

A Particle Filtering-based Framework for Real-time Fault Diagnosis of Autonomous Vehicles

Ioannis A. Raptis¹, Christopher Sconyers², Rodney Martin³, Robert Mah⁴,
Nikunj Oza⁵, Dimitris Mavris⁶, and George J. Vachtsevanos⁷

¹ *University of Massachusetts Lowell, Lowell, Massachusetts, 01854, USA*
Ioannis.Raptis@uml.edu

^{2,6,7} *Georgia Institute of Technology, Atlanta, Georgia, 30332, USA*
cskonyers@gatech.edu
dimitri.mavris@aerospace.gatech.edu
george.vachtsevanos@ece.gatech.edu

^{3,4,5} *NASA Ames Research Center, Moffett Field, California, 94035, USA*
Rodney.martin@nasa.gov
Robert.w.mah@nasa.gov
Nikunj.c.oza@nasa.gov

ABSTRACT

Reliability and supervision of autonomous systems and their corresponding subsystems, can be significantly improved using advanced methods of anomaly detection and fault diagnosis. A reliable fault detection module can improve the autonomy of the vehicle itself, by leading to efficient fault tolerant control designs and mission scheduling. This paper presents a fault detection framework for incipient faults that take place on the actuators of an autonomous-small scale hovercraft. The enabling technologies borrow from the fields of modeling, data processing, Bayesian estimation theory and in particular a technique called particle filtering. Condition indicators or features are derived based on first principles modeling of the actuator's and the vehicle's dynamics. The fault detection algorithm fuses information obtained from different subsystems of the vehicle that use distinct sets of sensors, producing a robust degree of confidence. In addition, the algorithm is decoupled from the control system. This achieves the goal of minimizing the fault masking, or the compensation of fault-induced navigational errors by the control system, and allows for early detection even for small fault conditions. The efficacy of the diagnostic approach is demonstrated via simulation results. The proposed fault detection methodology can be easily leveraged to other types of autonomous vehicles.

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

1. INTRODUCTION

Autonomous vehicles have seen unprecedented levels of growth over the last decade. Although autonomous vehicles have been mainly used for military applications, there is a considerable and increasing interest for civilian applications as well. It is not an exaggeration to consider that as the technology matures and as autonomous vehicles become cost-effective with proven reliability and safety, civilian applications will dominate the field.

The practical motivation for pursuing this objective stems from the need to address an area of concern in the aviation industry that relates to power asymmetry resulting from a slow power loss, stuck throttle, or no response to throttle coupled with automatic controls. Flying aids, such as the auto-pilot and auto-throttle, can mask significant power asymmetry until a control limit is reached. At this point, the flight crew has to intervene, understand the malfunction, and assume control of an airplane which may be in an upset condition. Better indications and/or annunciations of power asymmetry could warn crews in advance and allow them time to identify the problem and apply the appropriate procedures (Sallee & Gibbons, 1998). Although this problem has in part been solved for a newer generation model B-787 capable of providing the pilot proprioceptive feedback based upon the Improved Thrust Asymmetry Compensation System (ITACS), older airplanes without ITACS are still going to remain in use for quite a long time so this scenario is still relevant (<http://aviationknowledge.wikidot.com/aviation:ergonomics->

of-cae-7000-series-simulator-configured, n.d.). Thus, emulation of an asymmetric thrust event that causes a gradual drift of important system level safety parameters that go undetected by an operator/pilot due to compensation by a robust control algorithm is of great importance.

Specific to the issues of automatic controls masking the effects of degradations in thrust, there have been at least two accidents in recent years relating to Thrust Management Systems (TMS) effects. In both scenarios, data from the ensuing investigations have provided evidence that it is incorrect to assume that the flight crew will always detect and address potentially adverse TMS effects strictly from inherent operational cues. On March 31, 1995 in Bucharest, Hungary a Tarom Airbus Model A310-300 airplane crashed shortly after takeoff. The Romanian investigating team indicated that the probable cause of the accident was the combination of an auto throttle failure that generated asymmetric thrust and the pilot's apparent failure to react quickly enough to the developing emergency. On November 24, 1992 in Guilin, China, a China Southern Boeing Model 737-300 airplane crashed shortly before landing at Guilin. The Civil Aviation Administration of China team investigating the probable cause of the accident concluded that the right auto throttle did not react during descent and level off. As a result, the thrust asymmetry induced the airplane to roll to the right. The flight crew failed to recognize the abnormality and make correction in time, "followed by wrongful control input and crashed (Statement, 2022)." These two examples provide sufficient and defensible evidence that there is a pragmatic, safety-critical need for the results of the research problem to be investigated in this paper. Thus, it is important to recognize that the research efforts described in this paper are not purely academic in nature or purely as an exercise to demonstrate the utility of a particular approach.

Although most attention is given to ground and aerial vehicles, autonomous surface vessels are an emerging research topic. A special representative of the surface vessels family is the hovercraft. The main characteristic of the hovercraft is its ability to travel on different types of surfaces. This is achieved by sustaining a cushion between the hull of the vehicle and the surface with low-pressure air. The vehicle is moving almost frictionlessly since an air-lubricated layer is created between the cushion and the ground surface.

