

# A Testbed for Real-Time Autonomous Vehicle PHM and Contingency Management Applications

Liang Tang, Eric Hettler, Bin Zhang and Jonathan DeCastro

*Impact Technologies, LLC. Rochester, NY, 14623, U.S.A*  
*Liang.Tang@impact-tek.com*

## ABSTRACT

Autonomous unmanned vehicles are playing an increasingly important role in support of a wide variety of present and future critical missions. Due to the absence of timely pilot interaction and potential catastrophic consequence of unattended faults and failures, a real-time, onboard health and contingency management system is desired. This system would be capable of detecting and isolating faults, predicting fault progression and automatically reconfiguring the system to accommodate faults. This paper presents a robotic testbed that was developed for the purpose of developing and evaluating real-time PHM and Automated Contingency Management (ACM) techniques on autonomous vehicles. The testbed hardware is based on a Pioneer 3-AT robotic platform from Mobile Robots, Inc. and has been modified and enhanced to facilitate the simulations of select fault modes and mission-level applications. A hierarchical PHM-enabled ACM system is being developed and evaluated on the testbed to demonstrate the feasibility and benefit of using PHM information in vehicle control and mission reconfiguration. Several key software modules including a HyDE-based diagnosis reasoner, particle filtering-based prognosis server and a prognostics-enhanced mission planner are presented in this paper with illustrative experimental results. This testbed has been developed in hope of accelerating related technology development and raising the Technology Readiness Level (TRL) of emerging ACM techniques for autonomous vehicles.\*

## 1. INTRODUCTION

Autonomous unmanned vehicles (AUVs) are finding increasing use in real-world applications ranging from the ground (e.g. unmanned ground vehicles, or UGVs),

---

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

to sea surface (e.g. unmanned surface vehicles, or USVs), underwater (e.g. unmanned undersea vehicles, or UUVs), airborne (e.g. unmanned aerial vehicles, or UAVs), and space exploration missions (e.g. unmanned rovers and unmanned space vehicles). Unmanned missions using these vehicles include surveillance and patrolling, search and rescue, operations in contaminated and denied areas, space exploration and more (Army UAS CoE, 2010; Navy, 2007). Due to communication delay and bandwidth limitations, there has been increasing dependence on AUVs for critical tasks. This makes it vital to assure the performance of the vehicles under off-nominal conditions in an autonomous fashion without relying on remote operators.

In recent years, growing demand for improving the reliability and survivability of autonomous vehicles has led to the development of prognostics and health management (PHM) and automated contingency management (ACM) systems (Vachtsevanos et al, 2006). In this context, the term Automated Contingency Management has been introduced to describe intelligent systems capable of mission re-planning and control reconfiguration based on health diagnostic and prognostic information (Tang et al, 2008). As a new emerging technology, the development of real-time autonomous vehicles PHM and ACM techniques can greatly benefit from a testbed that is built on a real vehicle platform using commercial-off-the-shelf (COTS) computing devices. The use of such a testbed can accelerate the development and raise the Technology Readiness Level (TRL) of the enabling techniques, as well as provide a technology demonstrator for commercialization efforts. This paper presents the development of a ground robotic testbed for real-time autonomous vehicles PHM and ACM techniques. The testbed has been built to fulfill the following objectives:

- (1). to demonstrate the benefits of real-time PHM and ACM technologies for autonomous vehicles in terms of improved reliability, survivability and overall mission success;

(2). to demonstrate the advantages of using PHM information (particularly the prognostic information) in control reconfiguration and mission planning by applying a novel hierarchical ACM architecture;

(3). to demonstrate that real-time implementation of selected diagnostic and prognostic routines are feasible on affordable COTS computing devices;

(4). to raise the TRL of several selected techniques by deploying them on hardware and testing them on field systems in realistic environments;

The rest of this paper is organized as follows. In section 2, the system architecture and common features of a generic, hierarchical PHM-enabled ACM system for autonomous vehicles are briefly introduced. Section 3 presents the development of a robotic testbed on which the techniques described in Section 2 are applied. Hardware configuration and modifications, as well as a Failure mode, effects, and criticality analysis (FMECA) study of selected components and fault simulations are presented. Section 4 presents the real-time PHM and ACM software modules implemented on the testbed with illustrative experimental results. These modules include a diagnostic reasoner based on NASA's Hybrid Diagnostic Engine (HyDE), several particle filter-based real-time prognostic routines, and prognostics-enhanced control configuration and mission re-planning modules. The paper concludes with remarks on the main contributions of the presented work and planned future developments.

## 2. PHM-ENABLED ACM SYSTEM FOR AUTONOMOUS VEHICLES

Conceptually, an ACM system is a system that is designed to provide the ability to proactively and autonomously adapt to current and future fault and/or contingency conditions while either achieving all or an acceptable subset of the mission objectives. An ACM system is different from a fault tolerant control system mainly in two aspects: 1) it consists of not only low level control reconfiguration, but also high level (mission) planning and optimization; 2) it uses not only diagnostic information, but also prognostic information.

A typical ACM+P (PHM-enabled ACM system) implementation usually utilizes a hierarchical architecture as shown in Figure 1. The PHM and situation awareness modules provide fault diagnostics, prognostics and contingency information to the ACM+P system, which in turn, identifies and executes the optimal fault accommodation and/or mitigation strategies. Note that the PHM system is a precondition for implementing ACM strategies, thus the whole system architecture is referred to as a PHM-enabled ACM system.

