at a reduced cost.

The final area we consider is the machine learning algorithm itself. Different algorithms can consume varying amounts of energy, and we represent the energy consumption of the algorithm as E_3 . Simple algorithms, such as linear regression or classification trees, often have very simple processes for calculating a prediction while more complex algorithms, such as deep neural networks or ensemble methods, can consume significantly more energy. There are two aspects of the algorithm that need to be considered: training and testing. The training energy consumption is the amount of energy required to learn a model from the data. The testing energy consumption is the amount of energy required to make a prediction on provided data. In most cases, the testing energy consumption is the primary concern because a model has been pre-trained before being deployed on the distributed hardware. However, in some cases, once data has been collected using the distributed hardware, the model needs to be updated. The model can either be retrained from scratch, or updated using online learning (Smale & Yao, 2006).

Each of the three areas described above can be evaluated separately, but in most cases the total amount of energy consumed by the system is evaluated. The total energy consumed by the system can be written as

$$E = E_1 + E_2 + E_3. \quad (2)$$

There is a tradeoff between performance and total energy consumption and each area of energy consumption can have a different impact on the performance. If there is an energy budget E^* , or the maximum amount of energy that can be consumed by the system, the optimization problem can be rewritten as

$$\begin{aligned} & \max P, \\ & \text{s.t. } E < E^*. \end{aligned} \quad (3)$$

In this problem, the energy consumed in one area will take away from the energy that can be consumed by another area, i.e. selecting an expensive or high energy consumption sensor could limit the types of feature extraction methods or algorithms that could be used.

3. BACKGROUND AND PRIOR WORK

The focus of this research is on designing power constrained PHM systems for a variety of use cases, including: mechanical and hydraulic equipment, composite structures, as well as shipping and transportation applications. In each of these cases, the goal is to produce an energy efficient, rugged, and robust electronics system and reliable software to diagnose a system's health and use this information to inform operation and maintenance practices through efficient PHM solutions. This includes optimizing the embedded system's sensors, sensor locations, data collection methods, features extracted from data, machine learning algorithms, storage and processing hardware, and power management strategies. Many of the design strategies have been application specific; however a portion of this work has led to more generic research in feature selection methods that encompass aspects of cost in a manner that can be extended across a wide variety of applications.

The results presented in this paper focus on PHM strategies for monitoring industrial hydraulic rotary actuators, and evolve from previous studies on the effects of the machine learning algorithms on the training and testing power consumption and the error rate in Adams, Beling, Farinholt, Brown, Polter, and Dong (2016). In this prior work, computation time was used as a surrogate for power consumption, an assumption that has been verified through experimental data using physical hardware in Sections 5.2 and 5.3 of the present research study. Based upon our findings in Adams et al. (2016), we concluded that the random forest (RF) algorithm was always on the Pareto frontier with the lowest error rate and the highest power consumption of the eight machine learning algorithms considered in that study. Similarly, the classification tree

algorithm was always on the Pareto frontier with a tradeoff between error rate and power consumption (Meekins, 2018), while the k-nearest neighbor (KNN) algorithm was not always on the Pareto frontier. Further, our research team has evaluated various feature selection and feature extraction techniques on optimizing power consumption and total computation time in Adams, Meekins, Beling, Farinholt, Brown, Polter, and Dong (2017). We are currently working on automated solutions for selecting classification algorithms under resource constrained environments.

Embedding cost into the feature selection process has been a primary focus of our research as this is expected to be one of the most straightforward ways to limit energy consumption. We have developed a feature selection method for hidden Markov models where the cost of each feature can be conveyed to the algorithm through the use of informative prior distributions in Adams, Beling, and Cogill (2016). We have also conducted a study using the general cost-based feature selection framework presented in Bolon-Canedo et al. (2014). In this investigation we confirmed the results of the original study and expanded on the numerical experiments in Adams, Meekins, and Beling (2017). One limitation of the general cost-based feature selection framework is that it is limited to filtering feature selection techniques. We are currently performing numerical experiments on wrapper feature selection methods and embedding the cost of features into training algorithms.

4. EXPERIMENTAL HARDWARE / MEASUREMENTS

An extensive amount of data has been collected for this application using a multichannel embedded sensor node that is configured to monitor pressure, temperature, angular position, and acceleration for an industrial hydraulic rotary actuator, such as the one shown in Figure 1. The system relies on commercial transducers (detailed in Table 1) to instrument the test stand and monitor both process-related and dynamic responses using the low power data acquisition hardware developed by Luna Innovations, Inc. (Figure 2).