The main component that is responsible for providing the vehicle with autonomy during the execution of a mission, is the control system. An important requirement of advanced control systems is reliability. Reliability requirements necessitate that the control system is tolerant to incipient faults that affect the system. A key component of a fault tolerant control system is the anomaly detector. An anomaly detector is a module of the Prognostics and Health Management system intended to recognize as early as possible abnormal condi-

tions or anomalies in the operation of a monitored system. In most real applications, an anomaly detector is required to perform this task while minimizing both the probability of false alarms and the detection time (time between the initiation of a fault and its detection), given a fixed threshold for false positives (Orchard, 2007; Zhang et al., 2011). In the case of multiple faults or alternate fault types, it is assumed that the fault signature indicating the fault of interest, a change in a single fan motor resistance value, is distinct from other fault signatures.

A block diagram illustrating the complete architecture that involves the control system with the detection algorithm can be seen in Figure 1. Additionally, Figure 2 shows the inner motor control loops that regulate fan speed and stabilize the thrust output for each motor. The combination of these inner control loops and the outer navigation control is what primarily masks the fault. Autonomous vehicles are typically large-scale systems composed of several subsystems. Each subsystem has its own set of sensors with varying accuracies. The proposed fault detection algorithm is capable of merging information from different subsystems of the autonomous vehicle. Finally, the detection scheme can be easily applied to other autonomous vehicle with different actuating mechanisms.

This paper presents the implementation of an online particle-filtering-based framework for fault diagnosis in a autonomous robotic hovercraft. The hovercraft under consideration is actuated by two independent thrusters. The fault to be detected is the increase to the effective resistance value of the motor that operates one of the thrusters. This fault is a gradual degradation of component health within the thruster motor and is chosen to allow the diagnostic tools time to analyze the fault and any available reconfigurable control scheme to adapt to the fault beyond the scope of immediate catastrophic failure. This resistance value is internal to the motor and can not be easily measured during operation of the hovercraft. A variety of components ranging from sensors and actuators to communication, computing, and control apparatus may experience a severe fault mode. We are focusing in this study on a thruster fault mode, i.e. driving motor insulation breakdown, which limits the operational capability of one of the two thrusters, resulting in a deviation of the vehicle from its designated trajectory. Moreover, the selected fault mode is masked by control loops and can be easily simulated or injected in the actual vehicle. The faulty thruster is producing a reduced thrust value compared to its healthy performance. The methodology considers a diagnostic model, and assumes the existence of fault indicators (for monitoring purposes) and the availability of real-time measurements. The fault detection module uses a hybrid state-space model of the system and a particle filtering algorithm to calculate the probability of detection of the fault, simultaneously computing the fault probability density function (PDF) estimates. The enabling

technologies borrow from the fields of modeling, data processing, Bayesian estimation theory and in particular a technique called particle filtering. Once this information is available within the fault detection module, it is conveniently processed to generate proper fault alarms and to inform about the statistical confidence of the detection routine. These statistical outputs render the proposed algorithm significantly superior to classical approaches such as limit or trend checking that do not give a deeper insight into the fault diagnosis.

The fault detection algorithm can be executed on-line and in parallel with the control system. The decoupling of the fault detection routine from the control system allows an early detection, preventing the control system from “masking” the fault. By “masking” we express the suppression of the deviant system behavior—caused by the fault growth—from the control system (even for small fault levels). Therefore, the effects of the fault can be efficiently distinguished from external disturbances that act on the vehicle.

This paper is organized as follows: Section 2 presents the hovercraft motion dynamics and the model of the thrusters’ motors. In Section 3 the generic description of the fault growth models is given. The description of the features (or condition indicators), used in the detection algorithm, are given in Section 4. The technical approach of the detection algorithm is presented in Section 5. Results in the form of numerical simulations are given in Section 6. Finally, concluding remarks are given in Section 7.

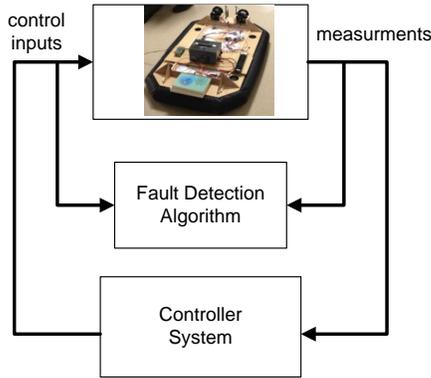


Figure 1. Block diagram illustrating the architecture of the control system enhanced with the proposed diagnostic algorithm.

2. HOVERCRAFT MODEL

The hovercraft under consideration is actuated by two independent uni-directional thrusters that are symmetrically located with respect to the median plane of the vehicle. This configuration is generating the hovercraft’s surge force and yaw moment. Since there is no direct control input applied to the sway motion the hovercraft is classified as an underactuated system. The input of each thruster is a voltage signal that

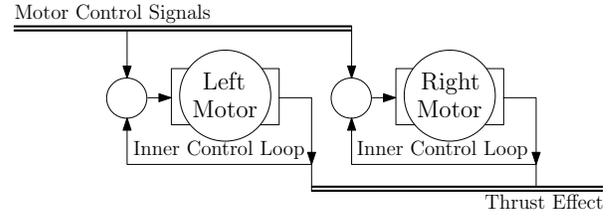


Figure 2. Inner motor control loops responsible for motor fault masking.

controls an electrical motor. The motor speed is operating the thruster’s propeller that generates the propulsion force. The hovercraft model can be divided into two subsystems. The first subsystem is related with the force and moment generation process. The second subsystem is associated with the hovercraft’s motion dynamics. The two subsystems and their connections are shown in Figure 3.

