Predictive System Reconfiguration for Fulfillment of Future Mission Requirements

Bruno P. Leao[1], Justinian Rosca[2], David P. Fecek[3], Karl Reichard[4] and Xianyong Feng[5]

[1,2]*Siemens Technology, Princeton, NJ, 08540, USA*
bruno.leao@siemens.com
justinian.rosca@siemens.com

[3,4]*Applied Research Laboratory, Penn State University, University Park, PA, 16802, USA*
dpf5013@arl.psu.edu
kmr5@arl.psu.edu

[5]*Center for Electromechanics, The University of Texas at Austin, Austin, TX, 78758, USA*
x.feng@cem.utexas.edu

ABSTRACT

Equipment failures can cause major disruptions to system operations. Although this is the case for engineered systems in general, it is especially applicable to autonomous systems as a maintenance crew may not be available to remediate the situation during operation. Autonomous vehicles, for instance, may be performing a critical mission miles away from the nearest manned support personnel when a failure occurs. In this work we propose and test an approach for automated predictive reconfiguration of an autonomous vehicle with the goal of delaying the occurrence of failures that would otherwise compromise mission accomplishment. The proposed approach is based on the Monte Carlo Tree Search (MCTS) method and assumes the availability of models describing relevant failure mechanisms and the relation between degradation and performance for each failure mode. Our solution introduces novel means for taking into account the uncertainty resulting from estimation of relevant parameters and states, with benefits in terms of reduction of computational cost compared to existing solutions. The proposed approach is successfully tested in a simulation environment.

1. INTRODUCTION

Failure diagnosis and prognosis solutions can potentially provide valuable information for operation and maintenance of a large variety of devices and systems. However, such potential is only converted into actual value when the corresponding information is used for taking operational and maintenance decisions. Therefore, the integration of decision-making and diagnosis/prognosis is of utmost importance for PHM researchers and practitioners. An increased interest in such integration can be noticed in recent years as evidenced, for instance, by recent review papers on this topic such as Bougacha, Varnier & Zerhouni (2020) who present a review of post-prognostics decision making and Hu, Miao, Si, Pan & Zio (2022) who present a review including decision making as one of the three main phases in PHM.

The focus of this work is on the predictive reconfiguration of an autonomous vehicle during mission execution to avoid that equipment failures jeopardize the accomplishment of his mission. One important requirement in this case is that degradation associated to relevant failure modes is controllable (Balaban, Johnson and Kochenderfer, 2019), i.e. it is possible to alter the rate at which equipment degrades based on changes in system configuration/operation. Although the integration of PHM and automated decision-making can be beneficial in various types of applications, autonomous systems correspond to a category of special relevance, as such type of system may operate for extended periods of time with no available manned support. The methodology is presented here in such context of an autonomous vehicle but it is potentially applicable to any situation where controllable failure modes may preclude the execution of a mission and it is beneficial to delay the failure so the mission can be accomplished.

The methodology developed and employed here is in line with the analysis presented by Balaban et al. (2019) which suggests that prognostics itself is not meaningful for controllable degradation processes and therefore health monitoring (and not prognostics) should be integrated with

Research reported in this publication was supported in part by the Office of Naval Research ONR-331 under award number N00014-21-C-1001

decision-making to achieve more efficient and effective solutions. The referred paper presents a detailed analysis on the topic. This contrasts with most of the predictive decision-making work developed in the PHM field where prognostics is performed first to produce RUL estimates which are in turn employed in decision making (Bougacha et al.,2020).

However, there are also important differences between the work presented here and that from Balaban et al. (2019). In the referred work, partially observable Markov decision process (POMDP) is presented as the mathematical framework to be employed for decision making under state estimation uncertainty. Although POMDP is a powerful framework which can be used for the type of problem described here, it brings considerable additional complexity compared to the standard Markov decision process (MDP) (Bellman, 1957), requiring computationally expensive methods such as POMCP (Silver & Veness, 2010) and DESPOT (Ye, Somani, Hsu & Lee, 2017) for achieving approximate solutions. Here we propose an alternative formulation which is based on the representation of the problem as a standard MDP and include novel adaptations to the standard Monte-Carlo Tree Search (MCTS) method (Browne et al., 2012) for efficiently solving the sequential planning problem taking into consideration state estimation uncertainty. Such uncertainty which plays a very relevant role in the context considered here as the predictive reconfiguration problem comprises estimating future degradation states of the system. Standard MCTS does not provide means for taking that into account.

The paper is organized as follows: the proposed methodology is described in section 2; a description of the autonomous vehicle and associated use case is presented in section 3; section 4 comprises experiments and results and section 5 is the conclusion.

