Autonomous Vehicle Battery State-of-Charge Prognostics Enhanced Mission Planning

Bin Zhang[1], Liang Tang[2], Jonathan DeCastro[3], Michael Roemer[2], Kai Goebel[4]

[1]*University of South Carolina, Columbia SC, 29063, USA*
zhangbin@cec.sc.edu

[2]*Impact Technologies, LLC, Rochester NY, 14623 USA*
Mr.tangliang@gmail.com; Mike.Roemer@impact-tek.com

[3] *Cornell University, Ithaca, NY, 14850,USA*
jdc1177@gmail.com

[4]*NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

Most mission planning algorithms are designed for healthy systems. When faults occur in a system, it is advantageous to optimize the mission plan by taking the system health condition into consideration. In this paper, a mission planning scheme is proposed to integrate real-time prognostics in a receding horizon path planning framework to accommodate the system fault. In this scheme, the state-of-charge of a battery is monitored and predicted by a particle-filtering based prognostic algorithm. The predicted state-of-charge and remaining useful life of the battery are used in the mission planning to minimize mission failure risk. A series of experiments are presented on a robotic platform, which is powered by a Lithium-ion battery, to demonstrate and verify the proposed scheme.

1. INTRODUCTION

Unmanned systems are playing important roles in real-world applications ranging from surveillance, patrolling, search and rescue, operations in contaminated and denied areas to space exploration, law enforcement, and battlefield support, among others (Jan, Chang, & Parberry, 2008, Tsai, Huang, & Chan, 2011, Sudha & Mohan, 2011, Sun & Reif, 2007, Lu, Huo, Arslan, & Tsiotras, 2011). Fault may occur in systems due to fatigue, abrasion, aggressive operations, environment corrosion, deformation, etc. The fault will consequently cause changes of dynamics and introduce new constraints on mission completion. If no action is taken, system may fail unexpectedly and may cause loss of asset/system. It is therefore critical to guarantee system safety when fault occurs, especially under situations where maintenance is difficult or even unavailable such as in the middle of a mission. To achieve this goal, system resource needs to be reconfigured and/or mission needs to be re-planned to perform tasks with high priorities and, at the same time, minimize effects of the fault. In this paper, failure prognosis is integrated with mission planning to address this problem.

There are two categories of mission planning algorithms, deterministic searching (Singh, Simmons, Smith, Stentz, Verma, Yahja & Schwehr, 2000) and probabilistic or randomized searching (Kavrak, Svestka, Latombe, & Overmars, 1996). In this research, deterministic searching is utilized. In deterministic searching, the A* algorithm aims to find the least-cost path from starting point to goal. This scheme requires a full knowledge of the map. On the contrary, D* is an incremental search scheme (Stentz, 1994) that explores and updates the map as the vehicle explores the area. This is more realistic for applications where the whole map is unknown and needs to be built in real time via onboard sensors. D* and its variants have been widely used for autonomous vehicles, including the Mars rovers Opportunity and Spirit (Carsten, Rankin, Ferguson & Stentz, 2007). Among variants of D* algorithm, field D* is an interpolation based algorithm (Ferguson & Stentz, 2005, Koenig &

Likhachev, 2005) that is able to generate direct, low-cost path through a non-uniform environment.

Most of the current works on mission planning are concerned with healthy vehicles (Singh et. al., 2000, Kavraki, et. al., 1996, Stentz, 1994, Carsten, et. al., 2007, Ferguson, et. al., 2005, Koenig, et. al., 2005). The planning cost functions are typically terrain (for non-uniform environments), path distance, mission time, energy consumption, or a combination of these factors. The constraints on maneuverability, survivability, availability, and mission failure caused by faults are not considered in these papers. To address this problem, we propose a mission planning framework which takes fault progression into consideration. One of our goals is to make sure that fault growth is smaller than a threshold that indicates vehicle failure before the end of the mission. This requires integrating failure prognostics in planning to provide information of vehicle's fault state at current time instance and at the end of the mission. The prognostic algorithm also calculates remaining useful life (RUL) that can be used as a constraint to evaluate mission completion. With these new constraints on vehicle's health state and RUL, the mission will be planned to guarantee system safety.

In order for mission planning to be integrated with prognosis, fault detection and isolation (FDI) and failure prognosis (FP) are needed. There are many examples of efforts and applications on FDI and FP reported in the past decades (Chen, Zhang, Vachtsevanos, & Orchard, 2011, Zhang, Sconyers, Byinton, Patrick, Orchard & Vachtsevanos, 2011, Chen, Zhang, & Vachtsevanos, 2012, zhang, Khawaja, Patrick, Vachtsevanos, Orchard, & Saxena, 2009, Tang, Kacprzynski, Goebel, & Vachtsevanos, 2010, Gebraeel, Elwany, & Pan, 2009, Saha, Goebel, Poll, & Christophersen, 2009, Goebel, Saha, Saxena, Celaya, & Christophersen, 2008). In general, the approaches are divided into two main categories: model-based and data-driven. Model-based approaches use physics based laws to build a model and identify its parameters to describe fault development and progression for fault state detection and prediction. Data-driven approaches, on the other hand, use recorded data to either build a model or conduct statistical analysis to achieve these goals. Each approach has its own advantages and limitations. When a fault is detected, a diagnostics algorithm estimates the fault state and prognostics projects fault state at the end of mission and calculates the RUL of the failing system. This information is integrated in mission planning to achieve an optimal mission plan, which reduces risk to failures in both the system and the mission.

In this paper, a prognostics-enhanced mission planning approach is developed, in which prognosis and path planning are integrated in a receding horizon planning (RHP) framework. The main contributions include integration of particle filter based prognostics (prediction of fault state and RUL) with RHP, and implementation in a remotely controlled mobile robot platform. For this platform, the battery provides power to critical functions including command, control, communications, computers, intelligence, surveillance and reconnaissance (Gebraeel, et. al., 2009, Saha, et. al., 2009, Goebel, et. al., 2008). It is important to monitor the state-of-charge (SOC) of battery to make sure that the mission is accomplished or the robot returns to base before the end of charge. To demonstrate its efficiency, the proposed approach with a prognosis of battery SOC is implemented on a Pioneer 3-AT robotic platform and a series of experiments are presented, analyzed, and compared.

The remainder of this paper is organized as follows: Section II introduces an overview of prognostic and health management (PHM)-enhanced unmanned system. Section III discusses integration of particle filter based prognostics with mission planning in a receding-horizon framework. Section IV addresses hardware and fault mode of interest, which is a Lithium-ion battery state-of-charge. Section V presents a series of experiments, which is followed by concluding remarks in Section VI.

2. CONFIGURATION OF MISSION PLANNING

For mission planning of a faulty system, PHM plays a critical role that takes into account not only low-level system input and dynamics, but also high-level constraints on system health, mission completion, resources and system dynamics. Figure 1 shows the hierarchical structure of the proposed PHM enhanced mission planning implemented on a mobile robot platform with closed-loop controllers. In this paper, we will not discuss the low-level system dynamics and its changes caused by faults. The focus of the paper will put on constraints on system health, measured by predicted fault state at future time instants and RUL, and mission accomplishment.