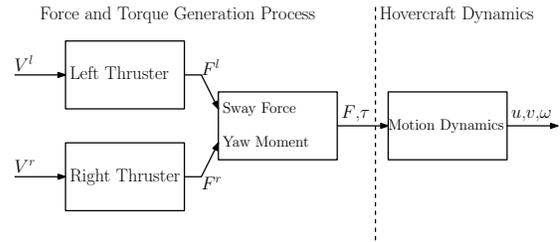


Figure 3. Hovercraft model subsystems

2.1. Hovercraft Dynamics

The first step towards the development of the hovercraft’s equations of motion is the definition of two reference frames. Each frame is characterized by its center and three mutually orthonormal vectors. The first one is the inertial frame defined as $\mathcal{F}_I = \{O_I, \vec{i}_I, \vec{j}_I, \vec{k}_I\}$. A typical convention of the inertial frame is the North-East-Down system where \vec{i}_I points North, \vec{j}_I points East and \vec{k}_I points at the center of the Earth. The second frame is the body-fixed reference frame defined as $\mathcal{F}_B = \{O_B, \vec{i}_B, \vec{j}_B, \vec{k}_B\}$, where the center O_B is located at the Center of Gravity (CG) of the hovercraft. The vector \vec{i}_B points forward, \vec{j}_B points at the aft right side of the hovercraft and \vec{k}_B points downward such that $\{\vec{i}_B, \vec{j}_B, \vec{k}_B\}$ constitutes a right handed Cartesian coordinate frame ($\vec{k}_B = \vec{i}_B \times \vec{j}_B$). The direction of the body fixed frame orthonormal vectors $\{\vec{i}_B, \vec{j}_B, \vec{k}_B\}$ is shown in Figure 4.

We consider only the planar 2-D motion of the vehicle disregarding the pitch, roll, and heave motion. Denote the hovercraft angular velocity by ω and the surge (forward) and sway (lateral) velocities by u and v , respectively. From Figure 4, the hovercraft dynamic equations, with respect to the body

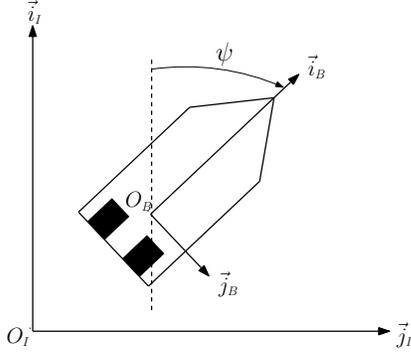


Figure 4. Direction of the hovercraft body fixed frame orthonormal vectors.

fixed frame, can be derived as

$$\begin{aligned} \dot{u} &= v\omega + F/m + n \\ \dot{v} &= -u\omega + n \\ \dot{\omega} &= \tau/\mathcal{I} \end{aligned} \quad (1)$$

where F denotes the net surge force, τ the net yaw moment, m is the mass of the hovercraft, \mathcal{I} is the inertia of the hovercraft (assuming symmetry with respect to the principal axis), and n is additive noise due to disturbances.

2.2. Force and Torque Generation

Let f_l , f_r denote the thrust force of the left and right fan respectively. The association of the net surge force and yaw moment with the thrusts of the two fans are given by a linear one-to-one mapping, namely

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ d & -d \end{bmatrix} \begin{bmatrix} f_r \\ f_l \end{bmatrix} \quad (2)$$

where d is a parameter related to the moment arm of each force. The two propulsion thrusts are produced by two identical fans that are operated by common motors. The last step of the modeling process is to include a simplified model of these motors. A reasonable assumption is to consider the motor dynamics significantly faster compared to the motion dynamics of the vehicle. Therefore the motors can be represented by a static instead of a dynamic model. In this section we are going to present the generic model of each motor. The distinction between the left and right motor will be take place by using the lower scripts l and r in each case. Denote by V the voltage applied to the fan motor. This voltage is the output from the control system. From standard results, the electrical part of the motor is described by the following equation:

$$V = I_m R_m + V_{emf} \quad (3)$$

where I_m is the motor current, R_m is the effective internal

motor resistance and V_{emf} is the back-emf voltage of the motor. The back emf voltage is given by

$$V_{emf} = K_\Omega \Omega \quad (4)$$

where Ω is the angular velocity of the motor and K_Ω is a positive constant. From the mechanical part of the motor, the produced torque is given by

$$T = K_t I_m - b\Omega \quad (5)$$

where T is the produced torque, b is the friction coefficient of the motor and K_t a positive constant. Finally, we assume that the produced thrust is proportional to the angular velocity Ω . Namely, we assume a linear relationship at the desired operating conditions;

$$F = K_f \Omega \quad (6)$$

where K_f is a positive constant. K_t and K_f are identified as part of the system ID in Section 2.3. From (3)-(6) the input-output description of the thrusters is given by