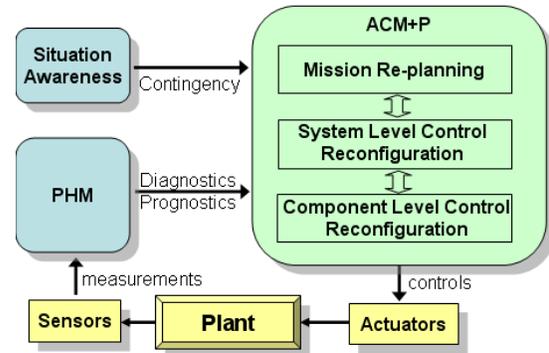


Figure 1: Conceptual PHM-enabled ACM system hierarchy

Some important features of the PHM-enabled ACM system include:

(1) Hierarchical architecture;

A component fault can often be accommodated at different levels in the ACM hierarchy and the decision should be made based on performance requirement and safety consideration. For example, if the left engine on a twin-engine, fixed-wing UAV is experiencing severe degradation, the thrust difference may generate an unwanted yaw movement. This fault can be accommodated at the lowest (component) level by adjusting the fan speed set point value in the engine controller or by adjusting the rudder position in the trajectory following auto-pilot at system level.

(2) Use of redundancy and trade-off;

Typically, it is possible to accommodate faults only in a system with redundancy, either physical redundancy or analytical redundancy. More advanced systems may include online healing concepts, including self-healing. When system performance cannot be totally recovered by the fault accommodation strategies, trade-off of mission objectives has to be made to secure the most important tasks.

(3) Online optimization;

If an ACM system is to be applied to an unmanned vehicle conducting complicated autonomous missions, it is often unavoidable to phrase the solution search as a dynamic optimization problem especially at mission planning level. This optimization problem may need to be solved online to arrive at the optimal strategies constrained by the available performance and resources to meet multiple (sometimes conflicting) mission objectives. It is important to realize that the optimization problems at different levels in an ACM+P system have different time horizons and real-time execution considerations.

(4) Uncertainty management and false alarm mitigation;

The use of prognostic information in the ACM system brings new challenges to both uncertainty management and false alarm mitigation. Since

prognosis projects the current system condition in the future using a prognostic model in the absence of future measurements, it necessarily entails large-grain uncertainty. This uncertainty has to be handled both in high level mission re-planning and middle/low level control reconfiguration modules. The implementation of this ACM architecture on the testbed and the enabling algorithms for the main functional modules are presented in section 4. More details regarding ACM technologies can be found in our previous publications (Tang et al, 2010; DeCatro et al, 2011; Zhang et al, 2011).

### 3. THE ACM TESTBED

The ACM Testbed is based on a Pioneer 3-AT robotic platform from Mobile Robots, Inc. The dimensions of the Pioneer 3-AT robot (without additional computer and sensors) are about 20" long by 19" wide by 11" high and it weighs about 26 pounds (see Figure 2). The robot is a four wheeled "skid steered" design with 8.5" diameter tubeless pneumatic rubber wheels. This means that the wheels are fixed in place and it is driven in a "tank drive" fashion. The wheels on either side of the robot are driven independently at different speeds to provide the freedom to turn. The wheels on one side of the robot are linked through timing belts and therefore always turn at the same speed. Each side is driven by two mechanically linked DC motors. The platform offers a build-in computer that hosts the baseline vehicle controller, serial communications, sonar sensors, encoders and other autonomous functions. This built-in controller uses PID control using the motor encoder signals to drive the robot at a commanded speed and calculate the robot's position using dead reckoning. It carries up to 3 hot swappable batteries. The eight forward and eight rear sonar array senses obstacles from 15 cm to 7 m. The robot can reach speeds of 0.8 meters per second and carry a payload of up to 32 kg. The robot uses 100-tick encoders which have been enhanced with inertial correction from a rate gyro measuring yaw movement for dead reckoning to compensate for skid steering.

#### 3.1 Hardware Modifications

Several modifications have been made to the robot to enhance its sensing, computing and fault simulation capabilities required for hosting the PHM and ACM functions. The major additions to the platform described in this section include the following: on-board computer and data acquisition system, batteries, load simulator, tire leakage simulator, vision system, and diagnostic/prognostic server.



Figure 2: The ACM testbed

An onboard computer is mounted on the robot and is powered by the robot's auxiliary power ports. The onboard computer is dual boot, running both Windows XP and LabVIEW RTOS. The onboard computer features a 1.40 GHz Intel Pentium CPU, 512 MB RAM, two 40GB Hitachi HDDs and communicates with the build-in controller through a serial port. It can communicate with other computers (such as a remote client laptop) on a network through a WiFi access point plugged into the computer's Ethernet port.

A NI PCI-6229 data acquisition (DAQ) card has been added to the onboard computer to monitor the health of the robot. The DAQ card has 32 analog inputs, 4 analog outputs and 48 digital I/O. With a few added circuit boards, wires and electrical components, the DAQ card monitors the current and voltage of the battery and motors while sensors are being added to monitor the air pressure of the tires.

To perform prognosis demonstrations using batteries, the 12 V sealed lead acid (SLA) batteries have been replaced with LiFePO<sub>4</sub> Li-Ion batteries. A 200-ohm, 250-watt variable resistor is used to simulate an aging battery, and can be re-wired to simulate a winding short fault in the motor.