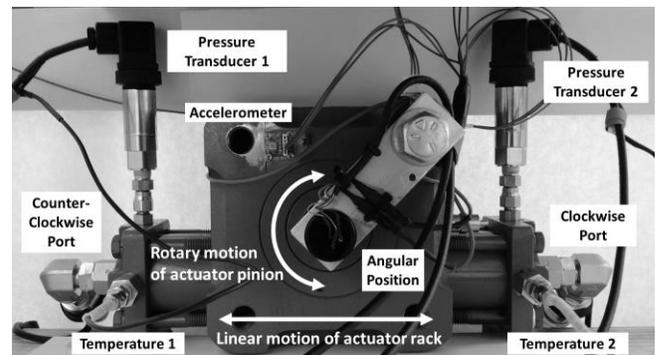


Figure 1. Hydraulic actuator testbed used in experiments

Table 1. Sensor Array

Function	Model / Type	Qty
Pressure	Omega PX319-3KGV	2
Temperature	Minco S202PD	2
Angular Position	Vishay 132B20103	1
Acceleration	ADXL1002	3
Trigger	APC 25.4 x 0.5mm PZT	2

The sensor node is built around Texas Instruments' MSP432 microcontroller and provides onboard signal conditioning, data storage, signal processing, analysis, and data transmission through wired or wireless interfaces. The sensor node can be powered from a variety of sources, including: 1) internal primary or rechargeable battery, 2) external DC power supply, or 3) external multi-source energy harvesting system; with acceptable supply voltages between 3.3V and 5.5V DC. Current and power consumption studies were performed using a constant 4.0V supply, with current levels being monitored using National Instruments' PXI-4130 Source Measure Unit. Power is calculated as the product of the 4.0V supply voltage and the measured current.

When used to monitor the hydraulic test stand, the CBM node records data from each of the sensors installed on the rotary actuator. To examine the system under controlled baseline and damage states, the *primary* actuator shown in Figure 1 is paired with a matching *load* actuator of the same class (Moog Flo-Tork 15,000 in-lbf) through a mechanical coupling designed to transfer full torque loads without mechanical distortion. A layout for the hydraulic actuator test stand is shown in Figure 3. A 5 horsepower pump supplies up to 3000 psi of pressure at 2 gallons per minute flow rate to produce an actuation stroke time of 1 second (0-90° rotation). Temperature and pressure sensors are located at each inlet to the actuator to monitor conditions in the clockwise (CW) and counter-clockwise (CCW) direction. A rotary position sensor tracks the angle of the actuator throughout its actuation stroke, while accelerometers monitor the x-, y- and z- axis vibration response. An opposing torque from the *load* actuator is produced by the backpressure generated when forcing fluid through a flow restriction, and a friction brake can be applied to mimic binding of the mechanical coupling using a manual hand pump and brass brake shoes.



Figure 2. Sensor node developed for embedded PHM.

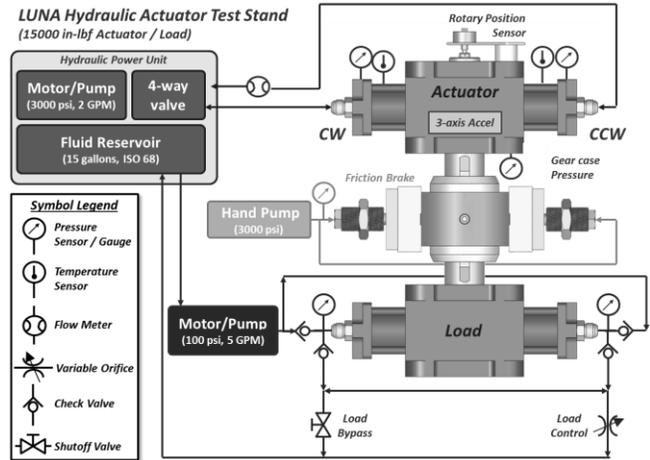


Figure 3. Actuator test stand hydraulics layout, including instrumentation locations (Adams et al. (2016))

A range of baseline conditions are collected using this test stand, accounting for minor variations in supply pressure, as well as low level opposing load conditions using the matched load actuator. A series of five damage cases have been imposed upon the system to simulate different scenarios that may be common for rotary actuators in different industrial or military applications. These include 1) excessive loss in supply pressure, 2) excessive external opposing load, 3) friction brake loading similar to a binding event, 4) external leak in hydraulic fluid from actuator seals, and 5) internal leak of fluid into the gear casing. Each of these damage cases can be imposed at increasing levels of severity, providing a wide range of damage states for use in CBM studies. At the highest level, the data can be binned in a binary *fault* versus *no fault* scenario. At the next level this can be considered a 6-class damage detection problem, while at the lowest level, it can be framed as a 20-class severity determination problem using the test conditions collected thus far.