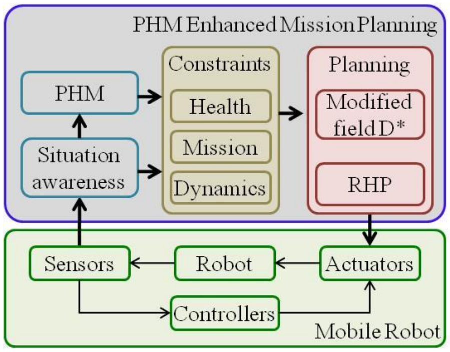


Figure 1: PHM-enhanced system hierarchy

In Figure 1, PHM module provides diagnostics and prognostics while situation awareness module provides map including obstacle identification and terrain classification. Since our focus is on failure prognosis, it is assumed that fault has been detected and isolated.

As we known, the prognostic algorithm predicts future fault state and RUL based on a prognostic model in the absence of measurements. Many uncertainties, such as future loading profile, environmental disturbances, and uncertainties with model parameters and structure, have impacts on prediction. As prediction horizon increases, the impacts of these uncertainties will accumulate and will eventually make the prognostic results less reliable in terms of accuracy and precision. It is necessary to properly handle these uncertainties in mission planning to achieve reasonable optimization results.

The map is constructed by evenly divided square grids. Nodes are defined on corners of grids. The onboard sensors are able to identify and classify the terrain of each grid. Therefore, the map is divided into two areas, observed area and unobserved unknown area. A RHP scheme is then applied to plan a path from vehicle's current location to the goal. The path is denoted as a series of waypoints located either on nodes or on edges of grids. As robot moves along a path, the onboard sensors are able to observe more grids and the path will be replanned if necessary.

To integrate prognostic into mission planning, the planned path will be used to calculate the future loading profile, which is an input to the model that predict the fault growth. The loading profile in the observed area is more accurate since the terrain is known while that in the unknown are is usually based on an assumption of terrains. This, along with RHP scheme, which only executes the path segments in the neighboring grid, will reduce the uncertainties of loading greatly.

Different from field D* algorithm that considers only neighboring grids, the RHP approach in Figure 1 searching is carried out in a much larger area that is observed by robot's onboard sensors and, because of this, the RHP approach is able to generate smooth path with more flexible heading directions without using interpolation. This is important because interpolation in field D* is based on an assumption of linearity, which does not hold when prognosis (typically nonlinear) is involved.

The following section outlines general (non-formatting) guidelines to follow. These guidelines are applicable to all authors and include information on the policies and practices relevant to the publication of your manuscript.

3. PROGNOSIS-ENHANCED MISSION PLANNING

Figure 2 illustrates the structure of prognostic enhanced mission planning. Map is built on assumption in the unknown area and the classified terrains in the observed area. The planning algorithm searches the optimal path based on a well-defined cost function, which usually includes multiple factors such as terrain and mission duration. In this paper, prognosis of robot health condition and/or RUL will be integrated into the cost function.

To this end, a particle filtering based prognostic algorithm is implemented to predict future health condition at the end of mission and calculate RUL. This result is evaluated against failure threshold to check if prognostic constraint is satisfied. Cost factors on mission duration and terrain are also investigated in the cost function to find the optimal path. If not all candidate paths can satisfy these constraints, path with minimum cost will be selected.

The optimal path is sent to the robot and the robot moves to the next waypoint. As the robot moves, the health state of the robot is updated in real-time with measurements and this becomes the initial condition for prognostics as re-planning becomes necessary either because of map update or change of operating condition. This process is repeated until robot reaches goal or it finds that there is no path to reach goal.

The challenge is that the predicted health state and RUL are usually described by a probability density function (pdf). In addition, traditional search algorithms like D* or field D* considers only neighboring grids. This brings difficulties to prognosis since grids are usually small and prediction horizons in neighboring grids are small to get any meaningful prognostic results.

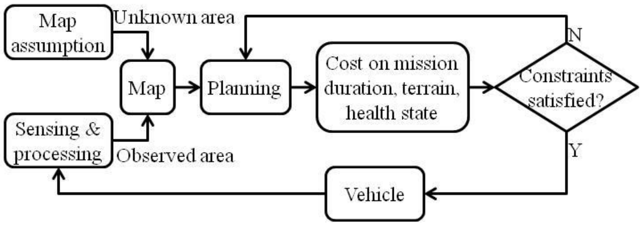


Figure 2: Prognostic enhanced mission planning

This limitation motives a RHP strategy that considers planning in an area beyond neighboring grids. In RHP, map is divided into implementation area, observation area, and unknown area. In the proposed approach, these areas are defined according to robot's current location when it reaches a waypoint, which is either on a node defined on a corner of a grid or a pointed located on an edge of a grid. The implementation area contains only neighboring grids next to vehicle's current location. When the robot waypoint is on a node (corner of a grid), the implementation area consist of 4 neighboring grids. When the robot is on an edge, the implementation area consists of 2 neighboring grids. The observation area consists of grids in the sensing range of onboard sensors centered at the robot's current location. The unknown area is the area remains unexplored. Observation area provides a reasonable large prediction horizon to conduct meaningful prognosis. This is important because loading profile in this area is known and prognostic routine can provide an accurate prediction result in this area. On the other hand, prognostic results in unknown area provide a reference of fault state prediction at the end of mission and RUL.

Since planning considers multiple factors, the cost function is a weighted sum of each individual factor. It is worth noting that the weighting factors play an important role in balancing effect of each individual factor. By properly assigning weighting factors, the path can be planned optimally according to the robot's health state. For example, if the fault is small, aggressive operations can be taken. However, if the fault grows fast and approaches a critical threshold, conservative operations need to be taken.

3.1. Receding-Horizon Planning

Field D* algorithm considers planning in neighboring grids (implementation area), it only carries very short prediction horizon. In addition, it is a fixed horizon planning scheme that has the following limitations. The first one is that it cannot handle the unexpected that is not included in the model. The second one is that it typically does not perform optimization as time approaches the end of horizon because there is little time to do any meaningful optimization. The RHP (Kuwata & How, 2011, Toit & Burdick, 2012), which is carried out iteratively in following steps, is introduced to address these limitations.

1. At waypoint p_i, a fixed horizon optimization problem over [p_{i+1}, p_{i+N}] is solved.
2. The robot moves to waypoint p_{i+1}.
3. Repeat optimization at waypoint p_{i+1} over [p_{i+2} p_{i+N+1}] and go back to step 2.