$$F = \Pi(R_m) V \quad (7)$$

where

$$\Pi(R_m) = \frac{K_t K_f}{b R_m + K_t K_\Omega} \quad (8)$$

The available measurements are all the states related to the motion of the vehicle (u , v , ω), the applied voltages to each motor (V^l , V^r) and the produced currents (I_m^l , I_m^r). Since the produced current of each motor is considered a measured quantity, the current-voltage mapping is required by the FDI approach. Therefore,

$$I_m = \frac{b}{b R_m + K_t K_\Omega} V \quad (9)$$

2.3. Simulation-Hardware Matching and System ID

A collaboration software framework is adapted for simulation and visualization studies. The vehicle dynamics are simulated in Gazebo. Rviz is used for the visualization of the vehicle's motion/environment accompanied by sensor/navigation relevant information. ROS is the operating system running the vehicle's software routines. The selected simulation/visualization environment facilitates troubleshooting and debugging of the control algorithms before installation on the actual platform, exploits realistic sensor and actuator models exhibiting the limitations of available measurements. An attempt is made to cope as accurately as possible with uncer-

tainty observed on the actual vehicle.

A dynamic environment overview of the ROS/Gazebo simulator is shown in Figure 5.

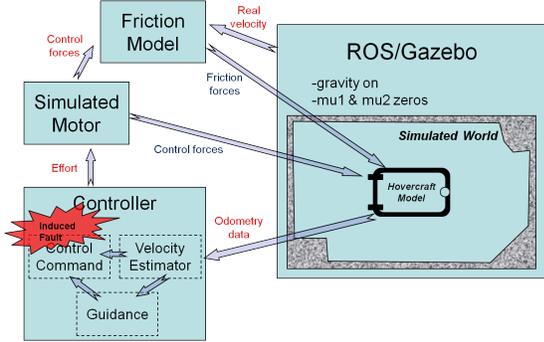


Figure 5. Dynamic environment overview of the ROS/Gazebo simulator.

The Gazebo hovercraft model incorporates component weights as measured individually. The friction model presents significant difficulties because of the inherent uncertainty associated with a moving hovercraft—the skirt/environment interface. Translational coefficients are estimated from strain gauge measurements while damping coefficients are tuned on the basis of hardware test data.

Hovercraft tests focused on way-point following (five way points) with a line-of-sight guidance law and a dynamic inversion control scheme. Several forward maneuvers were tested with a varying effect for several seconds. Acceleration, deceleration, and velocity rates were recorded. Simulation model parameters were tuned using a mixture of system ID and parameter optimization tools to maximize agreement between the simulation model and the hovercraft behavior.

Simulation versus test results are shown in Figure 6. Distance and velocity profiles for various percentage efforts are compared between simulated and experimental results. Good agreement is observed between the two cases.

Finally, 7 shows a comparison of test and simulation results in the way point trajectory following controller, using a five point trajectory that generates response for each unique hovercraft maneuver (left turn, right turn, straight). Discrepancies are attributed to parameter (such as friction) uncertainties, disturbances impacting the vehicle, and other unmodeled vehicle dynamics and environmental effects. As more test data is accumulated, the parameters are fine-tuned to better match experimental and simulation results.

3. FAULT MODEL

The fault under consideration is increase of the effective internal resistance value of one of the motors, a value that cannot be measured directly without dismantling the motor. Ad-

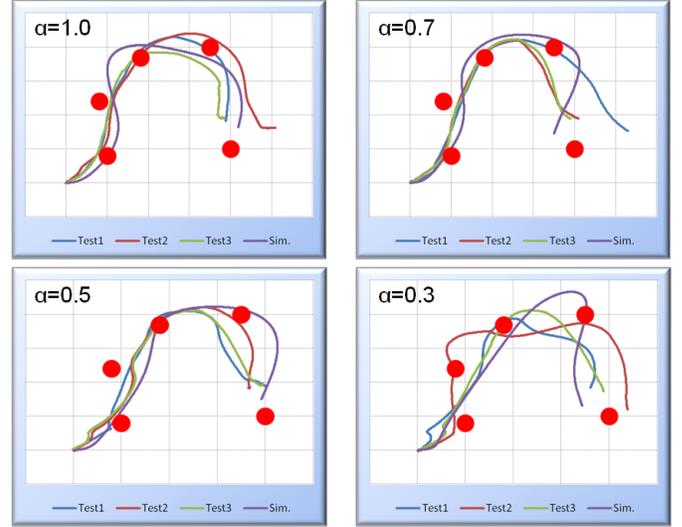


Figure 7. Example of trajectory following tests, simulation versus test results. α is a tuned parameter representing friction.

ditional simultaneous faults, including the case of a similar fault occurring on both motors, are not considered in this paper. The proposed analysis involves faults that are monotonically increasing functions of the load conditions. In this case study the load variable is the faulty motor current I_m . Therefore, the generic growth rate of the fault under consideration is given by the following differential equation:

$$\dot{R}_m^f = g(t, R_m^f, I_m) \quad (10)$$

with $R_m^f(t_f) = R_m$. By t_f we denote the time instant that the fault occurs, while R_m^f is the value of the faulty resistance. Furthermore, $g(t, R_m^f, I_m) > 0 \forall t, R_m^f, I_m \in \mathbb{R}_+$. The latter condition guarantees that the fault value is non-decreasing over time. Hence, the faulty resistance can be written as $R_m^f(t) = R_m + \Delta R_m^f(t)$, where R_m is the healthy value of the resistance and $\Delta R_m^f(t) \geq 0 \quad \forall t \geq t_f$.