In each of these cases, the sensor node is capable of acquiring data in either streaming or triggered data acquisition modes. Under normal operation, the actuator is energized for a short duration of time as it rotates between open and closed positions. Streaming mode is often used to acquire data over repeated cycles in high cycle count studies, capturing the full rotation cycle of the actuator in both CW and CCW directions. Figure 4 provides a time history of the angular position and pressure response for a typical clockwise rotation captured in streaming mode.

The first event of the actuation cycle occurs when the hydraulic power unit energizes the system (1). Local pressure builds within the hydraulic line until the actuator rack begins to move (2). At this point the actuator shaft begins rotating and transitions from 0-90°, coming to rest at an internal mechanical stop at full rotation (3).

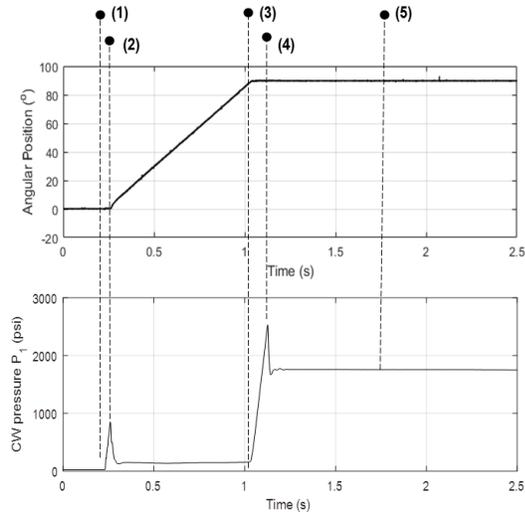


Figure 4. Angular position and pressure response measured for a normal clockwise rotation between 0° and 90° .

The system continues to be energized at this point causing inlet pressure to build until an overpressure relief valve is engaged within the HPU's fluid reservoir (4). Following this, line pressure is regulated to the relief valve's setting of 1,800 psi until the system is de-energized (5), awaiting actuation in the opposite direction.

In triggered acquisition mode, the response of the actuator remains the same; however the sensor node is kept in a low power sleep state until it receives an interrupt signal from piezoelectric transducers mounted to the actuator casing. This interrupt causes the microcontroller to direct power to the sensors before beginning to record. A latency period is observed during which the node does not capture the first 25ms of data (Figure 5). The system also contains a short transient period where sensors stabilize before beginning to acquire data, after which it converges to what is collected in streaming mode. For model training purposes, streaming data can be used to develop detection algorithms, however best results account for the latency and transient issues by excluding the first 100ms of data following the actuation signal. This allows the channel measurements to converge to one another, accounting for signal differences between streaming and triggered modes.

When monitoring power consumption for different data acquisition studies, the results correspond to data collected with the CBM sensor node in triggered acquisition mode. Data was acquired using the sensor node and then transmitted to a LabVIEW interface using RS-485 protocols for offline model training in Matlab. Once models were trained, they were then tested in both the Matlab environment on a laptop computer, as well as at the embedded sensor node using the MSP432 microcontroller. Data was acquired at two separate sampling rates, one at 1,000 Hz, and the other at 100Hz.

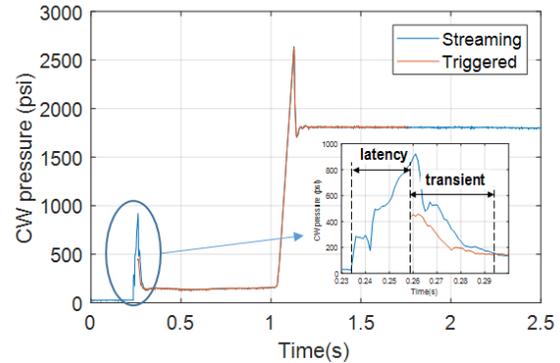


Figure 5. Comparison of the latency and transient issues observed between streaming and triggered acquisition.

5. METHODS FOR CONSERVING ENERGY AT THE NODE

In most applications involving hydraulic actuators, the equipment is only active for short durations of time while rotating between open and closed positions. In this scenario, the embedded sensors only need to monitor actuators while energized by the HPU. To conserve power in the timeframe between actuation cycles, the sensor node can be placed in a low-power sleep state where current consumption is limited to levels at, or below, $10\mu\text{A}$. Auxiliary components are powered off in this state, with only interrupt channels remaining open to conserve as much energy as possible. The following sections describe several techniques that have been used to characterize and reduce power consumption during the data collection, signal processing, and model implementation stages of the CBM process.

