Deployment of field robots in unstructured environments continues to rise, and the electrochemical battery remains the de facto standard for energy storage in robotic applications. However, robot mission planning, which relies on battery depletion time information, enforces conservative operation due to the lack of statistical rigor on the run-time predictions. A two tier self-supervised load characterization methodology for mobile robots operating in unstructured environments is proposed. Coupled with load characterization, a model-based statistical battery remaining run-time prediction algorithm utilizing particle filtering is presented. Given measured power loads during operation, the characterization algorithm employs Gaussian mixture modeling to cluster measured loads into a priori unknown regions. With clustered power demand regions, the computation of transition probabilities between the mixture models provides a jump Markov characterization of the historical power loads. An experimental study utilized Packbot data gathered during operation on general desert terrain. A particle filter prediction framework was shown to more accurately predict the remaining run-time of the Packbot given the unstructured terrain compared to existing load averaging techniques.

1 Introduction

Exploration of hostile unknown environments, such as interplanetary surfaces, abyssal/hadal ocean environments, hostile military reconnaissance, or irradiated areas, remains a task primarily suited for mobile robotics. As such, successful traversal and navigation of these environments requires reliable use of mobile energy sources. While some robotics systems utilize combustion engines (Kammel et al., 2008) or fuel cells (Wilcox et al., 2007), the electrochemical battery remains a key element of field robotics power systems (Laugier and Chatila, 2007). In many cases, mission success hinges on the successful return of a vehicle. However, there is a lack of reliable methods for remaining run-time prediction, especially for systems that operate under highly variable and stochastic conditions.

The deficiency of such reliable statistically based prognostic methods mandates conservative operation (Dorffel and Sharkh, 2006; Saha et al., 2011) by control algorithms or robot teleoperators (Lundberg et al., 2007). Conservative operation can result in mission time limitation, such as the preemptive charging of the Mars Exploration Rovers during a limited (24 hours) communication time frame (Brooks and Ingemnma,
2012) or anticipatory replacement of battery packs (Lundberg et al., 2007). Consequently, improvement of run-time prediction capabilities provides for increased operator confidence, allowing for extended mission time without explicitly upgrading hardware. Furthermore, with a statistically based framework, the operator or control system is provided with a level of confidence of remaining run-time and mission success. Given knowledge or estimates of remaining task time, vehicle velocities, and remaining mission distance, the remaining mission time statistics coupled with the remaining battery run-time statistics can provide a straightforward decision metric for early mission return. Other researchers (Kim and Shin, 2009; Truong et al., 2010) have begun to incorporate similar metrics into supervisory mission decision algorithms.

Prognostics of remaining run-time given local energy storage remains a tedious problem when environmental and task uncertainty are considered. Prediction requires characterization of both loading and energy storage performance. However, existing methodologies have failed to unite this source/load dichotomy and consider each only individually. Additionally, the run-time prediction must consider the system shutdown conditions, such as battery state-of-charge (SOC) depletion or voltage shutdown conditions of protective circuits. As a result of the increasing utilization of battery power in mobile applications and nonlinear dynamic complexities inherent to vehicle-terrain interactions, prediction of remaining run-time for battery systems sets the standard for performance of run-time prognostics algorithms (Saha et al., 2011). Scrutinizing the nuanced differences, existing battery run-time prediction algorithms in the literature are divisible into three overarching groups: static remaining run-time maps, model-based predictors, and data driven predictors.

Static run-time mapping algorithms typically rely on an empirical relationship known as Peukert’s law (Doerffel and Sharkh, 2006),

$$L = \frac{C}{\sum_{k=1}^{m} \alpha_k I_k^b},$$

where $I_k$ are constant current loads belonging to some finite set $I_k \in I_1, I_2, ..., I_m$ and $\alpha_k$ are the duty cycles of the respective current loads. Additionally, $L$ is the remaining life in hours, $C$ is the battery capacity in Ah, and $b > 1$ is some finite scaling coefficient. Fundamentally, Peukert’s law is a current counting technique with a polynomial modifier on the battery load current. Scaling the load current models the rate-capacity effect of battery systems which result from the internal ion diffusion process (Rakhmatov and Vrudhula, 2003). Contemporary adaptations (Kerasiotis et al., 2010) of the original empirical relationship expand the methodology to include sets of constant loads.

Additional reformulations of Peukert’s law tend to further refine and expand on the prediction capabilities by increasing empirical model fidelity. Some variants (Pesco et al., 1989; Matsushima et al., 1990; Wen et al., 2003) employ algebraic transformations, which remove the denominator polynomial term, and decompose the time discharge curve into an exponential region and a linear region. Resultingly, linear regression can then be employed for prediction of remaining run-time. Modified Peukert’s laws also accommodate further inputs through additional empirical relationships for prediction refinement, such as temperature (Ross and Budney, 1995), current duty cycles (Benini et al., 2002; Kang et al., 2008; Kerasiotis et al., 2010; Truong et al., 2010), battery age (Pop et al., 2009), and current SOC (Ross and Budney, 1995; Pop et al., 2009).

While the above techniques allow for direct computation of the remaining run-time via algebraic relationships, experimental studies (Doerffel and Sharkh, 2006) have demonstrated the limited applicability of Peukert’s law. For valid predictions, the battery must have lead-acid chemistry, operate at strictly constant temperatures, and have a constant current load (Doerffel and Sharkh, 2006). In unstructured environments, constant temperatures and currents cannot be ensured, as load demands on the actuators and sensors remain inherently stochastic (Ray et al., 2007). Additionally, modern mobile robotic vehicles rely on high specific energy chemistries, namely NiMH and Li-ion (Laugier and Chatila, 2007). As such, empirical methodologies are not likely to provide high fidelity predictions for field robotic vehicles.