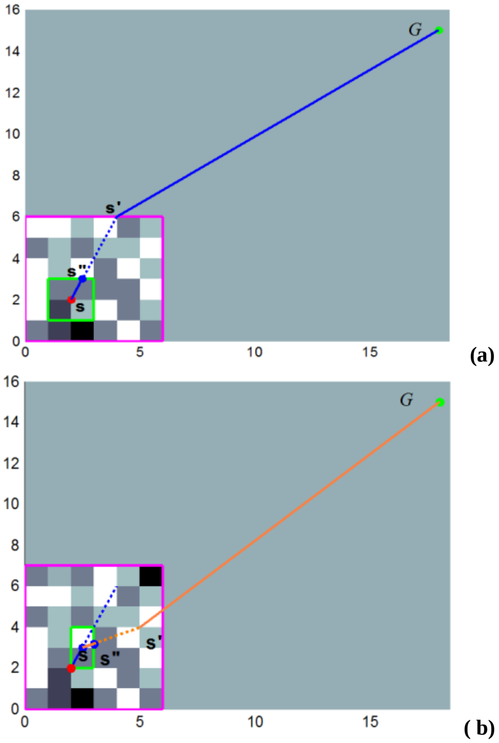


Figure 3 Illustration of receding-horizon planning; (a) Planning at step 1 from initial start s; (b) Planning at step 2 and *s* in this step is *s''* in step 1

Figure 3 illustrates the idea of RHP. In Figure 3 (a), vehicle location is *s*, green box is the implementation area, magenta box is the observation area, and outside magenta box is the unknown area. Note that in Figure 3 (a), the robot is located at *s*, which is a node on the corner of grid. The implementation area of the green box consists of 4 grids next to this *s* node. On the other hand, in Figure 3 (b), the robot is located at *s*, which is on the edge of grid. The implementation area of the green box consists of 2 grids next to this *s* waypoint. For each node in the observation area, it leads a candidate path from *s* to this node and to goal *G*. Then, the loading profile, mission time, and terrain etc. along this path can be obtained. A particle-filter based prognostic algorithm is employed to predict the fault state at the end of mission and RUL. These factors are used either in the cost function or as constraints to select the optimal path. In Figure 3(a), the optimal path is defined by node *s'* and is denoted as *s*-*s'*-*G*. The path segment *s*-*s'* crosses implementation area at *s''* and this waypoint is executed to move the robot to waypoint *s''*.

When the robot reaches *s''*, the map gets updated. The s'' in Figure 3(a) becomes the robot's location *s* as shown in Figure 3(b). The planning is carried out again for all nodes in the new observation area and a new optimal path is selected. Note that the observation areas (magenta boxes) and the implementation areas (green boxes) in Figure 3 (a) and (b) are different. This indicates that the increase of map exploration.

3.2. Particle Filter Based Prognostic Algorithm

Let $x_{0:t} \square \{x_0, x_1, \cdots, x_t\}$ and $y_{1:t} \square \{y_1, y_2, \cdots, y_t\}$ denote the signal and observations up to time *t*. For a filtering problem, it is of interest to estimate the posterior distribution $p(x_{0:t} \mid y_{1:t})$, the marginal distribution $p(x_t \mid y_{1:t})$, and the expectation $E[g(x_{0:t}) \mid y_{1:t}]$, where *g* is an integrable function.

Sequential Monte Carlo methods, also referred as particle filters, are a class of algorithms designed to obtain weighted samples sequentially from a target distribution $\{\pi_t\}$, denoted as a collection of *N* weighted random samples $\{w_t^{(i)}, x_{0:t}^{(i)}\}_{i=1,\cdots,N}$ such that it satisfies

$$\sum_{i=1}^{N} w_t^{(i)} \phi_t\left(x_{0:t}^{(i)}\right) \xrightarrow[N\to\infty]{} \int \phi_t\left(x_{0:t}\right) \pi_t\left(x_{0:t}\right) dx_t ,$$

where f_t is any p_t integrable function.

In Bayesian filtering, the target distribution $\pi_t(x_{0:t}) = p(x_{0:t} \mid y_{1:t})$ is the *a posterior* distribution of $x_{0:t}$.

Suppose that a set of *N* particles $\{x_{0:t-1}^{(i)}\}_{i=1,\cdots,N}$ distributed according to the target distribution is available at time *t*-1. The objectives of filtering is to obtain a set of *N* new particles $\{x_{0:t}^{(i)}\}_{i=1,\cdots,N}$ and this set of particles are distributed according to the target distribution at time *t*, *i.e.*, $\pi_t(x_{0:t})$.

To generate this new set of particles $\{x_{0:t}^{(i)}\}_{i=1,\cdots,N}$, the particles at time *t*-1 $\{x_{0:t-1}^{(i)}\}_{i=1,\cdots,N}$ are extended using a kernel function and obtain a new distribution $q_t\left(x_{0:t}^{(i)}\right)$, which is referred to the importance distribution or proposal distribution. To get the consistent estimate of posterior distribution $p\left(x_{0:t}^{(i)} \mid y_{1:t}\right)$, importance sampling corrects the difference between $q_t\left(x_{0:t}^{(i)}\right)$ and $p\left(x_{0:t}^{(i)} \mid y_{1:t}\right)$ by setting weighting factors for each particle, which is given by:

$$\tilde{w}\left(x_{0:t}^{(i)}\right) = \frac{p\left(x_{0:t}^{(i)} \mid y_{1:t}\right)}{q_t\left(x_{0:t}^{(i)}\right)}$$

and is normalized as

$$w\left(x_{0:t}^{(i)}\right) = \frac{\tilde{w}\left(x_{0:t}^{(i)}\right)}{\sum \tilde{w}\left(x_{0:t}^{(i)}\right)}.$$

With this new set of weights, the target distribution can be approximated as:

$$\pi(x_{0:t}) = p\left(x_{0:t} \mid y_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)} \delta\left(x_{0:t} - x_{0:t}^{(i)}\right) \tag{1}$$

In a simple case of particle filter, Bootstrap filter, the importance density function is set as the *a prior* pdf $q_t\left(x_{0:t} \mid x_{0:t-1}\right) = p\left(x_t \mid x_{t-1}\right)$. In this setting, the weights for the newly generated particles are proportional to the likelihood of new observations, i.e.

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p\left(y_{1:t} \mid x_{0:t}^{(i)}\right) = w_{t-1}^{(i)} \cdot p\left(y_t \mid x_t^{(i)}\right) \cdot \tag{2}$$

Particle filter based prognostic algorithm is used to predict fault growth and estimate the RUL and is beyond filtering problem itself. The initial health state of prognosis is given by a Gaussian distribution or a uniform distribution based on the prior knowledge about battery charge condition.

The prognostic algorithm involves three steps.

Step 1: The first one is to estimate the current health state with real-time measurement as shown in Equation (2). Such a measurement is often a feature or condition indicator that is extracted from the robot's raw measurement, such as current, voltage, vibration, temperature, or others. This step is a one-step-ahead prediction plus correction. The output is *a posteriori* state estimate, which is used as the initial condition for the second step, prognostics or long-term prediction of battery charge and remaining useful life.