To simulate varying loads on the drive system of the robot in the lab, which is often needed for the development and testing of PHM algorithms for batteries and motors, a dynamometer rig has been created as shown in Figure 3. The front two wheels of the robot rest on two rollers. One of the rollers is attached to a hysteresis brake, which can supply a constant braking force when a voltage is applied.



Figure 3: Simulated load test rig setup

A system for simulating tire leakage has been implemented as shown in Figure 4. On the right side of the robot, a rotary union has been connected to the hub of each tire. These rotary unions connect directly to the tire valve outlets, which have had the valves removed. Both of these unions are connected to a central manifold mounted on the top of the robot. This manifold provides ports for a pressure gauge for monitoring tire pressure, a needle valve for simulating slow tire leakage, and a Schrader valve for refilling the tires. Each tire also has its own shut off valve to allow for independent deflation. With this set up, a slow leak in the front, rear or both tires can be simulated.

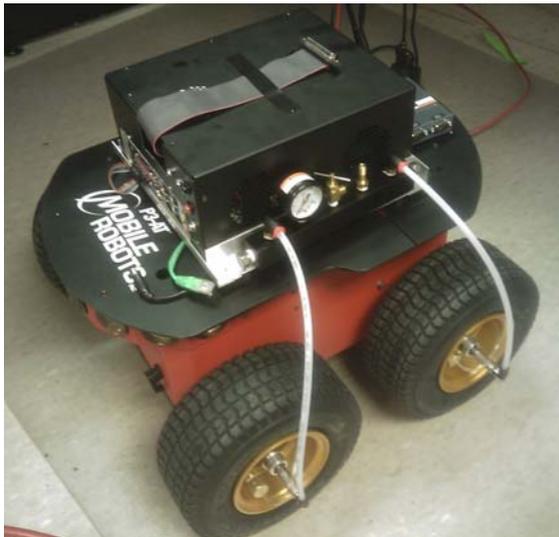


Figure 4: Tire leakage simulation system

To perform image processing, a Surveyor Stereo Vision System mounted on a Lynxmotion BPT-KT pan/tilt head is mounted on top of the onboard computer (see Figure 2). The stereo camera system is

intended for high level applications such as terrain classification, target tracking and classification.

The onboard computer acts as an autonomous server, receiving command signals from a client computer, in this case a laptop computer connected to the same network via WiFi. The laptop is a Dell Latitude D505 with a 1.60GHz Intel Pentium processor, 1.5 GB RAM, and an 80 GB HDD running Windows XP. This computer also acts as the server and image processor for the Stereo Vision System.

The server software on the robot can be configured in several different ways. To perform low-level control functions remotely, the remote client laptop can be configured to send signals that control either the speed of each individual motor, or the overall speed and angular speed of the robot. In this case, low-level sensor signals are sent to the client for processing. As an alternative, the client laptop can be configured to perform only high-level functions by simply sending waypoints to the robot. In this scenario, the onboard computer implements all lower level processing including path planning to the given waypoint, sending velocity signals to the motors, sensor signals processing, and obstacle avoidance and indoor localization using sonar array. This localization takes inputs of a pre-defined map of surroundings and information from the 16 sonar range finders to determine the accurate position of the robot which may be skewed due to the skid steering. The layout of the client software GUI can be seen in Figure 5.

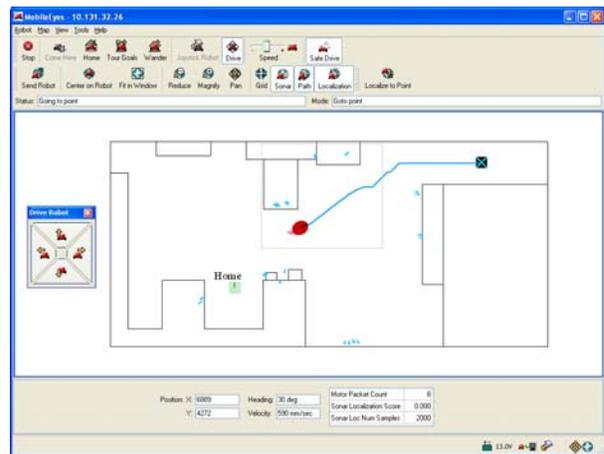


Figure 5: Client software GUI

### 3.2 FMECA Study

A Failure mode, effects, and criticality analysis (FMECA) study is conducted to identify possible faults and failure modes on the testbed platform. A subset of the failure modes, as well as their criticalities and possible diagnostic approaches and related sensor measurements, is listed below.

Table 1: Representative Failure Modes

Failure Mode	Cr.	Diag.
Motor Assembly		
Motor winding shorted/open	3	A, B, C
Motor shaft eccentricity	2	B, C
Motor bearing spall/wear	2	A, B, C, D
Seized motor bearing/shaft	4	A, B, C
Drivetrain		
Timing belt failure	4	A, B, E
Pulley degradation	2	A, B, E
Leak in tire	2	B, E, F
Tire blowout	4	B, E, F
Battery		
Battery short	3	G, H
Battery degradation	3	G, H
Sensors		
Encoder produces incorrect readings	3	B, E
Encoder produces no signal	3	B, E
Gyroscope produces incorrect readings	3	B, E
Gyroscope produces no signal	3	B, E
Cr.: criticality; Diag.: diagnostic approach and related sensors. A: Motor current; B: Encoder feedback; C: Motor accelerometer; D: Motor spindle; E: Gyroscope feedback; F: tire pressure sensor; G: Battery voltage; H: Battery Current;		

Some of the identified failure modes can be inserted or simulated on the testbed without causing permanent damage to the vehicle. These failure modes include tire leakage, tire blowout, battery short, battery degradation, incorrect encoder reading, no encoder signal, incorrect gyroscope reading, no gyroscope signal, motor wiring short, etc.