5.1. Data Collection

As the hydraulic actuator is energized, the sensor node receives a dynamic trigger from piezoelectric transducers bonded to the actuator casing. This signal initiates the data acquisition cycle, with a short surge in current as seen in Figure 6. This transient pulse in current is followed by a 2 second acquisition window during which data is acquired and the current consumed by the sensor node remains relatively constant. The first measurements are collected without any sensors physically attached to the sensor node electronics. In this arrangement, the microprocessor, ADC and signal conditioning circuits consume 21mA of current. When physically connected to the full sensor array, current demand increases to 25.5mA, accounting for a budget of 4.5mA to power the full array of sensors outlined in Table 1. Once the hardware acquires 2s of data, power is disconnected from the sensors and signal conditioning circuits, and the node transmits the entire dataset to a base station computer via RS-485 connection. In this configuration, the sensor node is used to collect data for model development and training, thus entire datasets are transmitted to the base station.

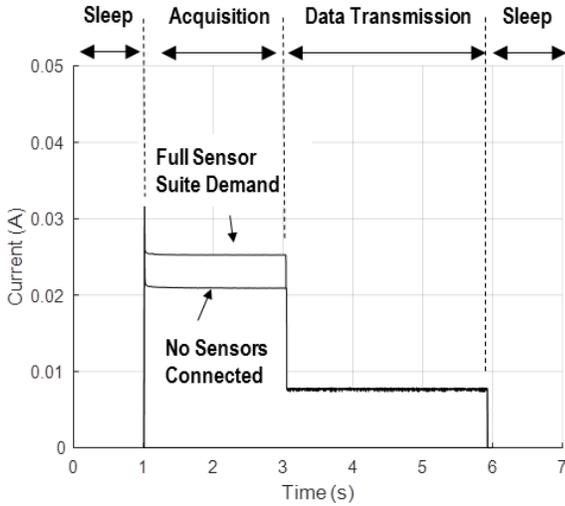


Figure 6. Current consumed by the PHM sensor node.

In the final application, models would be compiled and run directly at the sensor node, with only features or statuses transmitted, significantly reducing the time spent in transmission mode. When considering the use of wireless radios to send out data, minimizing the transmission time becomes important to conserve energy by the CBM node.

When considering the demand of specific sensor types, it is possible to quantify their consumption levels through selective connection with the sensor node. Figure 7 presents results specific to the inlet / outlet pressure transducers, which are seen to consume 2mA of current in normal operation. At lower data acquisition rates (those below 200Hz); the pressure transducer has a fast enough settling time to consider a modulated supply voltage strategy to power the sensor. In this case, the microcontroller strobos power to the pressure transducers in an attempt to reduce overall power consumption. In this strobed technique, the sensor is powered for 2ms out of every 10s (when configured to acquire data at 100Hz).

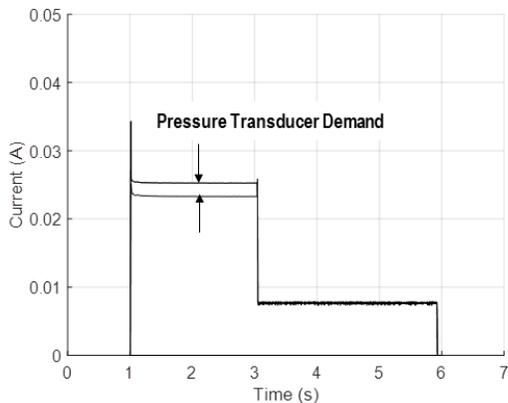


Figure 7. Current consumed by pressure transducers.

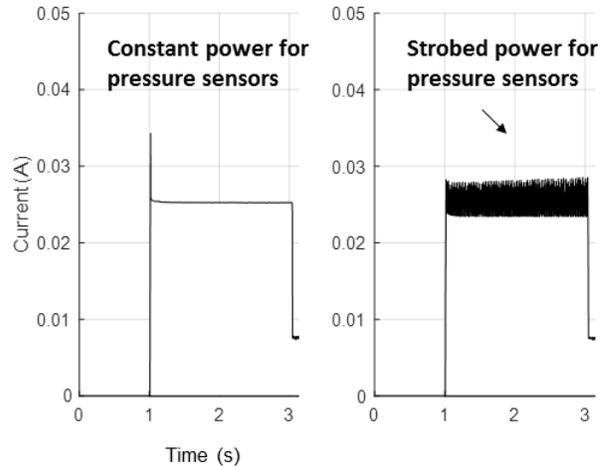


Figure 8. Current consumption using constant- and strobed-power approaches for pressure transducers.

This produces a 20% duty cycle for operation, and has been seen to reduce energy consumption by 2-5% during the data acquisition window (Figure 8). One detractor from this approach is that the strobed power technique produces transient spikes in current during the ON cycle for the transducer (Figure 8-right), exceeding the mean current demand when configured to run under a constant voltage approach.

Lower data acquisition rates could benefit more from this approach as fewer transient spikes would occur, however the break-even point for our hardware appears to occur near 100Hz. When operating above this frequency, steady-state current requirements for the constant voltage approach equal those of the transient profile generated in the strobed power mode of operation.