Step 2: In the second step, a long-term *p*-step ahead prediction for the distribution of fault state is generated using the fault growth model and the *a posterior* pdf from the first step recursively, which can be described as

$$p\left(x_{t+p} \mid y_{1:t}\right) \approx$$
$$\sum_{i=1}^{N} w_t^{(i)} \int \cdots \int p(x_{t+1} \mid x_t^{(i)}) \prod_{j=t+2}^{t+p} p(x_j \mid x_{j-1}) dx_{t+1} \cdots dx_{t+p-1}$$

In prognosis, the prediction depends on solely fault progression models since there are no measurements available. In addition, the calculation often requires significant computation, especially when the prediction horizon is large. To simplify the computation, the weights of particles remain unchanged for prediction. This is based on the assumption that the error caused by invariant weights is negligible compared to other sources of error, such as model inaccuracies. In practice, this assumption is valid and the results from this approach satisfy the engineering requirements.

Step 3: To estimate the remaining useful life, a failure threshold F_f is defined to indicate the failure of system. With this threshold and the prediction of future fault state distributions at future time instants from step 2, the law of total probabilities can be applied to compute the probability of failure at any future time instant, *i.e.* the time to failure distribution:

$$p_{failure}(t) = \sum_{i=1}^{N} \Pr\left(x_t^{(i)} = F_f\right) w_t^{(i)}$$

For battery SOC prognosis, the first step uses battery voltage and current measurement to estimate the current battery SOC. In step 2, the predictions of battery SOC at the future time instants are calculated. In the third step, a critical battery charge is defined as a threshold value indicating the battery death. The predicated remaining charge is compared against this critical value to estimate the time-to-failure and RUL of the battery. With the RUL distribution, the statistical estimations and confidence intervals of them can be calculated and used in mission planning.

3.3. Prognostics in Mission Planning

To implement particle filter based prognosis, a fault progression model is needed. The model can be written as:

$$f_{k+1} = F\left(f_k, L\right) + N_F \tag{3}$$

where f_k is the fault state at time instant k, L is the load applied to the robot, and N_F is noise subject to $N(0, Q_F)$.

The load L is a function of the robot's velocity v and terrain d, which indicates difficulty to traverse the terrain,

$$L = G(d, v) + N_L \tag{4}$$

where N_L is noise subject to $N(0, Q_L)$. With Equation (4), the selection of paths containing different terrains will have a direct influence on the robot's loading profile and, consequently, have a direct influence on fault growth.

Figure 4 is an illustrative example of the prediction of fault growth from the robot's current location s to the goal G. In this figure, horizon axis is the time of robot operation and vertical axis is the usage of battery charge. Particle filtering based algorithm is used since it is suitable to handle nonlinear fault growth applications with non-Gaussian noises. When the prognostic algorithm is carried out, a number of particles, with different initial values, are projected in the time horizon from current time instant to the end of the mission using a fault progression model. These particles form a fault state at each future time instant. By defining a critical threshold indicting the failure, the time instant that each particle cross the critical value can be obtained from which a RUL pdf can be estimated as shown in Figure 4.

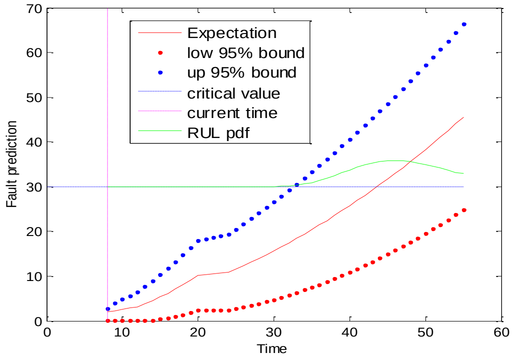


Figure 4 Failure prognosis and RUL pdf

It is worth noting that the predicted fault state and RUL have a large uncertainty due to the fact that there are noise and unmodeled uncertainties and there is no measurement available for correction. The uncertainty is also illustrated in Figure 4, in which the predicted fault states are presented by expectation and lower and upper bounds of 95% confidence interval. From this figure, the growth of uncertainties is clear and the 95% confidence bound become larger with the increase of prediction horizon. More details of particle filter can be found in (Arulampalam, Maskell, Gordon, and Clapp, 2002).

3.4. Prognosis Enhanced Receding Horizon Planning

The prognostic algorithm predicts the fault growth from the robot's current location *s* to goal *G*. The map is divided into three areas: implementation area, observation area, and unknown area, as shown in Figure 5. In this figure, prognosis in the observation area is denoted as short-term prognosis while prognosis in the whole path is denoted as long-term prognosis.

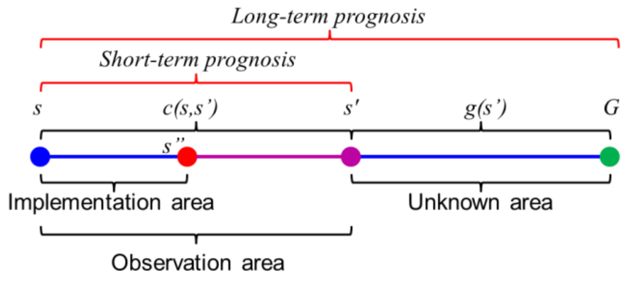


Figure 5 Short-/long-term prognosis and its corresponding areas

In Figure 5, the optimal path is assumed given from *s* to *s'* and to *G*, where *s'* is a node in the observation area that has the minimum cost. This optimal path crosses the implementation area at *s''* (as in Figure 3(a)). Note that *s''* could be located on either a node or any location of an edge. Path segment *s*-*s''* will be execute by the robot. During the execution, loading profile and measurement become available. With this information the fault state at each time instant can be estimated and updated to achieve *a posterior* pdf of the fault state.

Suppose each candidate path, defined by each node in observation area cross the boundary of the observation area at s_e. Then, each path can be denoted as $s \rightarrow s' \rightarrow s_e \rightarrow$goal. Clearly, the map information in $s \rightarrow s' \rightarrow s_e$ is always known and is the short-term prognosis horizon shown in Figure 5. The cost factors for path segment $s \rightarrow s'$, $s' \rightarrow s_e$, and $s_e \rightarrow$goal are denoted as $\boldsymbol{c}(s,s')$, $\boldsymbol{c}(s',s_e)$, and $\boldsymbol{g}(s_e)$, respectively. The cost on the entire path is:

$$\min_{s'\in obsv(s)} \left(c(s,s')+c(s',s_e)+g(s_e)\right) \tag{5}$$

where *obsv*(*s*) is the set of all node in the observation area when the robot is at *s*.

If *s'* is located on the boundary of the observation zone, we have $s'=s_e$. In this case, the cost on the entire path reduces to:

$$\min_{s'\in obsv(s)} \left(c(s,s')+g(s')\right) \tag{6}$$

In traditional planning algorithms, the cost factors only carry deterministic factors such as terrain, distance, and mission duration. The prognostic results, however, are given in a form of pdf. To include prognosis in the cost function, there are two means. The first one is to evaluate path in a probabilistic sense.

The second one is to convert state or RUL pdf to a value, such as the expectation and fault value at risk (FVaR). In this paper, the second means is selected. The fault state pdf is described by a single value of interest in the cost function.