### 3.3 Load Simulations

Since battery end of charge and battery end of life are currently being tracked as part of the prognosis of the robot, it is necessary to simulate different loading scenarios due to terrain changes and other factors while the robot autonomously performs its mission. The robot is currently configured only for indoor use on 2D terrain. In an actual mission, however, the robot would experience different battery loading scenarios based on terrain. To simulate this in an indoor, 2D environment, a variable load has been attached to the battery. This variable load is made up of three resistors, each wired in parallel to the battery. Each resistor can be activated via a relay controlled by the onboard computer. It

provides 8 different loading scenarios progressing linearly in magnitude. The onboard computer has a map of simulated terrain and when the robot crosses into an area of higher simulated difficulty to traverse, the onboard computer activates a larger loading scenario using the variable load. This allows for many simulated terrains while keeping the robot in a safe, indoor environment.

### 4. IMPLEMENTATION OF PHM-ENABLED ACM SYSTEM ON THE TESTBED

The software architecture of the prototype PHM-enabled ACM system is shown in Figure 6. Starting from the bottom of this hierarchy, the DAQ Server/Monitor collects signals from various components (such as battery, motor, sonar, encoder, gyroscope, etc) and sends the observations to the Diagnosis Reasoner and Prognosis Server. Typically, the prognosis service is only activated after a fault has been detected. The diagnostic and prognostic information are sent to the ACM system where control reconfiguration and mission re-planning take place to accommodate and mitigate both present and potential future faults and failures. The ACM modules send waypoints to the Auto-pilot to adjust the mission to optimize the usage of the vehicle. Set-point commands may also be sent directly to the Vehicle Controller when lower level control reconfiguration is required. The situational awareness sensors, such as the onboard stereo vision cameras and sonars, provide obstacle, target and terrain information to the Situation Awareness module and the Auto-pilot to avoid external threats.

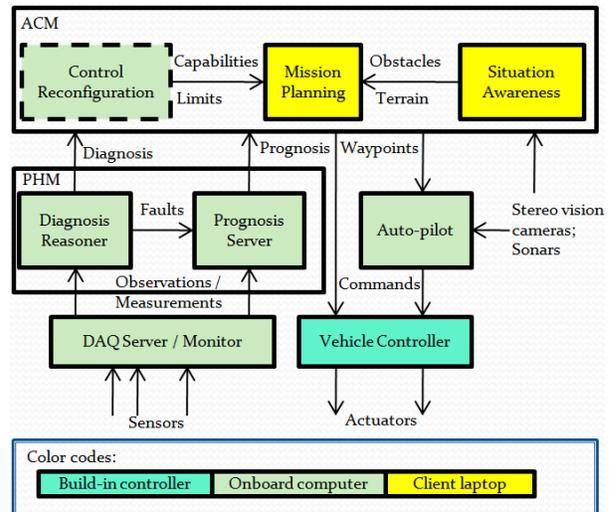


Figure 6: ACM System Software Architecture on the ACM Testbed

#### 4.1 HyDE-Based Diagnosis Reasoner

HyDE (Hybrid Diagnostic Engine) is a model-based diagnosis engine that uses candidate generation and consistency checking to diagnose discrete faults in stochastic hybrid systems. HyDE uses hybrid (combined discrete and continuous) models and sensor data from the system being diagnosed to deduce the evolution of the state of the system over time, including changes in state indicative of faults (Narasimhan and Brownston, 2007).

To demonstrate the generic applicability of HyDE diagnostic reasoning techniques to autonomous vehicles, a Diagnosis Reasoner was developed and deployed on the onboard computer on the robot testbed. The deployed reasoner is essentially a HyDE reasoner that receives sensor observation from the DAQ Server/Monitor module on the onboard computer and outputs diagnostic reasoning result. The Diagnosis Reasoner on the testbed has been developed to diagnose the following fault modes for a proof of concept demonstration.

- 1) Encoder: missing counts; lost – no output
- 2) Timing belt: slipping; failed
- 3) Rate gyro: drifting, lost – no output
- 4) Tires: leaking; deflated
- 5) Tire pressure sensors: biased reading
- 6) Sonar Sensors: erroneous reading

A part of the HyDE diagnostic model involving the encoders, timing belt, rate gyro, tire pressure sensors and the tires is shown in Figure 7.

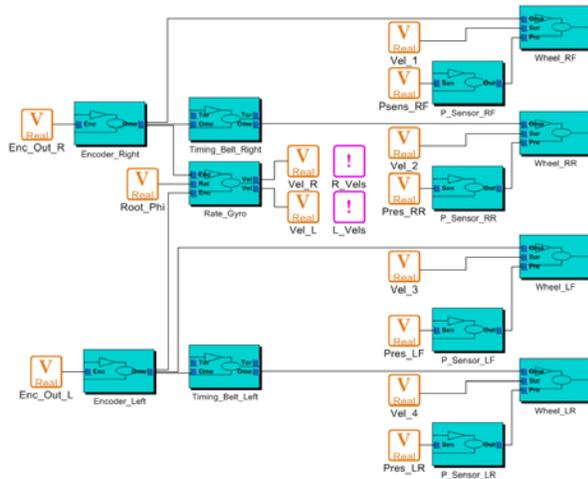


Figure 7: HyDE model for a few selected components on the testbed

To illustrate the reasoning capability of the HyDE-based solution as compared to simple logical calculation (which is often utilized in Expert System-based diagnostic systems), hybrid state reasoning has been used to determine the state of the tires. Using encoder measurements in addition to abnormal rotation

measured by the rate gyro during commanded forward movement, the diagnostic model can determine which side of the robot has a low pressure tire. With this information and the calculated logical constraints, the model can then determine whether the front or the rear tire is causing the reduced velocity. When combining this reasoning capability with tire pressure sensor reading, a pressure sensor fault can be diagnosed.