2. METHODOLOGY

2.1. MCTS with Uncertainty Propagation (MCTSUP)

Since the proposed formulation is based on the MDP framework, corresponding definitions must be made. $s_t \in S$ is defined here as a vector containing the value of all relevant states at time instant t. One notable subset of states in vector s_t is s_t^d which corresponds to the degradations associated to all failure modes of interest. As the formulation takes into consideration the uncertainty in state estimation, $\hat{s}_t$ is also defined as the estimate of the state vector, corresponding to a multivariate random variable $\hat{s}_t \sim \Phi_{s_t}$. Actions $a_t \in A$ correspond to system configurations which belong to the finite set A . Examples of configuration changes could correspond, for instance, to turning on or off a certain redundant component, changing the topology of an electrical or hydraulic circuit or defining a new power level among a set of pre-defined options. Although actions are assumed to be discrete, the method could be easily adapted to deal with continuous actions by means of usual MCTS variations such as Double Progressive Widening (DPW) (Couëtoux, 2011). Actions can also be state dependent, i.e. $a_t \in A(s_t)$. State transition functions (Eq. (1)) describe how a new state (s_{t+1}) is achieved as a function of the current state and the action taken.

$$s_{t+1} = T(s_t, a_t, m_t; \theta) \tag{1}$$

For the degradation states (s_t^d) this transition corresponds to the failure mechanism models. Transitions are parameterized by vector θ which, in the case of failure mechanism models, define how fast the degradations should evolve over time. Transitions also depend on inputs associated to the operating conditions which in turn are function of mission requirements at each time instant (m_t). A transition function similar to Eq. 1 can be defined for updating state estimates (Eq. (2)):

$$\hat{s}_{t+1} = T\left(\hat{s}_t, a_t, m_t; \hat{\theta}\right) \tag{2}$$

It can be noticed from Eq. 2 that uncertainty in parameters estimates ($\hat{\theta} \sim \Phi_\theta$) is also taken into consideration in such transition. The reward r_t is a scalar value calculated based on state estimates as defined in Eq. 3. More specifically, point estimates of Φ_{s_t} or $\Phi_{f(s_t)}$ are employed in the calculation. In Eq. (3) and Eq. (4), $f(.)$ is an arbitrary function and $\gamma(.)$ corresponds to a point estimate.

$$r_t = R(\hat{s}_t, a_t) = R\left(\gamma\left(\Phi_{f(s_t)}\right), a_t\right) \tag{3}$$

$$f(\hat{s}_t) \sim \Phi_{f(s_t)} \tag{4}$$

MCTS method comprises four steps: selection, expansion, rollout, and backpropagation (Browne et al., 2012). In the method proposed here, selection and backpropagation are the same as in standard MCTS, therefore they are not discussed further. Both other steps, expansion and rollout, include performing state transitions based on defined actions (although rollout may include simplifications to this process). Therefore, if the actual state and parameters were known, Eq. 1 could be used for such calculations, leading to the standard MCTS. However, assuming only uncertain estimates of those values are available, adaptations are required. For the application described here, it is assumed that $\hat{s}_t$ and $\hat{\theta}$ are produced by a health monitoring algorithm at time instant t. σ can then be defined according to Eq. (5), where Φ_{s_t,θ_t} is the joint probability distribution associated to the pair $\hat{s}_t$, $\hat{\theta}$.

$$\sigma = \hat{s}_t, \hat{\theta} \sim \Phi_{s_t,\theta_t} \tag{5}$$

A set of samples $\sigma'^{(i)}$, $i = 1,2, \ldots, n$ is obtained from Φ_{s_t,θ_t}, each of them combining a sample of the states ($s'^{(i)}$) and a sample of the parameters ($\theta'^{(i)}$) as presented in Eq. (6).

$$\sigma'^{(i)} = \begin{bmatrix} s'^{(i)} \\ \theta'^{(i)} \end{bmatrix} \tag{6}$$

The definition presented here employs random unweighted samples, but other related approaches such as random weighted samples as used in Particle Filters (Gordon, Salmond, & Smith, 1993) or structured weighted points such as Sigma Points (Julier & Uhlmann, 2002) could also be employed. Samples ${\sigma'}^{(i)}$ are employed as a representation of the states and parameters at the root node of the MCTS tree. Eqs. (7) and (8) present how those variables are updated when transitioning from the root node r to a child node c through action a, based on mission requirements m. In the equations, subscripts indicate the corresponding node. Updates corresponding to transitions between other nodes follow the same process. Therefore, each node in the search tree contains a full representation of the states and parameter probability distributions in the form of samples.

$$ {s'}_c^{(i)} = T\left({s'}_r^{(i)}, a, m; {\theta'}_r^{(i)}\right) \tag{7} $$

$$ {\theta'}_c^{(i)} = {\theta'}_r^{(i)} \tag{8} $$

One relevant point to consider is that, as all samples are submitted to the same calculations during transition between nodes, vectorization, as available in most modern computational platforms, can be employed to achieve computational costs that scale sublinearly as a function of the number of samples (Van Der Walt, S., Colbert, S. C. & Varoquaux, G., 2011).