FVaR has been adapted from the field of actuarial science and has been provided as a means for online evaluation of the severity of a fault condition (Precht, 2000, Schreiner, Balzer, & Precht, 2010). FVaR with a given risk α, denoted as FVaR$_\alpha$, is defined as a threshold of fault dimension such that probability of actual fault dimension being larger than this threshold is at most 1-α. That is, $p(x>\mathrm{FVaR}_\alpha)=1-\alpha$, where x is the fault dimension. In other word, FVaR$_\alpha$ associated to the fault state pdf is computed at a degree of confidence α.

3.5. Cost Function

The planning is multi-objective and the cost function considers factors of mission duration, terrain, and FVaR$_{95\%}$. Terrain factor in the cost makes the robot select the easiest path, which will reduce wear of components and lower mission failure risk. Mission duration factor in the cost makes robot find the quickest way to accomplish mission. When a fault occurs, FVaR$_{95\%}$ in the cost factor makes robot find path with the least fault growth at the end of mission. There is a trade-off between these factors.

The cost function is then a weighted sum defined as:

$$\begin{aligned}\min_{s'} J = {} & w_T\left(c_T(s,s')+c_T(s',s_e)+g_T(s_e)\right)+\\ & w_D\left(c_D(s,s')+c_D(s',s_e)+g_D(s_e)\right)+\\ & w_P\left(c_P(s,s')+c_D(s',s_e)+g_P(s_e)\right)\end{aligned} \tag{7}$$

where subscript *T*, *D*, and *P* stand for mission duration, terrain value, and prognostics, respectively; $\boldsymbol{c}(s,s')$, $\boldsymbol{c}(s',s_e)$

and $\boldsymbol{g}(s_e)$ are cost on segment $s \rightarrow s'$, $s' \rightarrow s_e$, and $s_e \rightarrow goal$, respectively; weighting factors w_T, w_D, and w_P have feature of $w_T+w_D+w_P=1$.

4. EXPERIMENTAL RESULTS

To verify the proposed PHM-enhanced mission planning, it is implemented on a mobile robotic platform powered by a lithium-ion battery. The prognostic algorithm is designed to monitor the battery state-of-charge (SOC).

4.1. Robot Hardware

The experiments are based on a Pioneer 3-AT robotic platform, as shown in Figure 6. The platform hosts an onboard computer that supports build-in controllers, sonar sensors, serial communications, encoders and other autonomous functions.

In the following experiments, the robot is allowed to travel at three velocities, 0.4m/s, 0.25m/s, and 0.2m/s. The robot has an acceleration phase after a speed command is received and a deceleration phase when it approaches a waypoint.

The robot is powered by two separate batteries. One battery powers the onboard computer, auxiliary sensors, and peripheral equipment. The onboard computer communicates with the build-in controller via a serial port. It also communicates with other computers (such as a remote client laptop running path planning algorithm) on a network through a wireless access port. The second battery, which is a 2.4A-hr $LiFePO_4$ Lithium-ion battery, powers the robot and we monitor the SOC of this battery. A NI PCI-6229 data acquisition (DAQ) card has been added to the onboard computer to measure current and voltage of the battery.

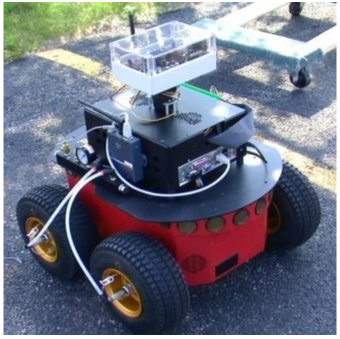


Figure 6: The robot platform

In an actual mission, robot would experience different battery loading scenarios based on terrain and velocity. To simulate this in an indoor, 2-D environment, a variable load has been attached to the battery. This variable load is made up of three resistors with resistance of 6.23Ω, 12.5Ω, and 25Ω, respectively. The resistors are wired in parallel to the battery. The onboard computer controls the connection/ disconnection of each resistor. It provides eight different loading scenarios progressing linearly in magnitude. When robot moves into a grid of high terrain value, it activates a larger loading scenario requiring high current draw using the variable load. When robot moves into a grid of low terrain value, it activates a low loading scenario requiring low current draw. This allows for many simulated terrains while keeping the robot in a safe environment.

4.2. System Configuration

The system configuration is illustrated in Figure 7. There are two computers in this configuration. The onboard computer hosts prognosis server, DAQ server, and UGV controller. It realizes all low-level functions including sensor data acquisition and analysis, robot position and velocity control, and localization, and prognosis. The remote client computer is configured to perform high-level functions such as path planning (Tang, L., Zhang, B. DeCastro, J., & Hettler, E., 2011, Zhang, B., Tang, L., DeCastro, J., & Goebel, K. 2011, Tang, L., Hettler, E., Zhang, B., & DeCastro, J. 2011).

The mission planning starts from the robot's current location and its observed area with onboard sensors. As soon as a path is planned, the first waypoint and loading profile of the path are sent to robot for execution according to RHP.

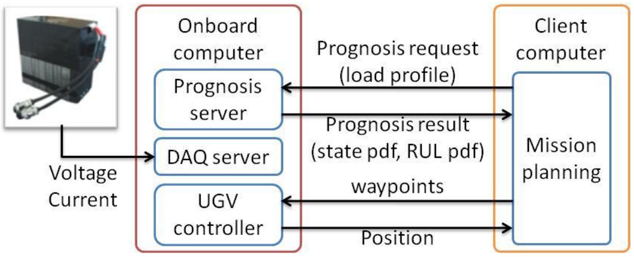


Figure 7 Hardware architecture

As robot moves along the planned optimal path, the voltage and current of the battery are collected via DAQ card. These data are used by the prognostic algorithm to estimate the battery's current SOC in real time. Then, the SOC pdf at the end of the mission and the RUL pdf are predicted. The SOC pdf and RUL pdf are sent back to client computer. At the same time, the map is updated via measurements of the robot's onboard sensors as the robot moves. After the robot reaches the waypoint, the current SOC pdf is used as initial condition for prognosis for re-planning.

4.3. Fault Growth Model and Prognosis

The fault growth model is given as follows:

$$\begin{cases} x_1(k+1) = x_1(k) + w_1(k) \\ x_2(k+1) = x_2(k) + C_1 \cdot v(k) \cdot I(k) \cdot \Delta T(k) + w_2(k) \\ v(k) = V_0 + C_2 \cdot x_2(k) - x_1(k) \cdot I(k) + o(k) \end{cases} \quad (8)$$

where x_1, x_2 are states indicating model parameters and the battery remaining charge, respectively, v is the battery voltage, I is the current drawn from the battery, ΔT is the sampling period indicating time duration for the current draw, C_1, C_2 are fixed constant, V_0 is the initial battery voltage, w_1,

w_2 are process noises while o is observation noise, k is index of time instant. In the model, C_1=10^{-5}, C_2=-5.5687, and V_0=12.