To test the model, real-time data was collected from the robot while varying both rate of movement in forward and reverse as well as individual tire pressures. Using this data, it was shown that a calculated channel could indicate an individual tire fault using accumulated significant error from the rate gyro. With the calculated channel input to the diagnostic model, it is possible to isolate an individual tire fault. The diagnosis result in a GUI is shown in Figure 8.

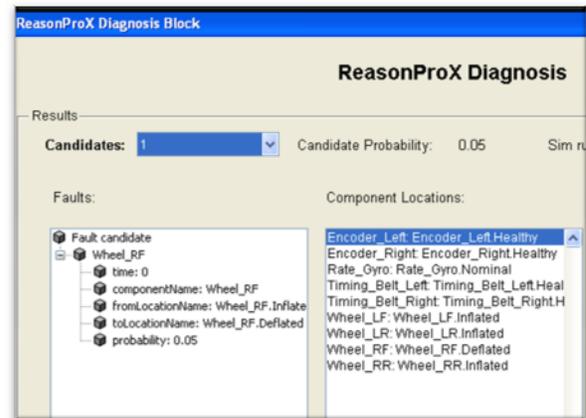


Figure 8: HyDE diagnosis given deflated right-front wheel data

#### 4.2 Particle Filtering-Based Prognosis Server

The purpose of prognosis is to predict the remaining useful life (RUL) of a system/subsystem or a component when a fault is detected. Various prognostic algorithms have been developed and applied to various mechanical and electrical systems in the past decade (Schwabacher and Goebel, 2007; Uckun et al, 2008; Saxena et al, 2010). Among these approaches, particle filtering-based approaches have been shown to be theoretically sound, generically applicable and demonstrating promising results especially on applications where online prognosis is required (Goebel et al, 2008; Saha et al, 2009). To illustrate the effectiveness and computational efficiency of real-time PF-based prognosis approach, three parallel PF-based prognostic routines have been implemented on the Prognosis Server to predict three failure modes: the RUL of the battery, state of charge of the battery and tire leakage situation. Details regarding the particle filtering-based prognosis and uncertainty management

algorithms can be found in (Orchard et al, 2010; Edwards et al, 2010).

#### 4.2.1 Battery Life Prognosis

To predict the battery end-of-life (EOL), features that reflect the aging condition of the battery such as capacity or stored energy must be tracked over time. Other features derived from electrochemical impedance spectrometry (EIS) measurement data (Goebel et al, 2008) may also be used but these features can only be obtained when onboard EIS devices are available. In this case study, the capacity of battery is used as the feature.

Because of its generic applicability and rich uncertainty management capabilities, a particle filtering-based approach is chosen. To date, three particle filtering-based prognostic approaches have been developed: i) the classic particle filtering algorithm (Orchard, 2009), ii) a routine that incorporated an Outer Feedback Correction Loop (Orchard et al., 2008), and iii) Risk Sensitive Particle Filter (RSPF) based routines (Orchard et al., 2010). The algorithm deployed on our testbed is the RSPF-based approach. Prognosis results obtained when applying the algorithm to a set of real Lithium-Ion battery data are shown in Figure 9. The capacity data measured per cycle is plotted in green while the estimated capacity which is the feature being tracked is plotted in magenta in the upper subplot. The critical capacity limit is centered around 1200 mAh as shown by the orange zone. The lower subplot shows the scaled probability density function (PDF) of the EOL predicted at cycle 100 (the peak of the PDF has been scaled to 1 for plotting purposes). In this case, the ground truth life of the battery is 168 cycles which is very close to the mean of the predicted PDF.

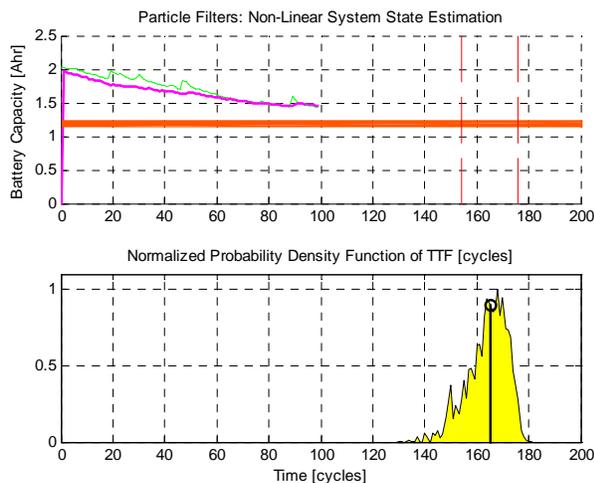


Figure 9: Battery life prognosis using RSPF-based approach

The algorithm and the implemented software module have also been tested on the battery data set provided by the Prognostics Center of Excellence at NASA Ames research center (Saha and Goebel, 2007) with comparable results. It should be noted that since the capacity data is collected per charge-discharge cycle, the software does not need to run in real-time.