4. FEATURE EXTRACTION

Feature or condition indicator selection and extraction constitutes the cornerstone for accurate and reliable fault diagnosis. A feature or condition indicator is an extracted value from a signal that describes the status of the process that fault diagnosis is applied to. Fault diagnosis depends mainly on extracting a set of features from sensor data that can distinguish between fault classes of interest, detect and isolate a particular fault at its early initiation stages. Feature evaluation and selection metrics include the monotonicity of the relationship between the feature and the true fault size and the variance (or covariance) of the feature at discrete fault levels compared to the feature range. A feature is sufficient if it shows a similar growth pattern to that of the ground truth data.

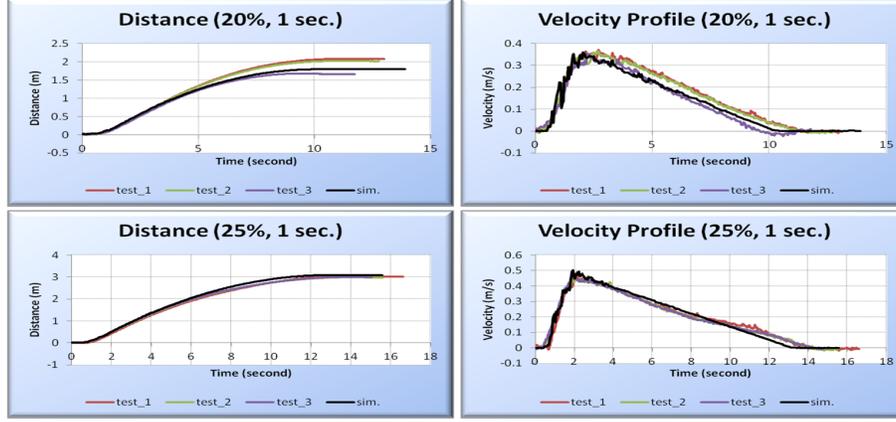


Figure 6. Hovercraft performance, simulated versus test results.

With the possibility of alternate fault types or of multiple simultaneous faults, it is assumed that a feature or set of features may be designed to identify only the fault of interest. Alternatively, a single feature may accommodate a set of similar fault types, such as a set of faults that have similar characteristics and effects at the component or system levels. For this paper, only one fault is being examined.

As indicated earlier the fault under consideration is the change in the resistance value of one of the two motors. From Section 2, the hovercraft model is composed by two interconnected subsystems: the force/moment generation and the motion dynamics subsystems. The faulty resistance affects the force/moment generation directly, and sequentially the vehicle's motion. The goal of this paper is to use features extracted from signals generated by both of these subsystems. The first feature, that belongs to the force/moment generation subsystem, is the resistance value itself. In particular, rearranging (9) one has

$$f_{em}(R_m^f) = R_m^f = \frac{V}{I_m} - \frac{K_\omega K_t}{b} \quad (11)$$

The second feature is derived by the motion dynamic subsystems. Both of the two features presented are based on the dynamic equation of motion of the vehicle given in (1). Combining (1), (2) and (7) the dynamics of the surge velocity can be written as

$$\dot{u} = v\omega + (\Pi(R_m^l)V^l + \Pi(R_m^r)V^r) / m \quad (12)$$

Assuming that we monitor the left motor for a fault ($R_m^f(0) = R_m = R_m^l$) and considering the above equation, the second feature is

$$f_{cr1}(R_m^f) = \Pi(R_m^f) = \frac{m(\dot{u} - v\omega) - \Pi(R_m^r)V^r}{V^l} \quad (13)$$

The second feature is the mapping from the voltage-to-thrust. This feature is valid only when $V^l \neq 0$ indicating the intuitive notion that the faulty motor must be operating in order to diagnose the fault. Similarly the dynamics of the angular motion are given by

$$\dot{\omega} = \frac{d}{I} (\Pi(R_m^l)V^l - \Pi(R_m^r)V^r) \quad (14)$$

The voltage-to-thrust mapping can be also derived by the angular motion as well. More specifically,

$$f_{cr2}(R_m^f) = \Pi(R_m^f) = \frac{(I/d)\dot{\omega} + \Pi(R_m^r)V^r}{V^l} \quad (15)$$

The features f_{cr1} and f_{cr2} are monitoring the same quantity. However, in each case different sensors are used. The goal of this paper is to conduct fault diagnosis by monitoring two subsystems of the vehicle. Therefore, from a theoretical perspective, either the tuple (f_{em}, f_{cr1}) or (f_{em}, f_{cr2}) can be used by the detection algorithm. Typically, for the sensing of the vehicle's motion an Inertial Measurement Unit (IMU) is used. In such cases, it is preferable to monitor motion variables related to the angular motion of the vehicle since they typically have better accuracy compared to variables related to the linear motion and are less affected by disturbances and varying environment conditions. The Advanced Systems Design Laboratory at the Georgia Institute of Technology has assembled a small-scale autonomous hovercraft platform in order to serve as a testbed for the evaluation of the algorithm. For indoor operation of the vehicle, the variables related with the motion of the vehicle are obtained by a LIDAR sensor using SLAM. Measurements from the GPS and IMU can be taken and combined with the LIDAR measurements for improved accuracy. The measurements related with the thrust generation are obtained by the PC unit onboard the vehicle. The autonomous hovercraft can be seen in Figure 8.