In this model, x_1 and x_2 are related to battery voltage tracking, in which x_1 is a parameter that reflects the relationship between battery voltage drop and current draw while x_2 is the battery remaining charge. The battery remaining charge is calculated as the current charge minus the voltage times current during a sampling period. The voltage drop is also related to battery remaining charge as shown in the third equation.

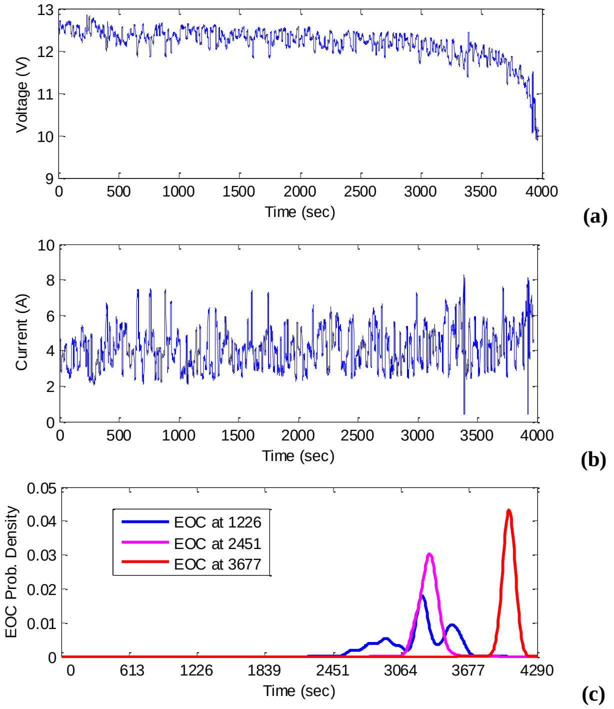


Figure 8: Battery data and end-of-charge prediction: (a) Voltage; (b) Current; (c) End-of-charge prognosis

Figure 8 shows a set of battery voltage and current data collected on the testbed running a random varying load and the prognosis results. It is clear that the voltage in subplot (a) is mainly affected by the load before the battery charge reaches a critical level at about 3500 seconds, then the voltage drops drastically to 10 V within 500 seconds. The current in subplot (b) is totally determined by the load and only a trivial increase tending is observed towards the end when voltage drops. In the subplot (c), three end-of-charge predictions probability density functions are predicted at 1226, 2451 and 3677 seconds, respectively, are shown. Since the true end-of-charge time in this case is 3990 seconds, it can be seen that predictions at 1226 and 1451 seconds have been made conservatively due to the uncertainties associated with initial charge state and future load. The prediction at the 3677 second was made when the voltage signal started to drop and was rather accurate.

4.4. Map

Using satellite images of the NASA Mars yard shown in Figure 9, a 37-meter by 20-meter low resolution map is scaled. The map has 8 types of terrains indicating 8 different difficulty levels. As mentioned earlier, each type of terrain is simulated by a corresponding resistor configuration connected to the battery. In the map, the 8 different terrains are indicated by colors from white to black with increase of difficulty to traverse and black is actually indicating obstacle.

Using the prognosis enhanced mission planning presented in previous sections, the robot navigates the simulated Mars Yard. In each case, the robot starts from coordinates X=2 m, Y=18 m to goal X=20 m, Y=7 m. The algorithm is set such that the robot can observe terrain within 2 meters of its center point. In each scenario, we use different weights in the cost function but keep the same starting point and the same goal.

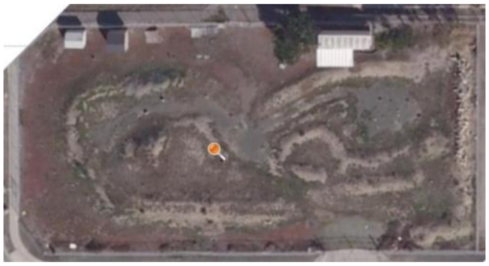


Figure 9: Satellite image of the NASA Mars Yard

4.5. Experimental Results

This section provides a series of experimental results to demonstrate the integration of prognosis in mission planning. The effects of each individual factor in the path can be reflected by adjusting weighting factors in the cost function. For these experiments, the battery starts at a full charge condition. After each experiment, the battery is fully recharged before the start of the next experiment.

4.5.1. Experiment 1: Prognostic optimality

The weighting factors in this experiment are set as [w_T, w_D, w_P] = [0, 0, 1]. Since we only have weighting factors on prognosis, the robot plans a path that minimizes the battery usage. The planned path is shown in Figure 10. This figure shows that mission duration is around 126 seconds and path distance is about 24.8 meters.

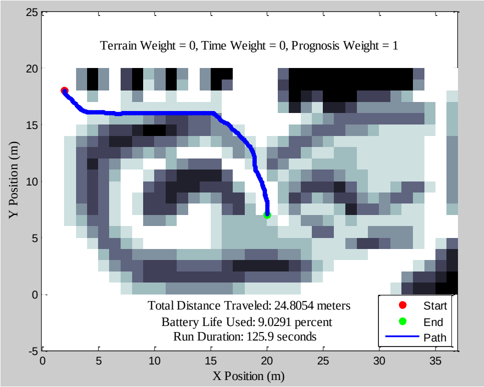


Figure 10: Planned path with [w_T, w_D, w_P] = [0, 0, 1]

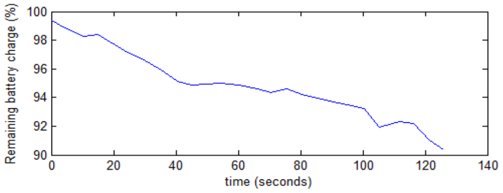


Figure 11 Expectation of SOC along path with [w_T, w_D, w_P] = [0, 0, 1]

Figure 11 shows the expectation of battery SOC pdf at each waypoint along the mission. This figure shows that the battery uses about 9% of its charge to accomplish the task.

4.5.2. Experiment 2: Mission duration optimality

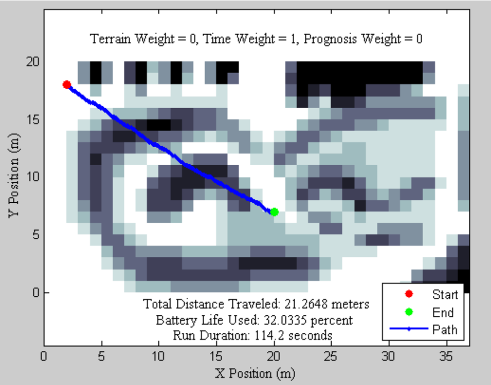


Figure 12 Planned path with [w_T, w_D, w_P] = [1, 0, 0]

When weighting factors change to [w_T, w_D, w_P] = [1, 0, 0], the planning algorithm seeks the shortest time to accomplish the mission. To this end, the robot finds the shortest path and travels at the highest speed in this scenario. The results are shown in Figure 12. It can be seen that the path is almost a straight line. The path distance is about 21.3 meters and mission duration is about 114.2 seconds.

The expectation of the battery SOC along the path is shown in Figure 13. In this case, the robot uses a large amount of energy, which is around 32% of the battery charge. The reason is that, when the robot travel in those grids with high terrain values, the current drawn from the battery is large.