#### 4.2.2 Battery End-of-Charge Prediction

In contrast to battery end-of-life prognosis which is important for the planning of long term or future missions, battery end-of-charge prognosis focuses on the prediction of battery charge state for the current mission given the health of the battery which has degraded over the course of use. When a vehicle is powered by batteries, its mission plan can be optimized in real-time if an accurate battery end-of-charge prediction capability is available on board the vehicle.

A particle-filtering based algorithm that uses a combined voltage and stored energy feature has been implemented in the Prognosis Server on the onboard computer. One of the challenges in predicting battery end-of-charge is to handle the uncertainties associated with the prognosis due to uncertain initial state of charge, ambient temperature, future load (discharge) profile and battery health, among other factors. Therefore, a particle filtering-based algorithm is chosen in this case study due to its unique uncertainty management capability and computational efficiency which enables real-time execution of highly accurate predictions.

Figure 10 below 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 signal (first subplot) 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 50 seconds. The current (as shown in second subplot) is totally determined by the load and only a trivial increase tending is observed towards the end when voltage drops. In the 3<sup>rd</sup> subplot, three end-of-charge predictions (PDFs) made at 1226, 2451 and 3677 seconds respectively are shown. Since we know the true end-of-charge time in this case (3990 seconds), it can be seen that the first and second predictions have been made conservatively due to the uncertainties associated with initial charge state and future load. The third prediction which was made when the voltage signal started to drop was rather accurate.

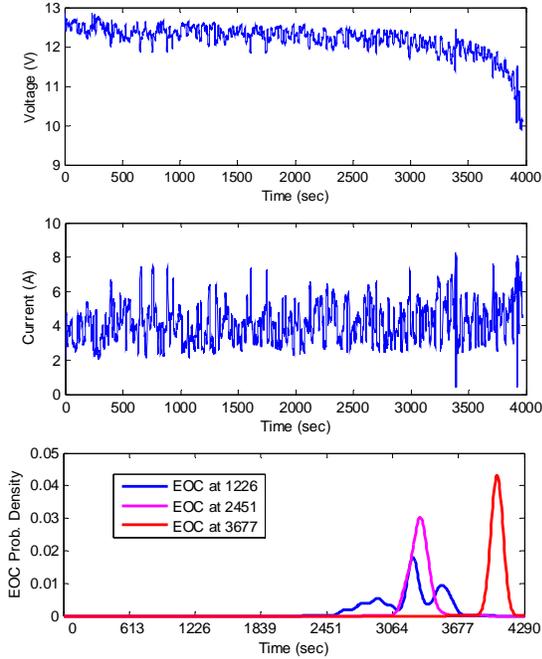


Figure 10: Battery data and end-of-charge prediction

### 4.3 Prognostics-Enhanced Mission Planning

A Mission Planning module which uses prognostic information to enhance the mission/path planning in a non-uniform environment has been implemented and deployed on the Client Laptop as shown in Figure 6. Prognostic information is introduced in order to ensure that the fault progression, or mission failure risk, can be minimized after the occurrence of a fault. This will enhance the performance of autonomous vehicles that often work in harsh environments that cause aging, fatigue, and fracture.

When a fault occurs, the mission planning module sends periodic RUL service requests to the Prognosis Server (Figure 6). The received RUL estimate (typically represented as a probability density function) is then used either as a constraint or an additional element in the cost function in the path planning algorithm, in this case, a field  $D^*$  search algorithm, in a receding horizon planning framework (Zhang et al, 2011).

In the field  $D^*$  algorithm, the map is described by grids while nodes are defined on corners of grids. The planning algorithm divides the map into three areas: implementation area, observation area, and unknown area as shown in Figure 11. The autonomous vehicle is equipped with onboard sensors that are able to detect and determine the terrain in the observation area (the area inside the magenta square in Figure 11). The implementation area (the area inside the green square in Figure 11) consists of the grid next to the current node.

The area beyond observation area is the unknown (un-observed) area where the terrain is unknown to the vehicle (the gray area in Figure 11). At a node, the vehicle plans the path from the vehicle's current location to the destination. However, only the path planned in the implementation area is executed. This is similar to the strategies used in Receding Horizon (Model Predictive) Control algorithms. This process is repeated until the destination is reached or it turns out that no route can lead to destination or the vehicle reaches its end of life. The cost function is the weighted sum of three factors and is defined as:

$$\min_{s'} J = w_T(t_o(s,s') + t_u(s')) + w_{Tr}(d_o(s,s') + d_u(s')) + w_{Pr}(p_o(s,s') + p_u(s')) \quad (1)$$

where  $t_o(s,s')$ ,  $d_o(s,s')$ , and  $p_o(s,s')$  are normalized travel time, terrain, and fault costs on the path segment from point  $s$  to point  $s'$ , respectively, while  $t_u(s')$ ,  $d_u(s')$ , and  $p_u(s')$  are those corresponding costs on path segment from point  $s'$  to final destination, respectively;  $w_T$ ,  $w_{Tr}$ , and  $w_{Pr}$  are the weighting factors on each cost factor and  $w_T + w_{Tr} + w_{Pr} = 1$ .