Data-driven or soft computing algorithms, which potentially overcome the limitations of Peukert’s law, provide for alternatives to the static mapping techniques. For example, artificial neural networks (ANNs) allow for nonlinear mapping frameworks to be trained with high fidelity data (Akdemir et al., 2009). Once
trained and given temperature, future load profiles, initial SOC, and battery age, the ANN can predict mean remaining run-time with errors on the order of ±10.55% (Akdemir et al., 2009). However, in field applications, the entire load history is not known, nor can we ensure the ANN model was trained with high fidelity data. Other data driven prediction methods, for example nonlinear SOC regression (Chen et al., 2011) or autoregressive moving-average (ARMA) modeling (Lu et al., 2010), run into classic issues in the soft computing field, such as the bias versus variance trade off, long term forecasting inaccuracies, and limited transient load considerations (Hastie et al., 2001). As a result, extensions of the data-driven model predictions are not physically motivated (Chen et al., 2011).

Ignorance of transient/dynamic power requirements remains as a major flaw of the above techniques, since battery systems exhibit transient load dependent effects. Common phenomena include the rate-capacity effect and the recovery effect (Di Domenico et al., 2008). As such, prediction including transients requires a model basis which captures the dynamic effects of the battery system (Vachtsevanos et al., 2006). Fortunately, a wide range of battery models have been proposed in the literature: physically motivated circuit equivalent models (Liaw et al., 2005), discretized SOC empirical Markov models (Panigrahi et al., 2001), internal electron diffusion models (Rakhmatov and Vrudhula, 2001; Rakhmatov and Vrudhula, 2003), and electrochemical models (Di Domenico et al., 2008). Given a mathematical model of the battery, several techniques exist for prediction of future states and residual run-time: a priori extended Kalman prediction (Gelb, 1974), unscented transformations (Wan and Van Der Merwe, 2000; Julier et al., 2004), batch Monte Carlo (Metropolis and Ulam, 1949; Cloth et al., 2007), and sequential Monte Carlo (particle filtering) (Thrun, 2002; Doucet et al., 2001; Butler and Ringwood, 2010). Due to the ability to account for both parametric/load uncertainty and non-Gaussian state distributions, we adopt a sequential Monte Carlo approach, akin to (Goebel, 2011).

While accurate dynamic battery models and model-based prediction algorithms have been previously reported in the literature, prediction of residual run-time, given an uncertain environment, remains an open problem. Characterization of loads imposed on a system operating in uncertain environments is treated as a secondary concern to prediction for model-based remaining run-time. In (Saha et al., 2011), sliding horizon averages of the power loads are used for model forecasting. However, as previously discussed, these techniques ignore transients of the power loads (Di Domenico et al., 2008). Other researchers (Rakhmatov and Vrudhula, 2001; Rakhmatov and Vrudhula, 2003; Chen and Rincon-Mora, 2006) assume knowledge of future power demands.

Contrastingly, some have characterized loads stochastically, either as stationary Gaussian power loads or as Markov models, to account for uncertainty in the power demands (Panigrahi et al., 2001; Cloth et al., 2007; Jongerden and Haverkort, 2009; Jongerden, 2010). Only jump-Markov models introduce transients in power demand forecasts (Cloth et al., 2007; Jongerden and Haverkort, 2009; Jongerden, 2010), however load characteristics were constructed a priori without real time data. As illustrated in the above discussion on remaining run-time prognostic methodologies, existing techniques fail to capture the spectrum of power load uncertainty encountered in a field setting experienced by a mobile robotic system.

In this paper, we present a novel load characterization and prediction algorithm which overcomes the limitations of existing battery remaining run-time algorithms, and which provides for an apropos extension to field robotic systems. For online transient load characterization, an unsupervised clustering algorithm identifies regions of power demand over time and a jump-Markov chain is used to characterize the region transition characteristics. In the robotic framework, these power clusters are associated with system operation mode and terrain traversement. A model-based prediction methodology, the particle filter, is employed to predict the statistics of battery remaining run-time, using the above load characterization scheme. Contrastingly, a legacy technique, the extended Kalman filter used for prediction, serves as a comparison for the more computationally intensive particle filter. All of the above methods are demonstrated with data collected during a field experiment using a Packbot ground robot.

The subsequent discussion is divided into three main components which sequentially elucidate the methodology and its application to the Packbot vehicle platform. The general framework, which includes both
self-supervised load characterization and the residual run-time prediction methodology, is discussed in Section 2. The ensuing discussion details the theory of the load characterization scheme and particle filtering (PF) prediction techniques in Sections 3 and 5, respectively. For verification, the proposed methodology is applied to data gathered from a Packbot ground robot, which is discussed in Section 4. Using field data gathered during testing with the Packbot, two illustrative prediction case studies were conducted. In Section 6, prediction is initially demonstrated using the Packbot data assuming an ideal non-dynamic energy source. Contrastingly, the dynamic battery model, discussed in Section 7, is used in Section 8 to predict actual Packbot run-time. The results and future work are summarized in Section 9.

2 Remaining run-time prediction framework

Traversal of unstructured terrain by mobile robotic vehicles imposes a wide range of load profile types on the power system. In addition to general hotel (idling) power requirements, tele-operated/fully autonomous field robotics utilize an array of sensors and actuators all of which draw power from the local energy storage mechanism during operation. Commonly, field robotic sensors include a combination of cameras (visible spectrum and infrared), light detection and ranging (LIDAR), GPS, locomotion encoders, proprioceptive sensors, Hall effect sensors, temperature gages and accelerometers (Ray et al., 2007; Lundberg et al., 2007; Brooks and Iagnemma, 2012). Actuators typically required are locomotion drives, which are commonly electromechanical actuators, but also potentially hydraulic or pneumatic. Other common actuators include articulated manipulators (Lundberg et al., 2007). Depending on the current terrain and mission requirements, certain sensors or actuators, or a combination thereof, might activate via user command or autonomous action. In Figure 1, the power loads of a hypothetical robotic vehicle are shown.

![Figure 1: Illustrative robot power load distributions.](image)

Power demands during steady state or constant operation exhibit Gaussianity (Kim and Shin, 2009). As such, in a laboratory setting one could hypothetically characterize each of the probability density functions (PDF) of power loads, shown in Figure 1, for a given robotic vehicle. However, a priori characterization of expected loads potentially discounts unexpected terrain or vehicle operation. Consequently, a mobile robotic system should be capable of self-supervised characterization of power demands online. Essentially, the clusters of power loads shown in Figure 1 should be identified independently and strictly from the robot’s “experience”. In addition to clustering of power loads, the self-supervised characterization scheme must determine the likelihood of transitions between the identified load clusters. Figure 2 illustrates hypothetical transition characteristics between the power loads of Figure 1 as a Markov chain.