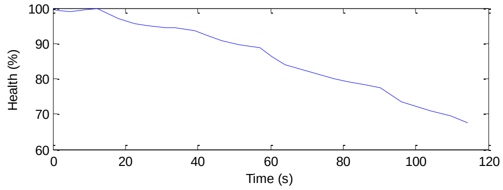


Figure 13 Expectation of SOC along path with [w_T, w_D, w_P] = [1, 0, 0]

4.5.3. Experiment 3: Terrain optimality

The weighting factors are given as [w_T, w_D, w_P] = [0, 1, 0]. In this case, the optimization is only carried out to find the easiest path with lowest terrain values. The prognostic results and the robot velocity do not have influence on path planning. In this experiment, the robot velocity is selected as the lowest speed at 0.2m/s. Figure 14 shows the path planned, which has a distance of about 24.6 meters and the mission duration is about 202.2 seconds. It can be seen that the path is similar to the one shown in Experiment 1.

Figure 15 shows the expectation of the SOC along the path. Although the path is similar to that in Experiment 1, the use of the battery energy is very different. In this experiment, the robot uses about 17.7% of battery charge to accomplish the mission. The battery energy use is about twice that in the Experiment 1.

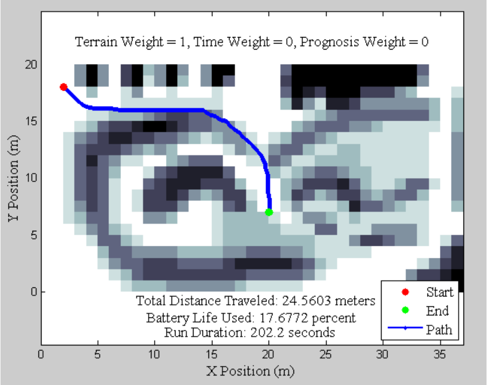


Figure 14 Planned path with [w_T, w_D, w_P] = [0, 1, 0]

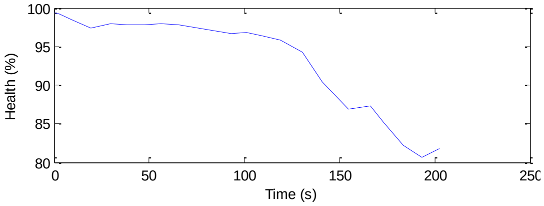


Figure 15 Expectation of SOC along path with [w_T, w_D, w_P] =[0 1 0]

4.5.4. Experiment 4: Weighted-sum optimality

In most cases, the cost function is a weighted sum of these factors in which none of the weighting factors are zero. In this experiment, the weighting factors are set as [w_T, w_D, w_P] = [0.33, 0.33, 0.34]. That is, the contribution of each individual factor in the cost function is equal.

Figure 16 shows that planned path. The path distance is about 24.5 meters and it takes the robot about 126.4 seconds to accomplish the mission.

Figure 17 shows the expectation of SOC pdf at each waypoint along the path. The use of battery charge in this experiment is about 11.9%. Clearly, this planning is a mixed case of the previous three cases. It uses less battery charge than the mission duration-based (Experiment 2) and the terrain- based (Experiment 3) planning. On the other hand, it uses more battery charge than the prognosis-based (Experiment 1) planning.

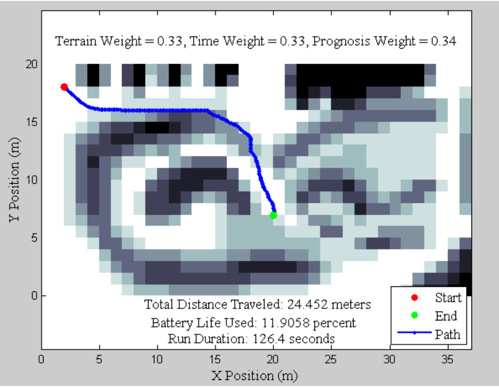


Figure 16 Planned path with [w_T, w_D, w_P] = [0.33, 0.33, 0.34]

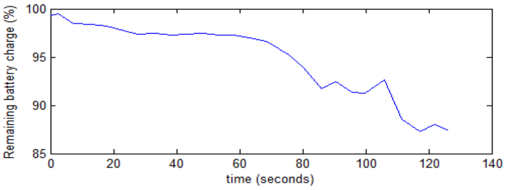


Figure 17 Expectation of SOC along path with [w_T, w_D, w_P] = [0.33, 0.33, 0.34]

4.5.5. Summary of Results

The experimental results are summarized in Table 1. From the comparison of battery charge usage, a general conclusion is that the battery charge use with prognostics in mission planning is less that that without prognosis.

Table 1 Summary of results

Weight factors	w_D	1	0	0	0.33
	w_T	0	1	0	0.33
	w_P	0	0	1	0.34
Charge use (%)		17.68	32.03	9.03	11.91
Distance (meters)		24.56	21.26	24.81	24.45

5. CONCLUSION

Technological advancements have enabled unmanned systems to play more and more important roles and perform a wide variety of present and future critical missions. Fault can occur in unmanned systems and cause degradation in performance and reduction in system availability. It is therefore critical to introduce prognostics and health management (PHM) into system design, development, and deployment. Under fault scenarios, mission planning should consider system health condition and/or RUL to guarantee system safety and mission success. To address this challenge, this paper presents a prognostics-enhanced mission planning scheme to accommodate system faults. The proposed method includes a failure prognostic algorithm and a mission planning algorithm. A mobile robot platform powered by a Li-Ion battery is utilized to verify the proposed scheme by monitoring battery SOC. A series of experiments are presented to verify the proposed method.

REFERENCES

Jan, G., Chang, K., & Parberry, I, (2008), Optimal path planning for mobile robot navigation, *IEEE/ASME Trans. on Mechatronics*, 13(4): 451-460.

Tsai, C.-C., Huang, H.-C., & Chan, C.-K. (2011), Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation, *IEEE Trans. on Industrial Electronics*, 58(5): 1907-1920.

Sudha, N. & Mohan, A. (2011), Hardware-efficient image-based robotic path planning in a dynamic environment and its FPGA implementation, *IEEE Trans. on Industrial Electronics*, 58(5): 1907-1920.

Sun, Z. & Reif, J. (2007), On robotic optimal path planning in polygonal regions with pseudo-Euclidean metrics, *IEEE Trans. on Systems, Man, & Cybernetics: B*, 37(4): 925-936.

Lu, Y., Huo, X., Arslan, P., & Tsiotras, P. (2011), Incremental multi-scale search algorithm for dynamic path planning with low worst-case complexity, *IEEE Trans. on Systems, Man, & Cybernetics: B,* 41(6): 1556-1570.

Singh, S., Simmons, R., Smith, T., Stentz, A., Verma, V., Yahja, A. & Schwehr, K. (2000). Recent progress in local and global traversability for planetary rovers, *Proc. IEEE Int. Conf. Robotics & Automation*, vol. 2, San Francisco, CA, 1194-1200.