Calculation of the reward function at each node is performed using the corresponding state samples for obtaining the point estimate defined in Eq. (3). Eq. (9) presents the point estimate calculation for node o where $\gamma'(.)$ corresponds to the approximate calculation based on the set of samples:

$$ \gamma\left(\Phi_{f(s_o)}\right) \approx \gamma'\left(\left[f\left({s'}_o^{(1)}\right), f\left({s'}_o^{(2)}\right), \dots, f\left({s'}_o^{(n)}\right)\right]\right) \tag{9} $$

The method described above is referred hereon as MCTS with Uncertainty Propagation (MCTSUP). In the case where samples correspond to random unweighted samples as described above, MCTSUP can be interpreted as including an additional layer of Monte Carlo calculations to the MCTS method, corresponding to the propagation of the samples through nodes for calculation of rewards taking into consideration the state and parameter uncertainties.

The characteristics of MCTSUP create additional opportunities for tuning the method. For instance, the number of propagated samples could be reduced for rollout operations to speed up computation.

2.2. Reconfiguration

The diagram in Figure 1 presents the proposed MCTSUP-based vehicle reconfiguration approach. The MCTSUP-based agent defines current and future actions $a_{t:t_f}$, t_f corresponding to the end time of the mission and each action corresponding to a system configuration. A health monitoring system provides stochastic estimates of current degradation parameters and states ($\hat{\theta}_t$ and $\hat{s}_t^d$, respectively) at time t. Mission requirements for current and future steps in the mission are also know ($m_{t:t_f}$).

Failure mechanism models consist of the transition functions (Eq. (2)) which are used to extrapolate the uncertain degradation estimates (in the form of samples) for future points in time, given the estimates at time t ($\hat{s}_{t:t_f|t}^d$).

Function f(.) (Eq. (4)) consists of the estimator of system capabilities corresponding to a mission requirement as a function of current degradation states. For instance, supposing that the mission requirement for time instant t is to have power produced by the system (P_t) above a certain minimum level $P_{t,min}$, i.e. $P_t \geq P_{t,min}$, function f(.) provides an estimate of how much power can be produced by the system given the degradation states at time instant t.

Finally, the rewards ($r_{t:t_f}$) are calculated based on the point estimate function (Eq. (9)) which consist of the negative of the risk of not fulfilling the mission requirement, e.g. considering minimum power level requirement as mentioned above, $p(P_t < P_{t,min})$ would correspond to the risk of not fulfilling such requirement, where p (.) corresponds to probability, and $r_t = -p(P_t < P_{t,min})$.

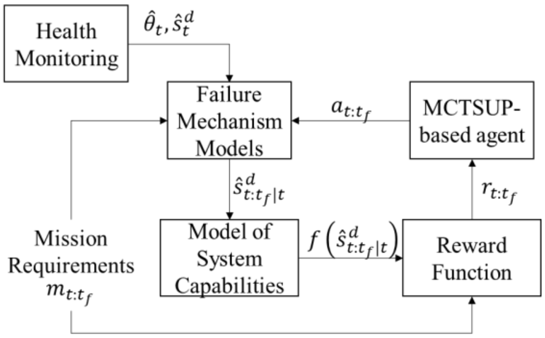
Health Monitoring
$\hat{\theta}_t, \hat{s}_t^d$
Failure Mechanism Models
$a_{t:t_f}$
MCTSUP-based agent
$\hat{s}_{t:t_f|t}^d$
$r_{t:t_f}$
Mission Requirements $m_{t:t_f}$
Model of System Capabilities
$f\left(\hat{s}_{t:t_f|t}^d\right)$
Reward Function

Figure 1 - Reconfiguration methodology

Description in this subsection corresponds to one version of the method, which is employed for analysis and experiments. However, alternative options for function f(.), point estimate function and reward function could be employed. For instance, the reward function could include terms that penalize changes in system configuration or the difference between a defined configuration and a standard one.

3. Autonomous Vehicle

The autonomous vehicle considered in this work is a surface ship where diesel engines and electric motors are employed respectively for generating electric power and driving the propellers. The diagram in Figure 2 provides a high-level view of the components of the vehicle's propulsion system which are relevant for this work. The solid arrows in the diagram indicate how mechanical power is transmitted from the diesel engines to the propulsion motors which in turn drive the propellers. Dashed lines indicate failure modes of interest affecting each piece of equipment. Given the architecture of the system as presented in Figure 2, propellers can still be driven in case of degradation or failure of an engine or a motor. However, degradations or failures will in general reduce the maximum power which can be employed by the system to drive the propellers. Therefore, even though the propellers may be driven using, for instance, a single engine, such configuration may not provide enough power for achieving required speeds at some points during the mission.