Given a characterization of the power demands imposed by an unstructured environment on the robotic system, model-based energy prediction becomes feasible. Load characterization, akin to Figure 1 and 2, allows for individual realizations of power demands. By generating many load forecast realizations or load “particles”, the particle filter framework can be employed for prediction. The generated particles are fed through a model for the battery system, which provides insight into the variance of the prediction. Given a statistical prediction of the remaining charge in the battery from the particle filter and the shutdown statistics of the battery protective circuitry, the remaining run-time prediction statistics can be found. The run-time prediction scheme, in its entirety, is detailed in Figure 3.
The overall self-supervised load characterization and run-time prediction scheme relies on a three tiered methodology. Firstly, the power load, given unstructured demands, are clustered into Gaussian distributions. Clustering, via the Gaussian mixture modeling (GMM) methodology, is discussed in Section 3.1. Once the individual load distributions are identified/clustered, the probability of transitioning between the distributions via jump-Markov chain theory is found. Jump-Markov chains are discussed in Section 3.2. Finally, particle filter prediction of remaining run-time, given characterized loads is discussed in Section 5.

### 3 Self-supervised unstructured load characterization

As a self-supervised characterization technique, the hybrid Gaussian mixture and jump Markov (GMJM) modeling algorithm must be able to characterize load demands online with field data. Each of these soft computing methodologies is discussed individually and then data characterization is illustrated, in Section 4, with Packbot data gathered in the field.
3.1 Load clustering via Gaussian mixture modeling

As discussed in Section 2, the load power demand can be viewed as a jump-Markov process alternating between an arbitrary number of discrete loads. Since the robotic system operates on unstructured terrain, the number of power loads demanded remains unknown a priori. As a result, the robotic system must characterize the load demands online given measured power requirements during mission operation.

The Gaussian mixture modeling methodology allows for the identification of clusters of power demands measured during field operations. Akin to Figure 1, loads are clustered into Gaussian distributions. Additionally, the arbitrary selection of Markov states is eliminated since the GMM procedure, via information criteria, can identify the number of Gaussian clusters. Thusly, the number of Markov states required for the GMJM process can be directly acquired (Hastie et al., 2001). Given a data vector \( X = (x_1, x_2, ..., x_n) \), such as the measured power load vector, the GMM procedure fits the overall measurements with a mixture of Gaussians of the form,

\[
P_\theta(x) = \sum_{k=1}^{M} \alpha_k \phi(x|\mu_k, \sigma_k^2),
\]

where \( M \) is the specified number of clusters, the model parameters to be found are \( \hat{\theta} = [\hat{\alpha}_k, \hat{\mu}_k, \hat{\sigma}_k^2] \), the distributions of \( \phi(x|\mu_k, \sigma_k^2) \) are normally distributed, and the Gaussian mixture is assumed to be univariate without loss of generality. The \( \alpha_k \) values are mixture weights of the Gaussian distributions which must satisfy the normalizing condition, \( \sum_{k=1}^{M} \alpha_k = 1 \). To procure the optimal parameters for a GMM, one must optimize the likelihood function, \( P_\theta(X|\theta) \), as follows,

\[
\hat{\theta}_{MLE} \in \arg\max_{\theta \in \Theta} P_\theta(X|\theta).
\]

The standard technique for computing the maximum likelihood parameters is the expectation maximization (EM) algorithm (Dempster et al., 1977), which iteratively calculates more likely parameters for the GMM until convergence. However, due to the lack of convexity, the algorithm does not guarantee convergence to the global optimum (Hastie et al., 2001). To heuristically overcome this limitation, the EM algorithm can be implemented with multiple reinitializations of initial conditions, \( \hat{\theta}_0 \). For each maximum likelihood optimization in this work, the EM algorithm is implemented five times with new initial conditions, and the resulting parameters with the overall maximum likelihood are selected as optimal. To initialize parameters for each EM algorithm, the initial conditions are randomized in accordance with (Hastie et al., 2001). Uniform random sampling of \( M \) values over the range of the data vector, \( X \), serve as the initial conditions for \( \mu_k \). The \( M \) mixture variances are identically initialized with the overall sample variance, \( \sum_{i=1}^{n} (x_i - \bar{x})^2 / n \), where \( \bar{x} \) is the mean of \( X \). Finally, the mixture density weights, \( \alpha_k \) are also identically initialized with equal weights, \( 1/M \). Convergence of each initialization of the algorithm is achieved with either a maximum number of iterations, or reaching a threshold in the step size of the likelihood function.

Determining the number of mixture models, \( M \), required for unsupervised clustering of the data vector remains a non-trivial computational problem which must be approached iteratively. One common heuristic metric for model order determination is the Akaike information criterion (AIC) (Akaike, 1974). The AIC enforces a trade off between model complexity and model fidelity,

\[
AIC = 2k - 2\ln(Q),
\]

where \( k \) is the total number of parameters in \( \theta \) and \( Q \) is the likelihood function which falls from the EM algorithm (Dempster et al., 1977). Essentially, the AIC punishes increasingly complex models while rewarding more accurate models. An iterative EM fitting process, which sequentially explores the possible number of clusters, attempts to minimize the AIC to determine the optimal cluster number. Through the AIC elimination procedure, the number of GMJM states is chosen directly, as opposed to the arbitrary selection of Markov states for load forecasting in other works (Lin et al., 2004; Johannesson et al., 2005; Opila et al., 2008). To alleviate the potential computational burden of a high number of clusters, the iterative EM fitting process is limited to fifteen clusters. Additionally, cluster separation drops significantly beyond fifteen clusters, meaning that the Gaussian distributions begin to overlap (Hastie et al., 2001).
For each element in the data vector, the posterior probability of belonging to each mixture model is given by,