5.2. Data Processing

In addition to using sampling rate and strobed power strategies to conserve energy in the data collection stage, feature selection and the manner with which data is processed impacts the overall efficiency of the embedded sensing system. The first method for conserving power in this stage of the process is to minimize the number of features used in detection algorithms, and to focus on those that require the shortest computation time. For the CBM sensor node discussed in this paper, individual computations require 200 μ A of current per operation when running on the MSP432 at 48MHz clock speeds. A series of basic addition and multiplication operations were implemented on data collected by the sensor node in an effort to assign computation costs for signal processing operations, with results shown in Figure 9. These findings support previous assertions that computation time can be used as a reliable surrogate for power consumption when comparing feature extraction techniques and model efficiencies.

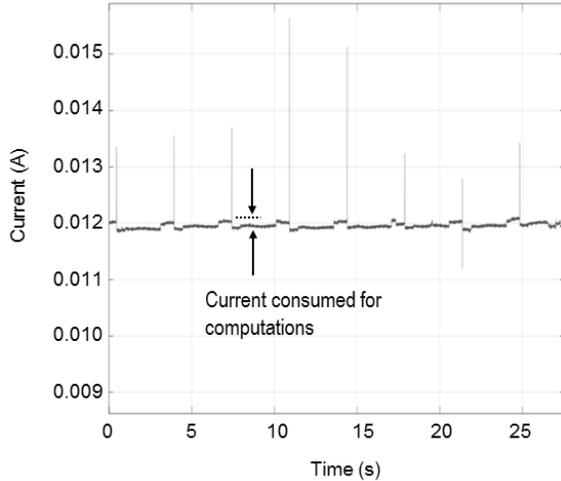


Figure 9. Current consumed for basic computations using the MSP432 microcontroller.

As this research has continued and more algorithms have been deployed at the sensor node, detailed power characterizations are validating the comparisons made in the earlier work by Adams et al. (2016).

As data is acquired, several methods have been examined to generate features such as mean, standard deviation, and other statistical metrics. The traditional approach that had been adopted at the outset of this program had been to 1) acquire data from the sensors, 2) post-process time histories to obtain features following the acquisition cycle, and 3) predict equipment status using algorithms deployed at the sensor node or base station. The current profile for this approach is shown in Figure 10, where feature calculation relates to the current consumed between 3.0 and 3.1s. In this example the mean, standard deviation, and variance were calculated for each of the sensor channels.

For lower acquisition rates, one alternate technique that can be used to conserve energy is to perform rolling calculations while the microcontroller is active but not being used between each individual measurement. While this technique is not possible for all features that may be of interest, it can be done for many statistical parameters such as mean and standard deviation. When applied using the CBM node in this study, there was no measurable increase in current consumption when features were calculated in this rolling approach throughout the data acquisition cycle. Thus, by interspersing feature calculations between measurements, it is possible to remove the 0.1s of additional time when post-processed following the acquisition window. Our work is currently compiling a list of features that are amenable to this process, and plans are to quantify the energy savings by interspersing calculations within the acquisition process as opposed to serially once all of the data has been collected.

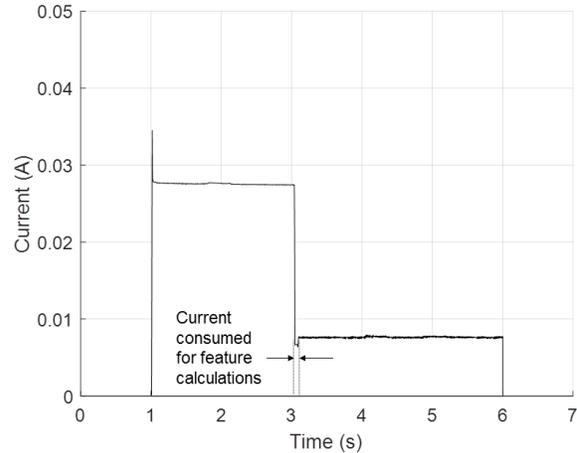


Figure 10. Current consumed to generate model features.

5.3. Model Implementation

When applying damage classification algorithms at the node, model complexity impacts energy consumption in much the same way that signal processing does. The results illustrated in Figure 9 indicate a current cost of 200 μ A per computation, supporting the assertion made in Adams et al. (2016) that computation time could be serve as a surrogate for power consumption in their preliminary analysis. Early development was performed using models developed and tested in Matlab, focusing on larger feature populations (up to 120 features) for the sensors summarized in Table 1. As work has transitioned these models to ones that can be deployed on the MSP432 microcontroller, the number of features has reduced to between 18 and 32 features. These have focused primarily on combinations of mean, standard deviation, variance, max / min, median, skewness, and kurtosis values. Preference is weighed between the feature's impact on model accuracy and complexity of computation required by the embedded microcontroller.