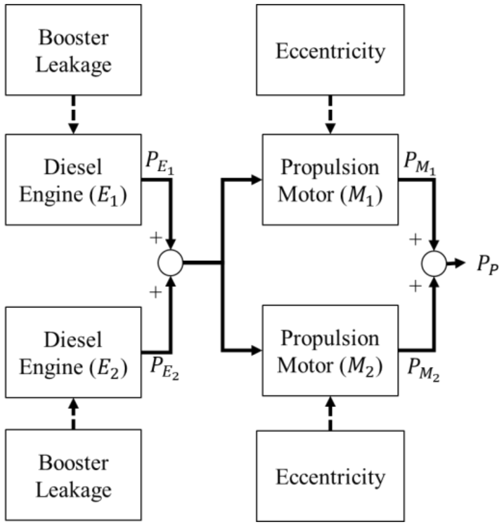

Figure 2. Autonomous vehicle's propulsion system. Solid arrows indicate the power transmission from the diesel engines (E_1, E_2), through the propulsion motors (M_1, M_2), reaching the propellers. P_P corresponds to the total power used to drive the propellers. Dashed arrows indicate the failure modes affecting each equipment.

The following subsections provide more details about propulsion system equipment as well as failure modes and models which are relevant for this work. The system, its architecture and components do not reflect those of a real autonomous surface vehicle, but a fictitious one. Corresponding simulation models have been developed based on data from literature.

3.1. Surface Vehicle Dynamics

For the application considered in this work, mission requirements are defined in terms of vehicle speed. The corresponding power demand associated to those speeds must be obtained so it can be used as requirement for the propulsion system. In this subsection, the vehicle dynamics model is used to estimate the power consumption based on a vessel speed. The vehicle dynamic equation is shown in Eq. (10).

$$M\frac{dU}{dt} = T(1 - t_d) - R_{ship} - F \qquad (10)$$

where, *M* is the total mass of the vehicle (kg), *U* is the vehicle speed (m/s), t_d is thrust deduction coefficient, R_{ship} is total vehicle resistance (N), *T* is the total propeller thrust (N), and *F* is wave disturbance (N). The vehicle resistance equation can be express as in Eq. (11):

$$R_{ship} = R_F + R_R + R_{wind} \qquad (11)$$

where, R_F, R_R, and R_{wind} are frictional resistance, wave-making resistance, and wind resistance, which are defined as presented in Eqs. (12):

$$\begin{aligned} R_F &= \frac{1}{2}C_F\rho_w U^2 S \\ R_R &= \frac{1}{2}C_R\rho_w U^2 S \\ R_{wind} &= \frac{1}{2}C_{air}\rho_{air} U^2 A_T \end{aligned} \qquad (12)$$

where, C_F, C_R, and C_{air} are drag coefficients for water-ship fraction, wave-making, and wind resistances, A_T is the advance facing area in the air, ρ_w and ρ_{air} are water density and air density, and *S* is the wetted area. The thrust and torque equations can be expressed according to Eqs. (13):

$$\begin{aligned} T &= sgn(n_p)\beta\rho_w n_p^2 D^4 K_T \\ T_T &= sgn(n_p)\beta\rho_w n_p^2 D^5 K_P \\ K_T &= 0.4 - 0.364 \cdot J_A \\ K_P &= 0.06 - 0.0545 \cdot J_A \\ J_A &= \frac{U(1-\omega)}{n_p D} \end{aligned} \qquad (13)$$

where, T_T is the propeller torque (N-m), n_p is propeller speed (rps), β is the loss factor (1: propeller stays in the water), *D* is the propeller diameter (m), and *w* is the wakefield. More model details could be found in Hou, Sun & Hofmann (2018) and Hou, Song, Hofmann & Sun (2021). The major model parameters are given in Table 1.

Table 1. Key Parameters for Vehicle Dynamic Model

Parameter name	Value
Ship mass *M*	145,000 kg
Thrust deduction coefficient t_d	0.2
Water resistance coefficient C_F+C_R	0.0043
Air resistance coefficient C_{air}	0.8
Advance facing area in the air A_T	9 m^2
Wetted area *S*	100 m^2
Sea water density ρ_w	1023 kg/m^3
Air density ρ_{air}	1.225 kg/m^3
Loss factor β	1
Propeller diameter *D*	1.4 m
Wakefield *w*	0.1

If we ignore the wave disturbance in the model, the steady state equation for vehicle dynamics can be expressed as presented in Eq. (14):

$$T(1-t_d) - R_{ship} = 0 \tag{14}$$

Assuming that the two propellers equally share the power demand and using the parameters from Table 1, a steady state equation can be obtained and solved to obtain a unique propeller speed n_p as a function of the ship speed U. The total power demand can then be obtained as a function of the ship speed as presented in Eq. (15):

$$P = 4\pi \cdot sgn(n_p)\beta\rho_w n_p^3 D^5 \cdot \left(0.06 - 0.0545 \cdot \frac{U(1-w)}{n_p D}\right) \tag{15}$$

3.2. Diesel Engine System

The engine employed in this work corresponds to a 4-stroke, compression-ignition diesel engine with variable geometry turbocharger and exhaust gas recirculation. Important engine specifications are given in Table 2.