$$P(C_m|x_i) = \frac{\hat{\alpha}_m \phi(x_i|\hat{\mu}_m, \hat{\sigma}^2_m)}{\sum_{m=1}^{M} \hat{\alpha}_m \phi(x_i|\hat{\mu}_m, \hat{\sigma}^2_m)},$$  \hfill (5)

where \(C_m\) are the individual mixture models respectively. As shown by equation (5), the GMM methodology is a soft clustering algorithm, meaning each element is not exclusively a member of each mixture set. As such, classification into a respective mixture requires a decision rule (Hastie et al., 2001). In this body of work, the classifier was chosen to select the most likely cluster for a given data point, or more generally known as maximum a posteriori (MAP) classification. Given the posterior probability in equation (5), the MAP for a given data vector value \(x_i\) is,

$$y_i = \arg \max_{m \in \{1, 2, \ldots, M\}} P(C_m|x_i),$$  \hfill (6)

where \(y_i\) are the classified load points. Classification translates the data vector, \(X\), into a classified data vector, \(Y = y_1, y_2, \ldots, y_n\). The classified load vector provides information on the movement of the load between the different identified clusters over time. A discrete process, such as \(Y\), can be readily modeled as a Markov jump process.

### 3.2 Markov jump process modeling

Modeling the stochastic loading process via a Markov jump process provides a methodology for capturing transient load demand behavior. Given a vector of classified power data from equation (6), the probability of “jumping” to another power load can be found; replicating Figure 2. For example, if the final power value, \(x_n\), was MAP classified in cluster \(y_n\), the jump Markov model provides the probability of \(y_{n+1}\) belonging to any of the \(C_m\) clusters. In other words, the conditional probability distribution for the next state relies solely on the previous value, or succinctly stated,

$$P(X_{n+1} = x|X_n = x_n) = P(X_{n+1} = x|X_n = x_n, \ldots, X_1 = x_1),$$  \hfill (7)

where \(X_1, \ldots, X_n\) are from stochastic process, \(X(t)\), which is defined on the countable set \(S\) (the state space of the Markov chain). Equation (7) above defines the celebrated Markov property (Costa et al., 2005). Due to the physical nature of the load demands, and thus the causality of the system, loading over time can be stated to hold to the Markov property (Lin et al., 2004; Pourmousavi Kani and Riahy, 2008; Hahn et al., 2009). Additionally, loading processes generally remain time-independent, and thus stationarity of the Markov chain applies (Costa et al., 2005) or, more generally,

$$P(X_{n+1} = b|X_n = a) = P(X_n = b|X_{n-1} = a).$$  \hfill (8)

As a result of the stationarity assumption, a matrix of transition probabilities between the elements of set \(S\) fully describes the Markov chain model. The transition matrix describes, for each possible current state, the probability of switching to any other of the possible states in set \(S\) in the next discrete step (Costa et al., 2005). Expressed mathematically, the transition matrix takes the form,

$$T_{ij} = P(X_{n+1} = j|X_n = i),$$  \hfill (9)

where \(\sum_{j=1}^{N} T_{ij} = 1\) ensures unitary total probability of transition for each given state. Realization of the next Markov state given a current state requires two steps: computation of the conditional probabilities for the next step and realization via weighted random numbers. The conditional probabilities at the next step are given by,

$$X_{n+1} = X_n^T T_{ij}.$$  \hfill (10)

The iterative unsupervised GMJM algorithm is able to identify the number of clusters and the transition characteristics between clusters in approximately 0.87 ± 0.19 seconds on a 2.6 GHz processor with data sets
ranging from 1000 to 5000 data points. Warm starting, or using the previous fit for the initial conditions, improves convergence time for all additional GMJM fits and improves the overall algorithm time to 0.35±0.28 seconds on the same machine. Additionally, the EM algorithm, MAP classification, and the calculating the Markov transition matrix all require no complex mathematical operations beyond standard operations and thus can be implemented directly onto a robot microcontroller.

4 Characterization of field power demands

To illustrate the characterization/prediction methodology, the following section details an experiment conducted near Twentynine Palms, California during which the power loads of a Packbot were measured. A brief discussion of the system and the environment are presented followed by an instance of measured data characterization.

4.1 Packbot field experiment

A Packbot unmanned ground robotic tracked-vehicle was used for the experimental measurement of power loads on unstructured terrain. As a differentially-steered tracked vehicle with additional flipper tracks, the Packbot can traverse a spectrum of terrain types and resultingly allows for sampling of diverse unstructured power loads. The Packbot deployed for this study was equipped with a UBI-2590 Li-ion battery pack in parallel mode with a voltage rating of 14.8 V and a nominal charge of 12 A-h. The overall system, including sensor packages and the modified battery pack, weighed 17.7 kg. To measure the power demands imposed on the battery system, current/voltage sensors, which were installed on the main power bus, recorded loads at one kHz. The bus voltage is regulated by a DC-DC boost converter which maintains a load bus voltage of 26 V. Additionally, the position of the Packbot during the study was recorded via GPS every second. The objective of the experimental run was to drive the vehicle over varied terrain until the battery system was fully exhausted resulting in Packbot shutdown conditions. The Packbot configuration used in these tests and the measured battery performance should not be interpreted as nominal performance for this system.

![Figure 4: (a)-(c) Illustration of the diverse terrain encountered during the experimental run.](image)

To ensure a diverse sample of terrain types, generic desert terrain was selected for data collection near Twentynine Palms, California. In addition to loose sand, the terrain consisted of organic foliage, gravel and larger stone obstacles, shown in Figure 4. A predetermined mission course, specified by way points, was identified which guaranteed all terrain-types would be encountered during the experiment. To guide the vehicle through the course, a vehicle operator remotely controlled the Packbot via a handheld controller. Additionally, the operator maintained direct line-of-sight observation of the vehicle throughout the entire experiment (no tele-operation control).

The actual path traversed by the Packbot during the experimental run is shown in Figure 5. With a fully-
charged battery, the Packbot was initially driven uphill on a paved road. After traveling approximately 30 meters, the operator veered the Packbot into the desert over a stretch of varnished rock. During the rest of the experiment, the operator employed judicious driving. As a result, all encountered terrain obstacles, ranging from steep rock banks to foliage penetration, were overcome during the first attempt. Ultimately, the Packbot traversed 762 meters in 55.15 minutes on a single charge. System shutdown resulted from large transient power demands with a peak battery current of 19.65 A, due to maneuvering through heavy brush.