Models have also focused on three levels of classification for the damage prediction problem. The highest level model considers the *binary* classification scenario, in which the algorithm is simply tasked with differentiating *fault* conditions from the healthy, *no fault* test conditions. The second level model considers the *6-class* detection problem, where the algorithm assigns a damage type to the measurement data, in the form of baseline (Class: 0), low supply pressure (1), excessive opposing load (2), binding from friction brake (3), external fluid leak (4), and internal fluid leak into gear casing (5). The third and final level of model complexity provides an assessment of damage type and severity, with a total of *20-classes* possible given the test conditions imposed in the laboratory. A summary of the data collected, trained, and tested using Matlab and the MSP432 microcontroller is provided in Table 2.

Table 2. Summary of data labels for the binary, 6-class, and 20 class damage detection models

Description	#Obs.	Binary	6-class	20-class
Baseline	720	0	0	0
Low Pressure	240	1	1	1-2
Opposing Load	160	1	2	3-4
Friction Brake	480	1	3	5-11
External Leak	480	1	4	12-17
Gear Case Leak	240	1	5	18-19

In the binary detection problem classification trees provide near 100% prediction accuracies using 5-fold cross validation techniques. In this case, models are trained using 80% of the measurement data and then tested against the remaining 20% for validation purposes. Earlier work by Adams et al. (2016) found that the classification tree model required much less computation time (by a factor of 10) than any of the random forest techniques, while providing higher prediction accuracies than techniques such as KNN models.

Use of a hierarchical modeling approach is also being adopted in this research to simplify which algorithms are imposed early on in the life of a piece of equipment such as the hydraulic actuator. In most applications, the actuator is received and installed in a healthy state, and the primary role of the CBM node is to identify the onset of damage. For most damage cases we discussed with original equipment manufacturers, these systems are subject to damage mechanisms that evolve over time, rather than ones that present themselves as catastrophic failures that manifest within 1 or 2 actuation cycles. In this scenario, the CBM node can use a binary classifier to determine that a fault has arisen, after which more intensive 6-class or 20-class algorithms can provide more detailed assessments and input into recommendations for maintenance activities. With actuators that are designed for fifty thousand to one million cycles of operation, the use of simple, binary classifiers early on while the actuator is healthy will translate into significant, cumulative energy savings for the sensor node.

6. RESULTS

When tested in the laboratory, the primary limitation in model complexity has been driven by the random access memory (RAM) allocation available on the MSP432P401R microcontroller. Portions of this RAM need to be assigned to basic operation, acquisition, and data processing tasks, with the remaining kept available for model implementation. For Luna's current CBM sensor node, 64kB RAM / 256kB Flash are available, with the underlying firmware to control data acquisition (DAQ) operation consuming 40kB RAM. This leaves 24kB available for classification tree and some smaller random forest models.

Earlier work in Adams et al. (2016) focused on RF models containing at least 25 trees; however this was found to exceed the available memory when compiled for execution on the MSP432P401R chip.

To better address the limitations imposed by smaller RF models, a series of simulation studies were performed to investigate how model accuracy behaves as a function of model size given the 2240 observations summarized in Table 2. The study focused on a 20-class detection problem with 18 features drawn from the data. These features comprise the mean, standard deviation, and variance for a 2 second time windows of each pressure, temperature, and angular position channel, as well as a summation of the acceleration channels (3 statistical values calculated over 6 sets of data). These results are presented in Figure 11, and include results for a fine decision tree implementation (99.2%), as well as several RF models ranging from 1 tree to 50 trees in size. The RF approach begins to outperform the tree model for forests greater than 3 trees, and appear to converge to 99.8 – 99.9% accuracy with 15 trees.

In this study, models are trained in Matlab and compiled to c-code using Mathworks' Matlab Coder and Embedded Coder toolboxes. Once in this form, the c-code is reviewed in Texas Instruments' Code Composer software package to remove any portions of the code that are redundant or unnecessary for implementation on the CBM microcontroller. From this point the software is combined with Luna's base firmware developed for acquiring data using the MSP432. For initial comparison, the decision tree model was compiled, optimized, and ported to the sensor node. Using features calculated from the dataset described in Table 2, model predictions were generated and are summarized in the confusion matrix shown in Figure 12. Most of the misclassifications seen within this data are attributed to varying severities of the external leak case (conditions 12-17), resulting in an overall cross-validation accuracy of 99.2 % for the sample population.