Table 2. Specifications for the Diesel Engine Model

Specification name	Value
Number of cylinders	4
Total displacement	4.4 L
Piston strokes per power cycle	4
Nominal operating speed	1800 rpm
Maximum torque	620 Nm
Rated power output	108 kW

A corresponding diesel engine simulation model was developed based on MATLAB/Simulink Powertrain Blockset[1]. It consists of a Mean-Value Engine Model, which calculates engine processes and effects in terms of lumped parameters averaged across the entire engine cycle. Individual cylinder dynamics are not considered.

The model includes a number of individual subsystems. The core engine subsystem calculates cylinder air mass-flow, engine output torque, and exhaust output temperature using a combination of physical equations and predefined lookup tables. The intake system model consists of several components modelling the dynamics of the air inlet, an air filter flow restriction, intake tubing, the turbocharger compressor, an intercooler, and a control volume representing the engine intake manifold. Similarly, the exhaust system model consists of an exhaust manifold control volume, the turbocharger turbine, exhaust tubing, exhaust flow restrictions, and finally an exhaust outlet. Torque calculated from excess exhaust gas enthalpy provided to the turbocharger turbine is transferred to the turbocharger compressor by a simple rotating shaft model. This allows the compressor model to calculate an increased, or boosted, intake pressure. The intake and exhaust system models calculate the resulting pressures, temperatures, mass flows, and energy flows entering and exiting the core engine subsystem.

In addition to the engine dynamics models, a controller model is included to manage simulated fuel delivery rates in response to engine load. This controller model uses predefined fuel map lookup tables to determine the amount of fuel provided for a given engine speed and load. To load the simulated engine, a simple generator model and speed governor is used.

Several failure mechanisms have been integrated into the engine model simulation. For the purposes of this paper, only the boost leak failure mode will be discussed. A boost leak failure can be described as the loss of increased intake pressure generated by the turbocharger as the result of a leak in the intake system venting to atmosphere. This leak may be created by an improper seal, damaged manifold gasket, cracked intake tubing, etc. Such a leak has the effect of reducing available engine output power, reducing fuel efficiency, and increasing emissions related to unburnt fuel in the exhaust. Unburnt fuel may also cause further damage to exhaust components, in particular the turbocharger or any catalysts or particulate filters that may be installed [Heywood, 1988].

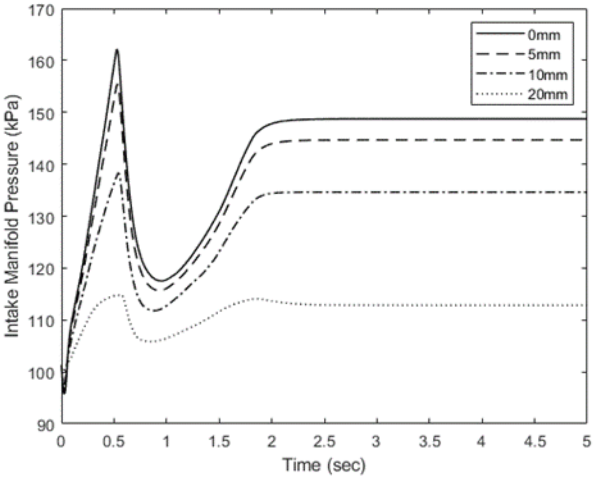

Figure 3. Effects of boost leak size on intake manifold pressure.

To simulate a boost leak, the intake system model is modified to include an orifice with variable cross-sectional area at the intake manifold, leading directly to atmosphere. The cross-sectional flow area is defined in terms of an equivalent

[1] ttps://www.mathworks.com/help/autoblks/propulsion.html

circular diameter. Figure 3 shows the effects of leak size on the intake manifold (boost) pressure during a short ramp to operating speed at 50kW load. It can be noticed from the plot that boost leak can significantly reduce engine performance.

Because diesel engines control torque by varying the fuel rate, reduced intake air flow as a direct result of reduced intake manifold pressure requires additional fuel to maintain a constant engine speed. The ability of the engine fuel rate controller to compensate for reduced intake pressure is limited, eventually causing the engine to stall, or in the case of a generator system, the engine speed to drop below the threshold necessary to maintain the correct electrical output frequency [Heywood, 1988].