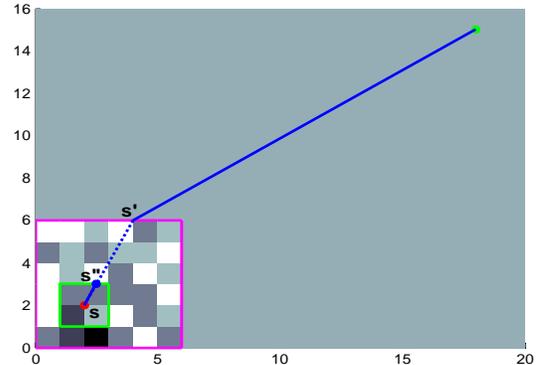


Figure 11: Receding horizon mission planning

To investigate how real-time prognostic information can be utilized in mission planning on the ACM testbed, we consider battery end-of-charge during the mission as a fault mode. In this case, the real-time battery prognosis routine described in section 4.2.2 is utilized to predict the state-of-charge of the battery. Note that battery remaining charge is a function of terrain difficulty and vehicle speed. Several laboratory experiments were conducted that illuminate the impact of battery prognostics on mission planning. Representative experimental results for four different optimality criteria are presented in Table 2. For example, it is apparent that when the mission plan is optimized for battery life (3<sup>rd</sup> column), the robot travels the longest distance and finishes the mission with the longest time, but consumes the least amount of battery life. In contrast, with the time optimal mission plan (1<sup>st</sup> column), the robot travels the shortest distance at a

higher speed, but consumes the most battery life. Both the mission planner and prognosis routine are implemented in real-time in the experiments.

Table 2: Representative Mission Planning Experiment Results for Four Different Optimality Criteria

	Optimality Criteria			
	Time	Terrain	Life	Weighted
$[w_T, w_{T_r}, w_{P_r}]$	[1,0,0]	[0,1,0]	[0,0,1]	[0.33,0.33,0.34]
Travelled distance (meters)	21.26	24.56	24.81	24.45
Travel time (seconds)	114.2	202.2	125.9	126.4
Remaining battery charge (%)	67.9	82.3	91	88.1

## 5. CONCLUSION

Real-time onboard PHM and ACM systems are needed to improve the reliability and survivability of autonomous vehicles engaged in critical missions. Research and development of enabling techniques have been conducted in recent years to achieve the required capabilities using relevant simulation programs with various levels of fidelity. This paper presents the development of a testbed that is built for the purpose of evaluating real-time vehicle PHM and ACM techniques on a real robotic platform. The testbed has been utilized to demonstrate the feasibility of a hierarchical ACM system that we have been developing over the past years highlighting the importance of using PHM information in control reconfiguration and mission planning. Several key software modules featuring real-time system level diagnosis, component fault prognosis and prognostics-enhanced mission planning have been successfully demonstrated on the testbed. Work continues on further development of the PHM-enabled control reconfiguration techniques and will eventually implement them on the testbed. More advanced situational awareness capabilities such as target tracking, localization and terrain classification will be developed so that the ACM techniques can be evaluated in more real world application scenarios. More importantly, efforts will be made to ensure the general applicability of the developed techniques to other types of autonomous vehicles such as UAVs and space exploration vehicles.

## ACKNOWLEDGMENT

This work is partly supported by the NASA under SBIR contract NNX09CB61C. The authors gratefully

acknowledge the productive collaboration with Dr. Kai Goebel, Mr. Edward Balaban, Dr. Bhaskar Saha, Dr. Abhinav Saxena, Dr. Jose Celaya and the Prognostics Center of Excellence (PCoE) at Ames Research Center. The authors also thank Dr. George Vachtsevanos, Mr. Gregory Kacprzyński, and Mr. Brooks Przybyłek from Impact Technologies, Dr. Marcos Orchard from Universidad de Chile, and Mr. Brian Bole from Georgia Institute of Technology for their contributions on the development of the testbed and related techniques.

## REFERENCES

- Army UAS Center of Excellence Fort Rucker AL (2010), *U.S. Army Roadmap for Unmanned Aircraft Systems 2010-2035: Eyes of the Army*, <http://www-rucker.army.mil/usaace/uas/US%20Army%20UAS%20RoadMap%202010%202035.pdf>
- DeCastro, J. A., Tang, L., Zhang, B. and Vachtsevanos, G. (2011), A Safety Verification Approach to Fault-Tolerant Aircraft Supervisory Control, in proceedings of *the AIAA Guidance, Navigation and Control Conference and Exhibit*.
- Edwards, D., Orchard, M., Tang, L., Goebel, K., and Vachtsevanos, G. (2010), Impact of Input Uncertainty on Failure Prognostic Algorithms: Extending the Remaining Useful Life of Nonlinear Systems, in proceedings of *the Annual Conference of the Prognostics and Health Management Society*, Portland, OR.
- Goebel, K., Saha, B., Saxena, A., Celaya, J., and Christophersen, J. (2008), Prognostics in Battery Health Management, *Instrumentation & Measurement Magazine, IEEE*, vol.11, no.4, pp.33-40.
- Navy's Program Executive Officer Littoral and Mine Warfare, Surface Warfare and Unmanned Maritime Vehicles Program Office (2007), *The Navy Unmanned Surface Vehicle (USV) Master Plan*, <http://www.navy.mil/navydata/technology/usvmprr.pdf>
- Narasimhan, S. and Brownston, L.(2007), HyDE – A General Framework for Stochastic and Hybrid Model-based Diagnosis, in proceedings of *the 18th International Workshop on Principles of Diagnosis (DX-07)*.
- Orchard, M., Kacprzyński, G., Goebel, K., Saha, B., and Vachtsevanos, G., (2008). Advances in Uncertainty Representation and Management for Particle Filtering Applied to Prognostics, in proceedings of *International Conference on Prognostics and Health Management*.
- Orchard, M., Tang, L., Saha, B., Goebel, K., and Vachtsevanos, G., (2010) Risk-Sensitive Particle-