Kavraki, L., Svestka, P., Latombe, J., & Overmars, M. (1996), Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. on Robotics & Automation*, 12 (4): 566–580.

Stentz, A. (1994), Optimal and efficient path planning for partially-known environments, *Int. Conf. Robotics & Automation*: 3310–3317.

Carsten, J., Rankin, A., Ferguson, D., & Stentz, A., (2007), Global path planning on board the mars exploration rovers, *IEEE Aerospace Conf.,* 1-11.

Ferguson, D. & Stentz, A. (2005), Field D*: an interpolation-based path planner and replanner, *Intl. Symp. Robotics Research*.

Koenig, S. & Likhachev M. (2005), Fast re-planning for navigation in unknown terrain, *IEEE Trans. Robotics & Automation*, 21(3), 354-363.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard M. (2011), Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filter, *IEEE Transactions on Industrial Electronics*, 58(9): 4353 - 4364.

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M., & Vachtsevanos G. (2011), A probabilistic fault detection approach: application to bearing fault detection, *IEEE Transactions on Industrial Electronics*, 58(5): 2011 – 2018.

Chen, C., Zhang, B., & Vachtsevanos, G. (2012), Prediction of machine health condition using neuro-fuzzy and Bayesian algorithms, *IEEE Trans. on Instrumentation & Measurement*, 61(2): 297-306.

Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M., & Saxena, A. (2009), Application of blind deconvolution de-noising in failure prognosis, *IEEE Trans. on Instrumentation & Measurement*, 58(2): 303-310.

Tang, L., Zhang, B., DeCastro, J., & Hettler, E., (2011) An integrated health and contingency management case study on an autonomous ground robot, *IEEE International Conference on Control and Automation*, 584-589.

Zhang, B., Tang, L., DeCastro, J., & Goebel, K. (2011), Prognostics enhanced receding horizon mission planning for field unmanned vehicles, *AIAA Guidance, Navigation and Control Conference*, Portland, OR.

Tang, L., Hettler, E., Zhang, B., & DeCastro, J. (2011), A testbed for real time autonomous vehicle PHM and contingency management applications, Annual Conference of the Prognostics and Health management Society, 1-11.

Tang, L., Kacprzynski, G., Goebel, K. & Vachtsevanos, G. (2010), Case studies for prognostics-enhanced automated contingency management for aircraft systems, *IEEE Aerospace Conf.*, Big Sky, Montana.

Gebraeel, N., Elwany, A., & Pan, J. (2009), Residual life predictions in the absence of prior degradation knowledge, *IEEE Trans. on Reliability*, 58(1): 106-117.

Saha, B., Goebel, K., Poll, S. & Christophersen, J. (2009), Prognostics methods for battery health monitoring using a Bayesian framework, *IEEE Trans. on Instrumentation & Measurement*, 58(2): 291-296.

Goebel, K., Saha, B., Saxena, A., Celaya J., & Christophersen J. (2008), Prognostics in battery health management, *IEEE Instrumentation & Measurement Magazine*, 11(4): 33-40,.

Kuwata, Y., & How, J. (2011), Cooperative distributed robust trajectory optimization using receding horizon MILP, *IEEE Trans. on Control Systems Technology*, 19(2): 423-431.

Toit, D. & Burdick, J. (2012), Robot motion planning in dynamic, uncertain environments, *IEEE Trans. on Robotics*, 28(1), 101-115.

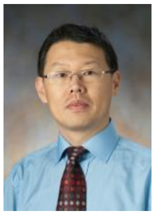
Arulampalam, M., Maskell, S., Gordon, N., & Clapp, T. (2002), A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. on Signal Processing*, 50(2): 174-188, 2002.

Precht, A. (2000), An empirical evaluation of value at risk by scenario simulation. *Journal of Derivatives*, 7:1074-1240.

Schreiner, A., Balzer, G., & Precht, A. (2010), Risk sensitivity of failure rate and maintenance expenditure: application of VaR metrics in risk management. *IEEE Mediterranean Electrotechnical Conf.*, 1624-1629, 2010.

BIOGRAPHIES

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



Dr. Liang Tang is currently a Principal Engineer at Pratt & Whitney with over 15 years' experience in the development of prognostics & health management systems. He received a Ph.D. degree from Shanghai Jiaotong University in 1999 and worked at Georgia Institute of Technology as a Post Doctoral Fellow prior to joining UTC. In 2004, Liang joined Impact Technologies, a Sikorsky Innovations company, where his work focused on the research and development of innovative PHM and automated fault accommodation solutions for a wide range of aerospace systems and components. He was the Principal Investigator on several government funded SBIR/STTR/NRA programs and published over 50 technical papers. Liang is a member of the PHM Society and serves as an Assoc. Editor of International Journal of Prognostics & Health Management.

Jonathan DeCastro is a Ph.D. student in Mechanical and Aerospace Engineering at Cornell University. His current research focuses on synthesis of provably-correct controllers for complex robotics. Prior to Cornell and after receiving his B.S. and M.S. degrees in Mechanical Engineering at Virginia Tech, he spent several years in industry as an aerospace engineer at NASA and Impact Tech (now Sikorsky).

Michael J. Roemer is currently a Technical Fellow with Sikorsky Aircraft with over 20 years' experience in the development of automated health monitoring, diagnostic, and prognostic systems for a wide range of military and commercial applications. He was previously the co-founder and Director of Engineering at Impact Technologies prior to its acquisition by Sikorsky/UTC. His experience includes a wide range of integrated vehicle health management system implementations to detect and predict system faults in real time and perform automated troubleshooting and maintenance planning. He has developed several diagnostic and prognostic capabilities utilizing technologies such as dynamic signature analysis, artificial intelligence, aero-thermal performance monitoring, finite element modeling, probabilistic remaining life analysis, and risk assessment methods. He is the co-founder and past Vice President of the PHM Society, past Chairman of the SAE HM-1 Integrated Vehicle Health Management Committee, board member and past Chairman of the Machinery Failure Prevention Technology (MFPT) Society, Member of the IGTI Marine Committee and ASME Controls and Diagnostics Committee, and Adjunct Professor of Mechanical Engineering at the Rochester Institute of Technology. He is the co-author of a book titled "Intelligent Fault Diagnosis and Prognosis for Engineering Systems" and has written or co-authored more than 100 technical papers related to integrated systems health management.

Dr. Kai Goebel works at NASA Ames Research Center where he is the Area Lead for Discovery and Systems Health. He received the degree of Diplom-Ingenieur from Technische Universitaet Muenchen in 1990 and the Ph.D. from the University of California at Berkeley in 1996. Dr. Goebel worked at General Electric's Corporate Research Center in upstate New York where he was also an adjunct professor at Rensselear Polytechniq Institute. He has been on the dissertation committee of seven Ph.D. students at RPI, Syracuse University, University of Cincinnati, Vanderbilt University, Georgia Institute of Technology and Stanford University. He is currently guest professor at University of Cincinnati. He holds eighteen patents and he has published more than 250 technical papers.