The failure mechanism model presented in Eq. (16) is employed to describe how booster leakage evolves over time as a function of engine operation.

$$\min\left(\int_{t_0}^{T}\left(C_P \max\left(\frac{p_{int}^2(t) - p_{amb}^2}{p_{amb}}, 0\right) + C_N\big(f(N(t))\big)\right)dt, A_{L,max}\right) = A_L \tag{16}$$

Leak area, A_L (m^2), growth is assumed to be a function of the intake manifold pressure, p_{int} (Pa), ambient pressure p_{amb} (Pa), and engine vibration [Naval Surface Warfare Center Carderock Division, 1992]. Operating load affects the equation through p_{int} . C_P (m^2/(Pa·s)) and C_N (m^2/s) are tuning parameters that can be used to control the leak growth rate regarding boost pressure and engine speed, respectively.

Table 3. Engine speed scaling factors for the leak growth model

Engine Speed (rpm)	Scaling Factor
0	0
800	0.1
1000	0.3
1800	0.1
2000	0.5
2500	1

Table 4. Baseline parameter values for the leak area growth model

Parameter name	Value
$A_{L,max}$	0.01 m^2
C_P	$1.4 \cdot 10^{-12}$ m^2/(Pa·s)
C_N	$1.2 \cdot 10^{-8}$ m^2/s
P_{amb}	$1.01325 \cdot 10^{-5}$ Pa

The maximum leakage area is limited to the value of $A_{L,max}$ (m^2) Engine vibration effects are implemented as a 1-D lookup table of scaling factors correlated to engine speed N (rpm), shown in Table 3. Vibration effects are assumed to be minimized at idle and nominal operating speed due to physical engine balancing and design. Baseline parameter values for the leak growth model are provided in Table 4.

3.3. Propulsion Motor

The propulsion motors are supplied by variable frequency drives to drive propellers and maintain ship speed at the desired level. In this study, squirrel-cage induction machines are considered. During field service, propulsion motors may experience various failure modes such as winding insulation failure, transient overload, eccentricity, and cracked stator housing. Eccentricity is chosen as the failure mode of interest affecting propulsion motors for this study.

Incorrect bearing positioning during assembly, worn bearings, or bent rotor shaft may cause rotor eccentricity in induction machines. Two common forms of rotor eccentricity include static eccentricity and dynamic eccentricity. The static eccentricity is caused by a static rotor displacement from stator bore center, but the rotor still rotates upon its own center. The static eccentricity rotor tends to pull in one direction, which makes the unbalanced magnetic pull difficult to detect. The dynamic eccentricity rotor rotates upon the stator center, but the rotor does not rotate upon its own center. The dynamic eccentricity rotor produces unbalanced magnetic pull and the force will reflect on the rotor's angular velocity in normal operation. It is relatively straightforward to detect the dynamic eccentricity through vibration monitoring. In this work, the focus is on the static eccentricity.

Equation (17) presents the failure mechanism model assumed for eccentricity, corresponding to a discrete time exponential growth. In the equation, e_t corresponds to motor eccentricity at time t, P_M is the mechanical power produced by the motor, ϕ_t corresponds to the ratio between the power produced by the motor and the total power produced by both motors and α adjusts the rate at which the degradation evolves. Baseline value considered for α is 0.1 and maximum eccentricity value, which corresponds to motor failure, is 10. When eccentricity exceeds this maximum value, the motor has reached failure.

$$e_{t+1} = max([max(\alpha \cdot P_M, 1)\,(0.6 + \phi_t)e_t], e_t) \tag{17}$$

Degradation associated to eccentricity is not considered to affect the capability of the motor to drive the propeller. Eccentricity will only limit vessel propulsion power and consequently speed when it turns into motor failure so the corresponding motor cannot operate any longer.

4. Experiments and Results

Simulated experiments have been performed to assess the performance of the proposed method. The methodology depicted in Figure 1 was implemented and tested. Failure mechanism models and models of system capabilities were based on the information described in section 3. The corresponding MDP model and MCTS were implemented based on the POMDP library for Julia language[2]. More details about experiments and results are described in the following subsections. The reward function is based on the risk of not fulfilling the mission due to equipment degradation or failure as presented in section 2.1.

4.1. Mission and Requirements

The mission to be accomplished by the autonomous vehicle has a duration of 16 days, also referred to as steps hereon. At each day, a certain speed is required to accomplish the mission. Such speed can be converted to required power by using Eq. (15). Figure 4 presents the power requirements for each step in the mission ($P_{m,t}$) employed for the experiments. It can be noticed that different activities are performed in each step of the mission, leading to different power requirements for each step. $P_{m,t}$ in this case corresponds to $P_{t,min}$ as defined in section 2.2. Using the definition of P_t from the same section, corresponding to the power the propulsion system can produce during step t, mission requirements m_t correspond to a single requirement: $P_t \geq P_{m,t}$ and the reward function corresponds to $r_t = -p(P_t < P_{m,t})$.