- Filtering-based Prognosis Framework for Estimation of Remaining Useful Life in Energy Storage Devices, *Studies in Informatics and Control*, vol. 19, Issue 3, pp. 209-218.
- Orchard, M. (2009). *On-line Fault Diagnosis and Failure Prognosis Using Particle Filters. Theoretical Framework and Case Studies*, Publisher: VDM Verlag Dr. Müller Aktiengesellschaft & Co. KG, Saarbrücken, Germany.
- Saha, B., and Goebel, K. (2007). "Battery Data Set", NASA Ames Prognostics Data Repository, [<http://ti.arc.nasa.gov/project/prognostic-data-repository>], NASA Ames, Moffett Field, CA.
- Saha, B., Goebel, K., Poll, S. and Christophersen, J. (2009), Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework, *IEEE Transactions on Instrumentation and Measurement*, vol.58, no.2, pp.291-296.
- Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K. (2010), Metrics for Offline Evaluation of Prognostics Performance, *International Journal of Prognostics and Health Management*, Vol.1(1), pp. 20.
- Schwabacher, M. and Goebel, K. (2007), A Survey of Artificial Intelligence for Prognostics, in *Proceedings of AAAI Fall Symposium*, Arlington, VA.
- Tang, L., Kacprzynski, G. J., Goebel, K., Saxena, A., Saha, B., and Vachtsevanos, G. (2008). Prognostics-enhanced Automated Contingency Management for Advanced Autonomous Systems, in *Proceedings of the International Conference on Prognostics and Health Management*.
- Tang, L., Kacprzynski, G., J., Goebel, K., and Vachtsevanos, G. (2010). Case Studies for Prognostics-Enhanced Automated Contingency Management for Aircraft Systems, in *Proceedings of IEEE Aerospace Conference*.
- Uckun, S., Goebel, K., and Lucas, P. J. F. (2008), Standardizing Research Methods for Prognostics, in proceeding of *the 1st International Conference on Prognostics and Health Management (PHM08)*, Denver CO.
- Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A. & Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc.,
- Zhang, B., Tang, L., Decastro, J. A., and Goebel, K. (2011), Prognostics-enhanced Receding Horizon Mission Planning for Field Unmanned Vehicles, in proceedings of *the AIAA Guidance, Navigation and Control Conference and Exhibit*.
- Liang Tang** is an Intelligent Control and Prediction Lead at Impact Technologies. Dr. Tang's career has been focused on the development of state estimation, health management and intelligent control solutions for a wide range of military and commercial systems. His recent work has also involved developing intelligent control, data fusion and navigation technologies for various unmanned autonomous platforms. He is currently the Principal Investigator of several NASA and DoD SBIR/STTR programs developing fault tolerant control strategies for autonomous vehicles, GPS-independent inertial navigation device, diagnostic algorithms for jet engine and distributed data fusion system for USVs. He received his Ph.D. degree from Shanghai Jiao Tong University, China, in 1999. Before he joined Impact Technologies in 2003, he was a postdoctoral research fellow with the School of ECE, Georgia Institute of Technology, where he conducted research on fault diagnosis and controls of UAVs. Dr. Tang has published more than 50 papers in his areas of expertise.
- Eric Hettler** is a Project Engineer at Impact Technologies, LLC. He earned BS and MEng degrees in Mechanical Engineering from the Rochester Institute of Technology concentrating in systems modeling and control systems. Under NASA funding, his recent work has focused on developing hardware and software code in support of the mobile robotic testbed for autonomous contingency management.
- Bin Zhang** is a Senior Project Engineer at Impact Technologies with over 10 years experience in intelligent systems, modeling and simulation, fault diagnosis, prognostics and health management. He received his Ph.D. degree from Nanyang Technological University, Singapore, where his research focused on intelligent learning control, system prototyping, and applications to power electronics and robotics. Before he joined Impact Technologies, he was a research fellow with the School of ECE, Georgia Institute of Technology, Atlanta GA, where he participated in some SBIR/STTR projects and developed novel signal processing algorithms, fault progression models, and fault diagnosis and failure prognosis algorithms. Dr. Zhang is a senior member of IEEE. He serves as an Associate Editor for IEEE Transactions on Industrial Electronics and International Journal of Fuzzy Logic and Intelligent Systems. He also serves in program committee for some international conferences. He has published more than 70 papers in his areas of expertise.
- Jonathan DeCastro** is Lead Engineer at Impact Technologies, LLC with over ten years of experience developing and maturing advanced control techniques,

high-fidelity control evaluation simulations, and control allocation/re-allocation for fault-tolerant and safety-critical aircraft propulsion systems. As a research scientist at the NASA Glenn Research Center, Mr. DeCastro had developed the commercial modular aero-propulsion system simulation (C-MAPSS) aircraft propulsion model and advanced control system, which has since become a NASA software product. His previous research accomplishments have centered on the development of development of advanced controls

technologies for aircraft, including receding-horizon control, fault-tolerant control, and distributed control. Mr. DeCastro earned the BS and MS degrees in Mechanical Engineering from Virginia Tech. He is author of over 20 publications and is the recipient of an AIAA Section Award for Best Young Professional Paper, a NASA Group Achievement Award for leadership in developing C-MAPSS, and two NASA Space Act Awards.