![Normalized GPS path data of the Packbot during the unstructured load measurement experiment.](image)

Figure 5: Normalized GPS path data of the Packbot during the unstructured load measurement experiment.

As a result of the unstructured terrain, the power demands on the battery exhibited high stochasticity, as seen in Figure 6. Current demands depended directly on the operator speed demand (ranging from 0 to 0.38 m/s), terrain type, and maneuver type (use of flipper, differential steering, etc.). Consequently, the statistics of the load current remain non-stationary, and as such, load averaging fails to capture the true variance of the actual demands. In contrast, the unstructured load characterization methodology, discussed in Section 3, does capture the non-stationarity exhibited by the measurements.

![Packbot bus current load demands during unstructured terrain traversement.](image)

Figure 6: Packbot bus current load demands during unstructured terrain traversement.

### 4.2 Jump-Markov characterization of loads

Forecasting future Packbot power loads for prediction requires characterization of the load current data given in Figure 6. For this example, the first fifteen minutes of power loading data are used for GMJM model characterization, while the remaining data is utilized for algorithm validation. Sequential execution of the EM algorithm for GMM fitting yields a series of candidate mixture models for the power data. The Akaike information criteria, through iterative fitting of GMMs, indicated eight clusters would provide the ideal characterization balancing between model variance and bias, shown in Figure 7.

Figure 8(a) illustrates the AIC selected mixture model distributions computed via the EM algorithm and
Figure 7: Akaike information criteria for GMM fit with optimal information in eight mixtures.

Table 1 provides the details on the mixtures. Additionally, the final column of Table 1 provides the mean power load for each mixture current, provided the 26 V constant bus voltage of the Packbot. As a physically motivated confirmation of the EM clustering algorithm, current distribution $m_7$ in Figure 8(a) identified the hotel loads of the Packbot. The GMM fitting algorithm models the hotel power loads as requiring 23.7 Watts as contrasted to the estimate of 25 Watts. Also note that the mixing proportion given in Table 1 indicates that the Packbot remains idle (requiring only hotel loads) for 15.4% of the first fifteen minutes.

Figure 8: (a) Mixture model for power data computed via the EM algorithm. (b) Transition probabilities between distributions given by the EM algorithm in (a).

After clustering the load data of Figure 6 into respective mixtures given in Figure 8(a), computing the transition probabilities between each mixture produces the Markov transition matrix. Although given one kHz power data, the sequential computational requirements of the particle filter mandate expanded time steps for online feasibility. Resultantly, the chosen jump-Markov time step explicitly specifies the particle filter step time. For particle filter computation times under 30 seconds, a Markov chain time step of one second was employed on extended prediction intervals of greater than 30 minutes.

Graphically, the Markov transition probabilities are illustrated in Figure 8(b). Each box in Figure 8(b) depicts the probability of transitioning from the prior value to the posterior value. The diagonal of the matrix gives the probability that the predicted distribution will remain the same.
Table 1: Gaussian mixture model parameters from EM algorithm of Packbot current loads and the equivalent mean power demand on the 26 V main bus.

<table>
<thead>
<tr>
<th>Mixture (k)</th>
<th>Mean ($\mu_k$) [A]</th>
<th>Standard Deviation ($\sigma_k$) [A]</th>
<th>Mixture Weight ($\alpha_k$)</th>
<th>Load Power [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.67</td>
<td>0.47</td>
<td>0.20</td>
<td>69.42</td>
</tr>
<tr>
<td>2</td>
<td>3.39</td>
<td>0.92</td>
<td>0.09</td>
<td>88.14</td>
</tr>
<tr>
<td>3</td>
<td>1.14</td>
<td>0.12</td>
<td>0.13</td>
<td>29.64</td>
</tr>
<tr>
<td>4</td>
<td>1.53</td>
<td>0.24</td>
<td>0.15</td>
<td>39.78</td>
</tr>
<tr>
<td>5</td>
<td>1.13</td>
<td>0.02</td>
<td>0.08</td>
<td>29.38</td>
</tr>
<tr>
<td>6</td>
<td>2.07</td>
<td>0.29</td>
<td>0.18</td>
<td>53.82</td>
</tr>
<tr>
<td>7 - Hotel</td>
<td>0.91</td>
<td>0.07</td>
<td>0.15</td>
<td>23.66</td>
</tr>
<tr>
<td>8</td>
<td>5.54</td>
<td>1.56</td>
<td>0.01</td>
<td>144.04</td>
</tr>
</tbody>
</table>

5 Particle filter prediction

The GMJM procedure discussed above provides for a characterization methodology of the non-stationary power loading exhibited by mobile robots operating on unstructured terrain. Resulting from the transient load effects included in the GMJM characterization, a model-based predictor utilizing a dynamic model is necessarily adopted. Additionally, the nonlinear dynamics of the electrochemical battery system, which is discussed in Section 7, necessitate a nonlinear dynamic predictor.

Techniques, such as linear variance propagation (extended Kalman filtering (EKF) without innovation) and unscented transformations (basis for the unscented Kalman filter (UKF) prediction step), provide nonlinear model-based prediction capabilities. Extended Kalman prediction, an analytical methodology, directly propagates the uncertainty of the system through local linearizations of a nonlinear system model. The dynamic propagation is commonly expressed as, $\dot{P} = FP + PF^T + Q$, where $P$ is the state covariance matrix which indicates the uncertainty associated with each model state, $F = \nabla_x f(x)$ which is the linearization of the nonlinear model and $Q$ is the variance of process noise (Gelb, 1974). The unscented transformation abandons the local linearization, $F$, for direct propagation of sigma points (or significant particles) through the nonlinear model. The number of sigma points necessary is given by $2L + 1$ where $L$ is the number of model states and inputs (Julier et al., 2004). As a result of the analytical nature of the EKF and the limited requirements of sigma points for the UKF, these techniques remain computationally efficient. However, neither the EKF nor the UKF provide for statistical considerations of jump-Markov inputs. Additionally, both methods assume Gaussian uncertainty, while the GMJM characterization methodology and nonlinear battery discharge characteristics are non-Gaussian processes (Candy, 2009).