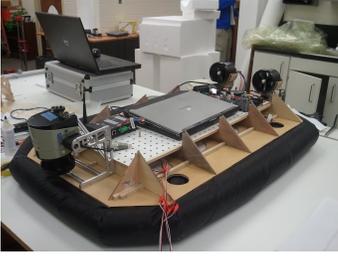


Figure 8. This figure illustrates the small-scale autonomous hovercraft developed by the Advanced Systems Design Laboratory at the Georgia Institute of Technology.

5. THE DIAGNOSTIC ALGORITHM

A fault diagnosis procedure involves the tasks of fault detection and isolation (FDI), and fault identification (assessment of the severity of the fault). In general, this procedure may be interpreted as the fusion and utilization of the information present in a feature vector (measurements), with the objective of determining the operating condition (state) of a system and the causes for deviations from particularly desired behavioral patterns. Several ways to categorize FDI techniques can be found in literature. FDI techniques are classified according to the way that data is used to describe the behavior of the system: data-driven or model-based approaches.

Data-driven FDI techniques usually rely on signal processing and knowledge-based methodologies to extract the information hidden in the feature vector (also referred to as measurements). In this case, the classification/prediction procedure may be performed on the basis of variables that have little (or sometimes completely lack of) physical meaning. On the other hand, model-based techniques, as the name implies, use a description of a system (models based on first principles or physical laws) to determine the current operating condition.

A compromise between both classes of FDI techniques is often needed when dealing with complex nonlinear systems, given the difficulty of collecting useful faulty data (a critical aspect in any data-driven FDI approach) and the expertise needed to build a reliable model of the monitored system (a key issue in a model-based FDI approach).

From a nonlinear Bayesian state estimation standpoint, this compromise between data-driven and model-based techniques may be accomplished by the use of a Particle Filter (PF) based module built upon the dynamic state model describing the time progression or evolution of the fault (Orchard, 2007; Orchard & Vachtsevanos, 2007, 2009). The fault progression is often nonlinear and, consequently, the model should be nonlinear as well. Thus, the diagnostic model is described by:

$$\begin{aligned}x_d(t+1) &= f_d(x_d(t), n(t)) \\x_c(t+1) &= f_t(x_d(t), x_c(t), \omega(t)) \\y(t) &= h_t(x_d(t), x_c(t), v(t))\end{aligned}$$

where f_d , f_t , and h_t are nonlinear mappings, $x_d(t)$ is a collection of Boolean states associated with the presence of a particular operating condition in the system (normal operation and multiple fault types), $x_c(t)$ is a set of continuous-valued states that describe the evolution of the system given those operating conditions, $y(t)$ denotes the available measurements, $\omega(t)$ and $v(t)$ are non-Gaussian distributions that characterize the process and feature noise signals including disturbances, respectively. Since the noise signal $n(t)$ is a measure of uncertainty associated with Boolean states, it is advantageous to define its probability density through a random variable with bounded domain. For simplicity, $n(t)$ may be assumed to be additive uniform white noise (Orchard, 2007). The PF approach using the above model allows statistical characterization of both Boolean and continuous-valued states, as new feature data (measurements) are received. As a result, at any given instant of time, this framework provides an estimate of the probability densities associated with each fault mode, as well as a PDF estimate for meaningful physical variables in the system. Hypothesis testing through calculating current and baseline PDFs is used to generate fault alarms, and other statistical analysis tools may be used to extract additional information about the detection and diagnostic results (discussed further in Section 6). One particular advantage of the proposed particle filtering approach is the ability to characterize the evolution in time of the above mentioned nonlinear model through modification of the probability masses associated with each particle, as new data from fault indicators are received.

The PF based FDI module is implemented accordingly using the non-linear time growth model given in (10) to describe the faulty motor's resistance value. The goal is for the algorithm to make an early detection of the increase to the effective resistance value (leading to an open-circuit). Two main operating conditions are distinguished: The normal condition reflects the fact that there is no fault in the motor while a faulty condition indicating an unexpected growth to the resistance value. Denote by $x_{d,1}$ and $x_{d,2}$ two Boolean states that indicate normal and faulty conditions respectively. The nonlinear model is given by:

$$\begin{aligned}\begin{bmatrix} x_{d,1}(t+1) \\ x_{d,2}(t+1) \end{bmatrix} &= f_b \left(\begin{bmatrix} x_{d,1}(t) \\ x_{d,2}(t) \end{bmatrix} + n(t) \right) \\ \dot{R}_m^f(t) &= x_{d,2}(t)g(t, R_m^f, I_m) + \omega(t) \\ y(t) &= h(R_m^f(t)) + v(t)\end{aligned}\quad (16)$$

where

$$f_b(x) = \begin{cases} [1 & 0]^T & \text{if } \|x - [1 & 0]^T\| \leq \|x - [0 & 1]^T\| \\ [0 & 1]^T & \text{else} \end{cases} \quad (17)$$

$$\begin{aligned} [x_{d,1}(t_o) \quad x_{d,2}(t_o) \quad R_m^f(t_o)]^T &= [1 \quad 0 \quad R_m]^T \\ h(R_m^f(t)) &= [f_{em}(R_m^f) \quad f_{cr1}(R_m^f)]^T \end{aligned}$$