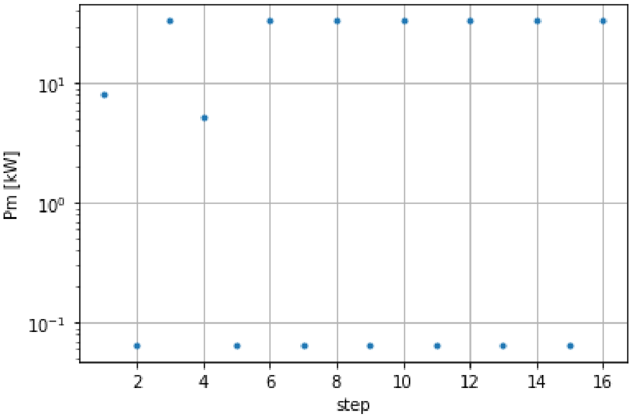

Figure 4. Required power (kW) for each step (day) in the mission.

It is assumed that the predictive reconfiguration yielded by the method proposed here can be applied at the beginning of each day, therefore, each day corresponds to a (potentially different) configuration.

4.2. Experiments

Possible configurations for the experiments consist of defining which diesel engines and propulsion motors are operating at a certain step in the mission. Standard configuration consists of both engines and both motors operating the whole time. Possible engine configurations will be referred to E1, E2 or E12 indicating which engines are operating (respectively engine 1, engine 2 or both above). Possible motor configurations are M1, M2 and M12 analogously. When both engines or both motors are operating they equally share the loads. In this case, standard configuration corresponds to (E12, M12). It must be noticed that configuration can be defined independently for diesel engines and propulsion motors. Therefore, the set of actions A, as defined in section 2.1, corresponds to all possible combinations of engine and motor configurations.

Defining $l_{i,t}$ as the degradation (corresponding to leak area) associated to engine $i, i = 1{,}2$ at time t, and $e_{j,t}$ as the eccentricity affecting motor $j, j = 1{,}2$ at time t, the state vector $\mathrm{s_t}$, as defined in section 2.1, here corresponds to the definition in Eq. (18). Eq. (19) describes the initial conditions employed for the experiments.

$$s_t = \left[l_{1,t}, l_{2,t}, e_{1,t}, e_{2,t}\right]^T \tag{18}$$

$$\left[l_{1,0}, l_{2,0}, e_{1,0}, e_{2,0}\right] = [0{,}0{,}2 \cdot 10^{-3}, 1 \cdot 10^{-5}] \tag{19}$$

It is assumed that a health monitoring system provides stochastic estimates of such degradation values and of corresponding parameters required for extrapolating degradation to future times. Those parameters correspond to C_P and C_N for the booster leak at each engine (Eq. (16)) and α for eccentricity at each motor (Eq. (17)). Therefore θ, as defined in section 2.1, here corresponds to the definition in Eq. (20).

$$\theta = [C_P, C_N, \alpha]^T \tag{20}$$

Estimated states and parameters are updated after each step in the mission so that configuration can be defined for the following step. Initial uncertainty associated to each parameter is defined in the form of Gaussian distributions as presented in Eq. (21).

$$\begin{aligned} \hat{C}_P &\sim \mathrm{N}(1.4 \cdot 10^{-12}, 2 \cdot 10^{-13}) \\ \hat{C}_N &\sim \mathrm{N}(1.2 \cdot 10^{-8}, 2 \cdot 10^{-9}) \\ \hat{\alpha} &\sim \mathrm{N}(0.1, 3 \cdot 10^{-3}) \end{aligned} \tag{21}$$

100 samples from the joint state/parameter distribution are employed in the experiments (Eq. (6))

4.3. Results

Before analyzing the results of application of MCTSUP, the baseline solution is considered for comparison consisting of the system operation under the standard configuration (E12, M12) throughout the whole mission. Initial conditions and parameter uncertainties are according to Eqs. (19) and (21) respectively and power requirements as presented in Figure 4.

[2] https://github.com/JuliaPOMDP/POMDPs.jl

Table 5. Experiment results for standard configuration

Step	Configuration	Risk
1	(E12, M12)	0
2	(E12, M12)	0
3	(E12, M12)	0
4	(E12, M12)	0
5	(E12, M12)	0
6	(E12, M12)	0
7	(E12, M12)	0
8	(E12, M12)	0
9	(E12, M12)	0
10	(E12, M12)	0
11	(E12, M12)	0
12	(E12, M12)	0.09
13	(E12, M12)	0
14	(E12, M12)	0.33
15	(E12, M12)	0
16	(E12, M12)	0.91

Figure 5 presents the quartiles associated with the evolution of the four failure modes of interest.