Contrastingly, the particle filter, a sequential Monte Carlo technique, represents the uncertainty in a model as a discrete set of values known as particles. This discretization allows for representation of all model states and inputs as non-Gaussian functions. However, effective coverage of all states necessitates a large number of particles, which mandates a more substantial computational effort as compared to the EKF or UKF (Thrun, 2002). Nonetheless, the capabilities for jump-Markov load inputs (Doucet et al., 2001) and non-Gaussian probabilities suggest the particle filter over other model-based prediction methodologies. Furthermore, the particle filter has become universally adopted as the methodology for non-Gaussian uncertainty prediction in fields such as fault detection/prognostics (Vachtsevanos et al., 2006) and nonlinear estimation (Candy, 2009).
5.1 Sequential Monte Carlo

Prediction, given an initial probability density function, through a nonlinear system can be analytically computed via the Chapman-Kolmogorov equation (Candy, 2009),

\[ P(x_{t+1}|z_{1:p}) = \int_{-\infty}^{\infty} f(x_{t+1}|x_t)P(x_t|z_{1:p})dx_t, \]  

where \( f(x_{t+1}|x_t) \) is the nonlinear system defined as a Markov process, \( P(x_t|z_{1:p}) \) is the state PDF given all existing measurements, and \( P(x_{t+1}|z_{1:p}) \) is the prediction given all measurements, and \( p \) is the time at which prediction begins indicating no further measurements. However, equation (11) remains analytically intractable given nonlinearities and non-Gaussianity. As such, the particle filter framework adopts a discretized approach to equation (11)(Candy, 2009),

\[ P(x_{t+1}|z_{1:p}) \approx \sum_{i=1}^{N} f [\delta(x_t - x^*_i)], \]

where \( \delta(x_t - x^*_i) \) are discrete particle representations of \( P(x_t|z_{1:p}) \) and \( N \) is the number of particles. Euler-Maruyama discretization provides for conversion of continuous stochastic nonlinear dynamic models to discrete Markov chains to satisfy equation (12). The discretized probability functions, or “particles”, are then propagated through the nonlinear dynamics sequentially yielding a PDF for each step in time, as shown in Figure 9. Additional implementation details of the PF can be found in (Candy, 2009).

![Figure 9: Single step illustration of particle filtering prediction](image)

5.2 Shutdown condition characterization

In addition to the characterization and prediction of the unstructured power demands for the robotic system, uncertainties associated with the energy storage system must also be considered for run-time prognostics. Uncertainty in storage systems arises from natural production variation in devices (Tang et al., 2010), individual user operation (Kerasiotis et al., 2010), uncertainty in estimated SOC (Di Domenico et al., 2008), and power system enforced shutdown conditions (Tang et al., 2010). For example, instances of undercharged battery systems might occur, and as such the remaining run-time prognostics algorithm must account for variability in initial charge. For characterization of the shutdown condition uncertainty, two methodologies demonstrate levels of prediction fidelity.

For the first characterization methodology, an ideal storage mechanism is assumed for illustrative purposes in Section 6. The lack of prediction fidelity given a traditional load averaging scheme is shown via the idealized prediction scheme. The second characterization methodology deals directly with the battery model which is discussed extensively in Section 8. The ideal storage mechanism is assumed to exhibit 100% efficiency of energy transfer with no dissipation and also maintains a constant output voltage until full depletion. In the
case of this ideal device, system shutdown occurs given the energy consumed by the mobile robot exceeds the energy stored.

Characterization of the uncertainty of energy stored can be determined via many methodologies: experimental derivation (Tang et al., 2010), real-time estimation (Di Domenico et al., 2008), and uncertainty propagation via Peukert’s law (Ross and Budney, 1995). Given uncertainty in both the predicted energy consumed and estimated energy stored, the probability of shutdown at a given instant of time can be calculated via,

\[
P(\dot{E}_L > E_B) = \int_{-\infty}^{\infty} \Phi_{\dot{E}}(t) \phi_E(t) dt, \tag{13}
\]

where \(\Phi_{\dot{E}}\) is the cumulative density function (CDF) of the predicted consumed energy and \(\phi_E\) is the estimated probability density function of the energy in the ideal energy storage device. Comparison of analytical failure condition PDFs often arises from reliability theory (Green and Bourne, 1972) and fault detection prognostics (Vachtsevanos et al., 2006).

For the characterization of shutdown conditions given a battery system, the battery protection system and bus line regulating converters enforce system shutdown prior to complete exhaustion of the storage device via terminal voltage (Tang et al., 2010). In other words, the estimated SOC could remain positive while the battery protection system imposes shutdown. As such, the uncertainty associated with the terminal voltage at shutdown is required for run-time prognostics with a battery system. Characterization of battery voltage shutdown conditions remains straightforward and Figure 10 demonstrates the procedure.

### 6 Energy forecasting Packbot case study

Before illustrating full-fledged battery remaining run-time prediction, the ideal energy storage concept is used to demonstrate the erroneousness of energy prediction assuming averaged loads. Prediction results in the subsequent section utilize the ideal energy storage assumption. With accordance to the ideal power storage, given in Section 5.2, a straightforward integration of the power load gives the energy consumed. In this idealized scenario, when the total energy exceeds the stored energy, the system shuts down.

Load characterization for the PF utilizes the analysis shown in Section 4. To contrast with the PF forecast discussed in Section 5.1, linear variance propagation (or EKF prediction), given the mean and standard deviation of the load power for the first thirty minutes, is used for run-time prediction (Vachtsevanos et al., 2006). The ideal energy storage was assumed to hold an average of 175 kJ with a standard deviation of 10 kJ. Prediction of the mean consumed energy is shown in Figure 11 as compared to the experimental data.