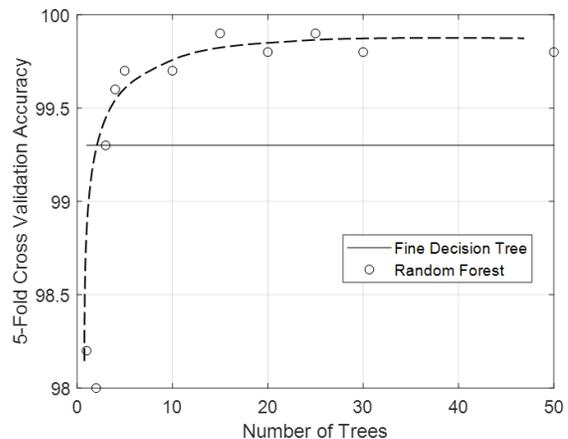


Figure 11. 5-fold cross validation accuracy as a function of the number of trees within random forest models

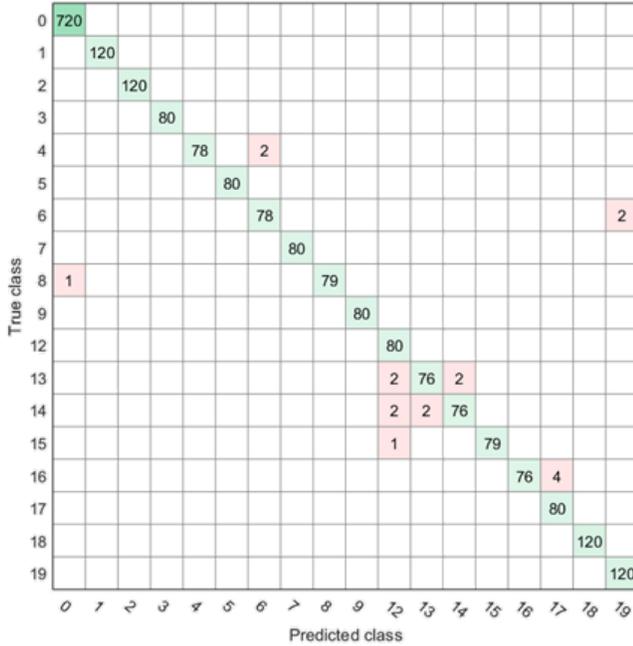


Figure 12. True v. predicted comparison generated for a classification tree model designed to run on the MSP432.

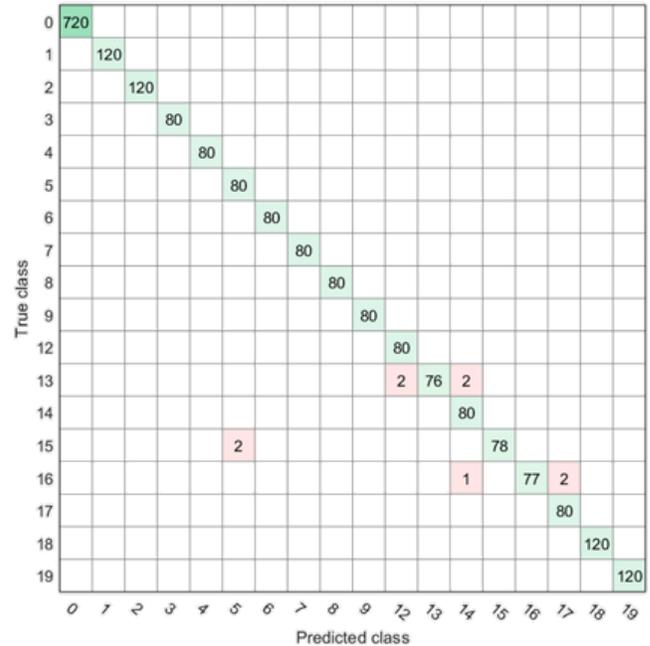


Figure 13. True v. predicted comparison generated for a 5 tree random forest model designed to run on the MSP432.

When extended to the random forest model, an iterative process was used to determine that a 5 tree model using 18 features could reliably run on the CBM sensor node. As predicted in the cross-validation study in Figure 11, the 5 tree RF model provided 99.7% accuracy in its prediction capabilities, with the results shown graphically in the confusion matrix of Figure 13. As in the previous case, most misclassifications for the RF model were observed to occur between severity levels within the external leak case.

The data used to generate these predictions were acquired at 1 kHz for each measurement channel. As mention before, lower sampling rates provide the opportunity to reduce power consumption, and consequently tests were performed to assess how data sampled at 100 Hz would impact prediction accuracy. For the most direct comparison, the dataset used to generate the predictions in Figure 12 and Figure 13 was down-sampled to a rate of 100 Hz. Models were then trained and tested with 5-fold cross validation accuracies of 99.2% calculated for a classification tree model, and 99.8% for the 5 tree RF model. It is interesting to note that there is a minor small increase in RF model accuracy using the 100 Hz data. More tests are planned to determine if this is a result of the natural variance in the 5 tree RF model itself, or if this can be attributed to a smoothing effect to the data at slower acquisition rates. Future studies are scheduled that use a new variant of the MSP432 that quadruples available random access memory. This will allow the team to test more intensive algorithms with higher prediction accuracies, in an attempt to validate the power to performance tradeoffs discussed earlier.