In the above equations R_m is the initial healthy value of the resistance. In (17), the condition indicators $x_{d,1}(t)$ and $x_{d,2}(t)$, after the addition of noise $n(t)$, are thresholded to restrict them to Boolean values, with the possibility of changing to new values at $t + 1$. The above system can be written in a more compact form as:

$$\mathcal{X}(t+1) = \Phi(t, \mathcal{X}(t), \mathcal{N}(t+1)) \quad (18)$$

$$\mathcal{Y}(t) = H(\mathcal{X}(t), t) + v(t) \quad (19)$$

where $\mathcal{X} = [x_{d,1} \quad x_{d,2} \quad R_m^f]^T$, $\mathcal{Y} = [f_{em} \quad f_{cr1}]^T$ and $\mathcal{V} = [n \quad \omega]^T$. The steps of the PF algorithm execution are described below:

1. From (18) generate N state estimates (particles) denoted by $\hat{\mathcal{X}}^i(t)$ where $i = 1, \dots, N$.
2. From (19) calculate the feature estimates, substituting the particles $\hat{\mathcal{X}}^i(t+1)$ to the mapping $H(\cdot)$.
3. Calculate the N errors $\varepsilon^i = H(\hat{\mathcal{X}}^i(t), t) - \mathcal{Y}$ with $\varepsilon^i = [\varepsilon_1^i \quad \varepsilon_2^i]^T$, and assign to each particle $\hat{\mathcal{X}}^i(t)$ a weight $w^i(t) = \phi(\varepsilon_1^i) \phi(\varepsilon_2^i)$, where $\phi(\cdot)$ denotes the standard normal distribution.
4. Normalize the weights $w^i(t)$. The normalized weights $\bar{w}^i(t)$ represent the discrete probability masses of each state estimate.
5. Calculate the final state estimate $\tilde{\mathcal{X}}(t)$ using the weighted sum of all the states $\hat{\mathcal{X}}^i(t)$.

An important part of the PF algorithm is the resampling procedure. Resampling is an action that takes place to counteract the degeneracy of the particles caused by estimates that have very low weights. A block diagram of the PF algorithm is given in Figure 9.

6. SIMULATION RESULTS

The performance of the proposed FDI algorithm was tested via numerical simulations. The hovercraft dynamics are described in (1) and the thrusters model in (3)-(7). The resistance fault is seeded to the left motor according to (10). The actual fault can be seen in Figure 10. The vehicles parameters

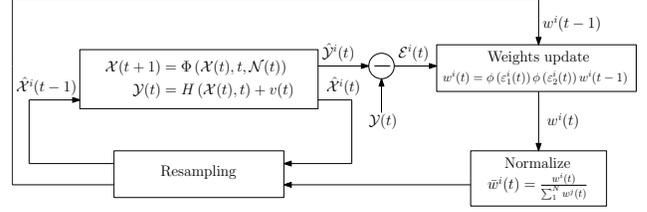


Figure 9. Block diagram of the PF algorithm for state estimation

used for the simulation are summarized in Table 1. In real-life applications these parameters can be obtained by the thruster specifications and simple system identification tests.

Test experiments were designed to closely mimic a fault continuum over time. Initially, a fault is not present in the system for the purposes of establishing a baseline. At $t = 100sec$, the fault is seeded with an initial value close to zero and grows at a constant rate until, at $t = 200sec$, the fault growth ceases and the vehicle behavior and feature characteristics can be analyzed at a static fault level.

With the desired vehicle trajectory given, position and orientation of the hovercraft is extracted from a simulated LIDAR/SLAM visual odometry sensor. Motor control signals are extracted from the motor control software.

The number of particles used for the estimator were $N = 100$. The estimator fault value can be seen in Figure 10. Although the diagnostic particle filter retains an accurate estimate of the fault after the maximum fault level is reached ($t \geq 200sec$), the upper and lower feature bounds exhibit large spikes as the control algorithm fails to compensate for the motor fault during vehicle maneuvers.