The results from Figure 11 demonstrate the forecasting capabilities of the PF and EKF without the incorporation of model dynamics. While the mean predictions alone cannot provide conclusive evidence for forecast efficacy, analysis of the predicted run-time for each methodology can. Using the shutdown conditions discussed in Section 5.2, the remaining run-time probability density functions can be calculated. The resulting
Figure 11: Mean prediction (t = 30 min) comparison between the EKF and PF predictors with Packbot power load

run-time PDFs are illustrated in Figure 12.

Figure 12: Remaining run-time prediction made via model-based predictors at (t = 30 min) assuming ideal energy storage

As a result of the nonstationarity of the power demands, average load prediction for the ideal power storage (which exhibits no transient effects) demonstrates conservative predictions of run-time (Doerffel and Sharkh, 2006). Additionally, the linear variance estimator does not account for the multimodal nature of the power demands of unstructured terrain, and as such, prohibitively underestimates the variance of energy demands. Even without the incorporation of model dynamics into the prediction, existing methodologies underestimate the variance of predicted loads.

7 Dynamic battery model

Remaining run-time prediction for battery systems requires a dynamic model which captures the phenomena associated with variable current loading. The following discussion of the basic structure of an electrochemical storage device elucidates the necessity for a dynamic model. Two electrodes (anode/cathode) and an electrolyte separator form the basis for an electrochemical cell. While many cell chemistries exist for mobile applications, this discussion focuses on the general structure of all electrochemical devices (Pistoia, 2009).

The electrodes form half-cells which typically consist of a solid matrix electrolytic material suspended in additional electrolyte solution (Huggins, 2010). During the discharge process, the cathode (or positive electrode) accepts electrons (reduces) while the anode (negative electrode) loses electrons (oxidizes). Internally,
the electrolytic separator allows transport diffusion of ions between the electrodes. During the discharge reaction, the two electrode half-cells eventually reach potential equilibrium resulting in battery depletion (Huggins, 2010). During a discharge cycle with varying load currents, several macroscopic phenomena of interest occur, e.g., nonlinear discharge curve effects, rate-capacity effects, and recovery effects (Rakhmatov and Vrudhula, 2003).

A nonlinear relationship exists between the remaining SOC of a battery and the open circuit voltage, referred to as the battery discharge curve, shown in Figure 13(a). The rate-capacity and recovery effects consist of transient behavior in the electrochemical reaction as a result of ion diffusion speed and reaction polarization limitations, shown in Figure 13(b) (Huggins, 2010).

Figure 13: (a) Discharge curve for the UBI-2590 Li-Ion battery pack on the Packbot. (b) Rate effects on battery terminal voltage Adapted from (Huggins, 2010).

Myriad models exist in the literature to model the above electrochemical process (Chen and Rincon-Mora, 2006). First principles based models utilize porous electrode theory to model ion diffusion in cells (Doyle et al., 2000; Garcia et al., 2005). However, these physics-based models remain complex and are limited by the microstructure averaging approach of the porous electrode theory (Doyle et al., 2000; Garcia et al., 2005). Equivalent circuit models (ECM), conversely, model the electrochemical reaction via the lumped parameter abstraction with linear/nonlinear resistances and capacitances. Each lumped parameter empirically represents a physical process that occurs in the electrochemical process (Jongerden and Haverkort, 2009). Since ECMs cannot capture all electrochemical dynamics, accuracy suffers; however, modeling errors are typically no greater than 2% (Chen and Rincon-Mora, 2006). A spectrum of ECMs exist in the literature with a variety of applications in mind, i.e., battery impedance modeling (Chen and Rincon-Mora, 2006), SOC estimation (Di Domenico et al., 2008) and health prediction (Saha et al., 2007). To model the macroscopic battery effects discussed above without additional complexity, the Thévenin ECM, shown in Figure 14, was selected.

In the Thévenin ECM, the battery charge, $q_B$, indicates the remaining energy in the electrochemical system.

Figure 14: Thévenin equivalent circuit model of Li-ion battery system.
and explicitly dictates the battery open circuit voltage through the discharge curve characteristics shown in Figure 13(a). For the simplification of all expressions, battery charge is normalized, \( \bar{q} = q_B/q_0 \), with the maximum initial charge, \( q_0 \). Battery SOC is then computed directly from the normalized battery charge, \( SOC = \bar{q} \times 100\% \). Given the battery charge, \( \bar{q} \), internal feedback modulates the open circuit voltage according to the discharge curve, expressed as \( V_{oc} = \Gamma(\bar{q}) \). Internal lumped parameters, \( R_D, R_I, C_D \) and \( R_P \), are the dissipative resistance, internal resistance, diffusion capacitance and polarization resistance respectively. The diffusion and polarization coefficients model the transient effects of varying current on the output voltage of the battery. The nonlinear dynamic equations for the Thévenin ECM are given as follows,

\[
\begin{align*}
\dot{V}_D &= -\frac{V_D}{R_P C_D} - \frac{\Gamma(\bar{q})}{q_0 R_D} + \frac{1}{(1-D)} \left( \frac{1}{C_D} - \frac{1}{q_0} \right) I_L \\
\dot{\bar{q}} &= \frac{1}{(1-D)} \left( \frac{1}{C_D} - \frac{1}{q_0} \right) I_L
\end{align*}
\]

where the states, \( V_D \) and \( \bar{q} \) are the diffusion voltage and normalized remaining charge respectively, \( I_L \) is the load current and \( D \) is the boost converter duty cycle which regulates the bus line voltage at 26V. The battery terminal voltage is given by,

\[
V_B = \Gamma(\bar{q}) - V_D - \frac{R_I}{(1-D)} I_L
\]

where the duty cycle, \( D \), can be implicitly determined via \( D = (V_L - V_B)/V_L \). Since the bus line is fast regulated at 26V, the power converter controller is assumed instantaneous and \( V_L = 26V \).