7. CONCLUSIONS

In this paper, we presented a summary of our prior work on applications of machine learning in embedded PHM systems where energy efficiency is primary concern. These embedded and distributed systems are designed to monitor physical equipment and structures for degradation and failure. These embedded systems are designed to run on a limited power supply for many years, and may be used in harsh or remote environments where access to an unlimited power supply is an unreasonable assumption. We describe the challenges we have faced in developing these systems and the approaches we use to reduce the energy consumption of the system.

Through these research projects, we have developed a method for characterizing and categorizing the power consumption of a system using machine learning algorithms. We outline three areas of the system that must be considered: (1) data collection; (2) data processing and feature extraction; and (3) model implementation. We frame this challenge as a multi-objective optimization problem where we wish to maximize performance and minimize energy consumption. If there is an energy budget, the multi-objective problem can be reduced to an optimization problem with constraints.

We discuss several methods for conserving energy in these power constrained applications. These involve collection of data at lower acquisition rates where the use of intermittent power to sensors can conserve energy. We also discuss the use hierarchical modeling techniques and algorithms that are

developed for implementation at the node, were energy storage and memory allocations dictate the complexity of models that can be compiled and run using the embedded microcontroller selected for application. Presently, the research team has identified a new variant microcontroller that increases memory allocations by a factor of 4, providing 256kB RAM / 2MB Flash. Hardware upgrades are underway and upcoming experiments will focus on how this impacts model complexity and energy utilization by the CBM sensor node. Reductions in sampling rate have also been shown to have no significant effect on prediction accuracy when dropping from 1 kHz to 100 Hz acquisition rates.

Additional testing is underway to determine how slowly data can be acquired before impacting accuracy. In this manner it may be more feasible to implement data collection techniques such as the strobed power method in efforts to reduce the energy costs associated with the data collection portion of the health management process.

ACKNOWLEDGEMENT

This material is based upon work supported by the Naval Sea Systems Command under Contract No N00024-16-C-4042 and N00024-17-C-4008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Sea Systems Command.

REFERENCES

- Adams, S., Beling, P., & Cogill, R. (2016). Feature selection for hidden Markov models and hidden semi-Markov models. *IEEE Access*, 4, 1642–1657.
- Adams, S., Beling, P., Farinholt, K., Brown, N., Polter, S., & Dong, Q. (2016). Condition based monitoring for a hydraulic actuator. *Annual Conference of the Prognostics and Health Management Society*.
- Adams, S., Meekins, R., & Beling, P. (2017). An empirical evaluation of techniques for feature selection with cost. *IEEE International Conference on Data Mining Workshops*, 834–841.
- Adams, S., Meekins, R., Beling, P., Farinholt, K., Brown, N., Polter, & S., Dong, Q. (2017). A Comparison of Feature Selection and Feature Extraction Techniques for Condition Monitoring of a Hydraulic Actuator. *Annual Conference of the Prognostics and Health Management Society*.
- Agresti, A. (2003). *Categorical Data Analysis*, Vol 482. John Wiley & Sons.
- Batzel, T., Swanson, D. (2009). Prognostic health management of aircraft power generators. *IEEE Trans. Aerospace Electron Systems* 45, 2.
- Bolón-Canedo, V., Porto-Díaz, I., Sánchez-Marroño, N., & Alonso-Betanzos, A. (2014). A framework for cost-based feature selection. *Pattern Recognition*, 47/7.
- Dash, M., Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1/3, 131–156.
- Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157–1182.
- Holland, S., Barajas, L., Salman, M., & Zhang, Y. (2010). PHM for automotive manufacturing & vehicle applications. *Prognostics & Health Management Conference*.
- Iswandy, K., Koenig, A. (2006). Feature selection with acquisition cost for optimizing sensor system design. *Advances in Radio Science: ARS*, 4, 135.
- Kalgren, P., Byington, C., & Roemer, M. (2006). Defining PHM, a lexical evolution of maintenance and logistics. *Autotestcon. IEEE*, 353–358.
- Meekins, R. (2018). *A multiple objective classifier selection methodology for real world problems* (Master's thesis). University of Virginia. Charlottesville. Virginia.
- Smale, S., Yuan Yao, Y., (2006). Online learning algorithms. *Foundations of Computational Mathematics* 6, 2, 145–170.
- Uckun, S., Goebel, K., & Lucas, P., (2008) Standardizing research methods for prognostics. *Prognostics and Health Management, International Conference on. IEEE*, 1–10.
- Vogl, G., Weiss, B., & Helu, M., (2016). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, 1–17.