Besides detecting the faulty condition, it is desired to obtain some measure of the statistical confidence of the alarm signal. For this reason, an additional output will be extracted from the FDI module. This output is the statistical confidence needed to declare the fault via hypothesis testing (H_0 : The motor is healthy vs H_1 : The motor is faulty). The latter output needs another PDF to be considered as the baseline. The statistical parameters of the baseline PDF are derived from known healthy data, typically collected from the beginning of a component's lifecycle when it is known that no fault exists or any fault is negligible. In this case, a normal distribution $N(0, \sigma)$ is used to define this baseline data. This indicator is essentially equivalent to an estimate of type II error, or equivalently the probability of detection. The statistical confidence can be seen in Figure 11. Customer specifications are translated into acceptable margins for the type I and II errors in the detection routine. If additional information is required, it is possible to compute the value of the Fisher Discriminant Ratio. The baseline PDF of the faulty resistance and the estimated one at times $t = 107sec$ and $t = 200sec$ can be seen

on Figures 12 and 13, respectively.

m	21 kg	\mathcal{I}	$0.0948 \text{ kg} \cdot \text{m}^2$
d	0.3 m	R_m	1.779Ω
K_Ω	0.1066 V/RPM	b	$0.000114 \text{ N} \cdot \text{m} \cdot \text{sec}$
K_t	0.1066 V/RPM	K_f	$5.1991 \cdot 10^{-4} \text{ N} \cdot \text{sec}$

Table 1. Simulator parameters

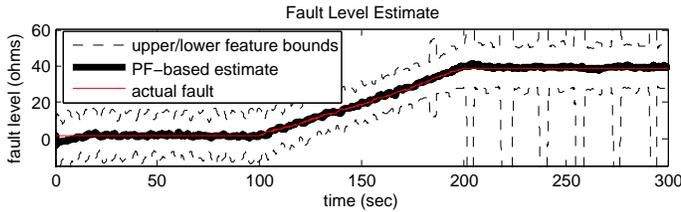


Figure 10. This figure illustrates the actual fault value and the estimated value from the PF algorithm.

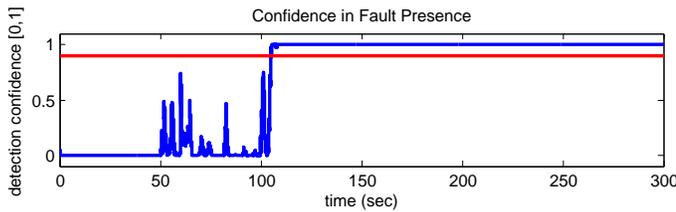


Figure 11. This figure illustrates the confidence metric.

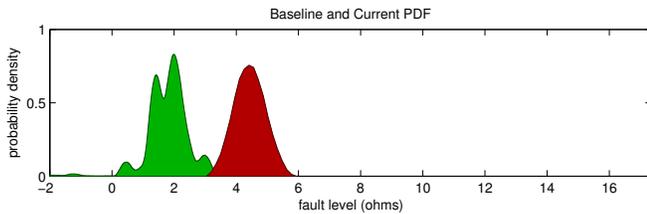


Figure 12. This figure illustrates the baseline (green) and estimated PDF (red) of the faulty resistance at $t = 107 \text{ sec}$.

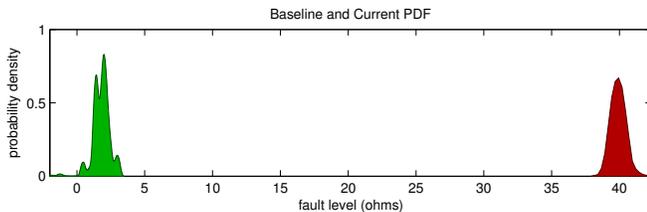


Figure 13. This figure illustrates the baseline (green) and estimated PDF (red) of the faulty resistance at $t = 200 \text{ sec}$.

7. CONCLUSIONS

This paper is introducing a method for the development, implementation, testing and assessment of a particle-filtering-

based framework for FDI. The proposed algorithm for FDI has been successful and very efficient in pinpointing abnormal conditions in very complex and nonlinear processes, such as the detection of a faulty motor to an autonomous hovercraft. A series of features has been derived using first principles modeling of the vehicles and thruster dynamics. The FDI algorithm can be executed on-line and is decoupled from the autonomous vehicle's control system. This allows for early fault detection minimizing the "masking" that can be potentially caused by the control system. In addition, the algorithm is capable of fusing information obtained from different subsystems of the vehicle.

ACKNOWLEDGMENT

This work was sponsored by NASA's Ames Research Center and Impact Technologies. We gratefully acknowledge their support and contributions to the research effort. We would also like to recognize the contributions of Edward Balaban and Brian Bole for their valuable input and feedback in reviewing this paper.

REFERENCES

- <http://aviationknowledge.wikidot.com/aviation:ergonomics-of-cae-7000-series-simulator-configured>. (n.d.).
- Orchard, M. (2007). *A particle filtering-based framework for on-line fault diagnosis and failure prognosis*. Unpublished doctoral dissertation, Georgia Institute of Technology.
- Orchard, M., & Vachtsevanos, G. (2007). A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine. In *Control automation, 2007. med '07. mediterranean conference on* (p. 1-6). doi: 10.1109/MED.2007.4433871
- Orchard, M., & Vachtsevanos, G. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, 31, 221-246.
- Sallee, G. P., & Gibbons, D. M. (1998). *Aia/aecma project report on propulsion system malfunction plus inappropriate crew response (psm+icr)* (Vol. I&II). Aerospace Industries Association.
- Statement, F. P. (2022, March). Faa policy on type certification assessment of thrust management systems. *FAA Policy Statement Number ANM-01-02*.
- Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M. E., & Vachtsevanos, G. (2011, May). A probabilistic fault detection approach: Application to bearing fault detection. In *Ieee transactions on industrial electronics* (Vol. 58).