\section{Packbot battery run-time prediction}

We now demonstrate the battery remaining run-time prediction capabilities of the particle filter framework with GMJM characterization of current demands utilizing the Packbot experimental data discussed in Section 4. During the field experiments, the Packbot housed a Li-ion battery with a rated voltage of 14.8V. Characterization of the parameter values for the prediction model and shutdown conditions for typical Li-ion battery packs was conducted with data from both (Saha and Goebel, 2007) and in house discharge tests. Prediction via the PF methodology was conducted with the GMJM characterized load data from Section 4. Figure 15 illustrates the measured and predicted battery voltage for all time. The “particle” cloud (95% confidence interval) shown in Figure 15 was computed piecewise in time.

Using the shutdown conditions discussed in Section 5.2, the run-time probability density functions can be calculated. The resulting remaining run-time PDFs are illustrated in Figure 16(a). For analysis of remaining run-time prediction fidelity over time, load characterization and particle filter predictions were conducted at five minute intervals ranging from 10 to 50 minutes. Prior to each prediction, the power load was characterized.
via the GMJM process given all measured data. Subsequently, the particle filter algorithm was employed to predict the run-time distribution. Comparison of run-time predictions to the ground truth, or actual, Packbot run-time forms the prognostic horizon (Saxena et al., 2008). Given the ground truth run-time of 55.15 minutes, the PF prediction interval made at 10 minutes should bound 45.15 minutes (the ground truth run-time at 10 minutes). As seen in Figure 17, the 95% PF run-time prediction envelope bounds the ground truth run-time for the entire prognostic horizon. Overall accuracy for the particle filter predictor can be summarized via the cumulative relative accuracy metric. The accuracy metric lies on the interval [0,1] with unity representing perfect prediction (Saxena et al., 2008). For the GMJM load characterization/PF scheme mean predictions, the cumulative accuracy was found to be 0.8767. Contrastingly, the EKF only approaches the actual run-time within the final 10 minutes of operation and exhibiting a conservative prediction for the prior duration of the prognostics horizon. Additionally, the 95% confidence interval for the EKF prediction is not visible given the absolute scale.

The PF, as expected, was found to require a higher computation time versus the EKF algorithm. Over the prognostics horizon, the PF run-time distributions were found in 30.25 ± 10.41 seconds whereas the EKF predictions were found in 3.65 ± 0.23 seconds on a 2.6 GHz processor. Resulting from the high computational load, the PF computational burden could be offloaded to a remote computer, given only a local microcontroller, for supervisory monitoring. The GMJM load characterization data could be transmitted efficiently, as only a Markov matrix and the distribution means, standard deviations and weights are needed for a full characterization.
9 Conclusions

The preceding discussion illuminated an approach for the self-supervised statistical prediction of battery remaining run-time for a mobile robotic systems operating on unstructured terrain. In contrast to existing run-time prediction techniques, the above procedure accounts for uncertain and transient load demands which provides for a less conservative remaining run-time prediction algorithm. Consequently, statistical predictions provide the operator with a quantifiable level of confidence for mission success. Both the self-supervised load characterization methodology and the statistically based prediction algorithm were elucidated technically and experimentally demonstrated.

The load characterization scheme, which is a soft computing and model-based approach, utilizes both Gaussian mixture modeling and jump Markov models. The GMM framework takes all existing load data and clusters load currents into Gaussian models via the expectation maximization algorithm. Determining the number of mixture models directly results from the analysis via the AIC, and thus the number of clusters is not arbitrarily selected. These transition probabilities between clusters were shown to form a jump Markov model which describes the likelihood of jumping between the various derived Gaussian mixtures. The GMJM characterization methodology was demonstrated with Packbot data acquired during operation on unstructured desert terrain.

Due to the inherent nonlinear nature of the environment and battery, a particle filtering prediction framework was employed. The PF predictor sequentially propagates the probability density function through the dynamic model for a full nonlinear statistical realization. For experimental demonstration, a Packbot was driven over generic unstructured desert terrain via directly observed remote control. The contrasting model-based EKF prediction approach, which only relied on load averaging as opposed to the GMJM procedure, was shown to always underestimate the remaining run-time distribution. However, the PF predicted run-time distribution was shown to accurately assess actual Packbot time to battery exhaustion over an extended prognostic horizon from 10 to 50 minutes.

In this study, implementation of the GMJM algorithm/particle filter for the run-time prediction algorithm utilized only one data set collected with a Packbot system. The single test remains the main limitation to this study, as an explicit proof of the statistical significance of the methodology is impossible. Furthermore, single data sets are prone to statistical overfitting. However, despite these flaws, this initial experiment demonstrates the feasibility of the methodology in a field setting and builds a framework for continued study in battery run-time prediction in the robotics community. Additionally, for further verification, a stochastic terrain simulator is being constructed in a laboratory setting, which provides for statistical control of the likelihood of loads encountered. In addition to the Packbot, the prediction scheme is being implemented on other robot topologies to demonstrate the generality of the algorithm. Statistical validation of the run-time prediction algorithm, using this setup and a large number of robot discharge tests, is underway.

Further refinements and extensions of the GMJM characterization methodology are possible. Other non-linear predictors which potentially offer computational advantages over the particle filter but still allow non-Gaussian states and jump-Markov inputs, such as the Gaussian sum filter, are being explored. Additionally, extensions of the GMJM characterization scheme to inform vehicle decision and energy storage aware control are ongoing. A model predictive control (MPC) methodology for short horizon energy aware control is currently under development. The goal of this MPC algorithm is to incorporate information from the GMJM load characterization and restrict command efforts given knowledge of a constrained battery SOC. Finally, integration of the predicted run-time PDF within a mission decision criterion is also in progress. Early efforts have focused on the prediction of remaining mission time via characterization of mission velocities and mission task times, and directly computing the mission success probability given the mission time PDF and the predicted run-time PDF. In addition to the mobile robotics community, the GMJM characterization methodology and remaining run-time prediction algorithms remain practical and promising for other devices/vehicles with local energy storage as the primary/sole power source. Next generation automobiles, medical devices, renewable energy power storage and consumer electronics are further
auspicious applications.

Acknowledgments

The authors acknowledge support by the Office of Naval Research through the Electric Ship Research and Design Consortium (ESRDC). Additionally, the authors thank Andrew Moore and Joey Groff from Southwest Research Laboratories and Dr. Eric Krotkov from Griffin Technologies for their support and insight into the field test results and data.

References