Table 5 presents the risk of not being able to fulfill mission requirements at each step. It can be noticed that such risks become considerable during the final steps of the mission, achieving 91% at the final step.

For the MTCTUP-based solution 10000 iterations were used. Table 6 presents results of application of the method for each time step including the selected configuration, risk of not fulfilling the mission and computation time. It can be noticed that computation time is in most cases reduced from one step to the next. This is expected as in each step the current and future steps until end of the mission are considered in the calculations. A standard laptop computer with i7-6820HQ processor, 32GB of RAM and MS Windows 10 Enterprise operating system was employed for computation. Comparing these results to those yielded by the standard configuration (Table 5) it can be noticed that system is predictively reconfigured since step 1, with engines and motors being selectively turned off whenever possible to delay degradation. Risk of not fulfilling mission requirements is successfully reduced, achieving a maximum value of 12% at the last step of the mission (compared to 91% for the standard configuration case).

Figure 6 presents quartiles associated with the evolution of the failure modes of interest. Comparing those plots with the ones presented in Figure 5 it can be noticed that degradation has been delayed in a significant way.

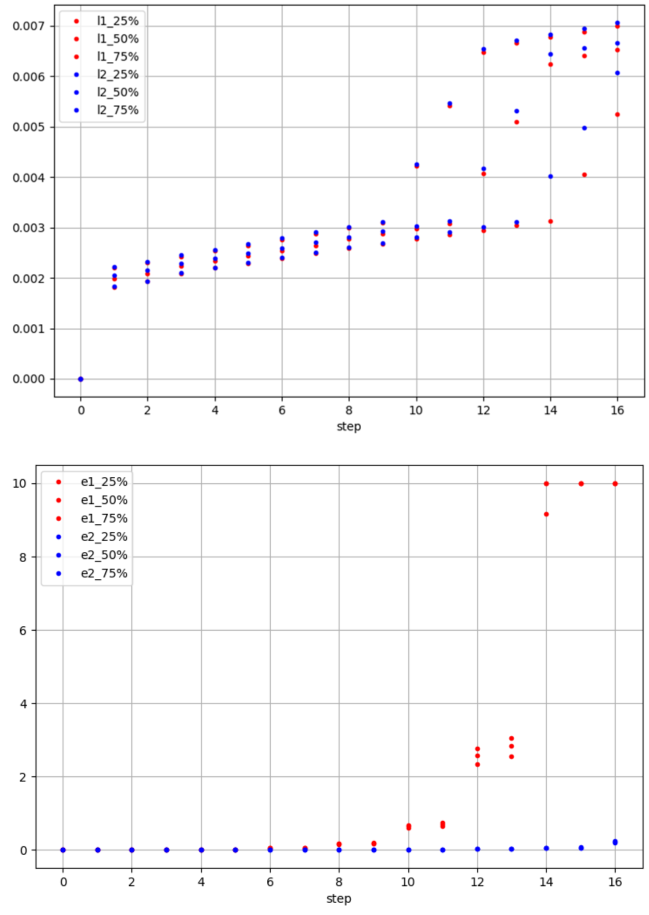

Figure 5. Quartiles associated to degradation evolution for each failure mode of interest (standard configuration). Top and bottom plots present respectively evolution of booster leak for engines 1 and 2 and eccentricity for motors 1 and 2.

Table 6. Experiment results for MCTSUP application

Step	Computation Time [s]	Configuration	Risk
1	1303.39805	(E12, M2)	0
2	1473.245654	(E1, M12)	0
3	1226.815517	(E12, M12)	0
4	1003.561821	(E1, M2)	0
5	924.4212584	(E2, M2)	0
6	988.6514444	(E12, M12)	0
7	796.3154714	(E2, M2)	0
8	793.7924268	(E12, M12)	0
9	622.0841589	(E2, M2)	0
10	539.383497	(E12, M12)	0
11	480.2801492	(E12, M12)	0
12	367.8734899	(E12, M12)	0
13	309.5346775	(E1, M2)	0
14	211.391253	(E12, M12)	0.03
15	147.4896767	(E2, M2)	0
16	65.25488496	(E12, M12)	0.12

A final test was performed increasing the number of samples to evaluate the corresponding impact in processing time. As the proposed methodology enables the use of vectorization, it is expected that the processing time should grow slower than linearly as a function of the number of samples. For this test the number of samples was increased 10x compared to the previous test, resulting in 1000 samples. Processing time at each step in this case increased on average 7.86x compared to the original calculations, confirming the expected performance.

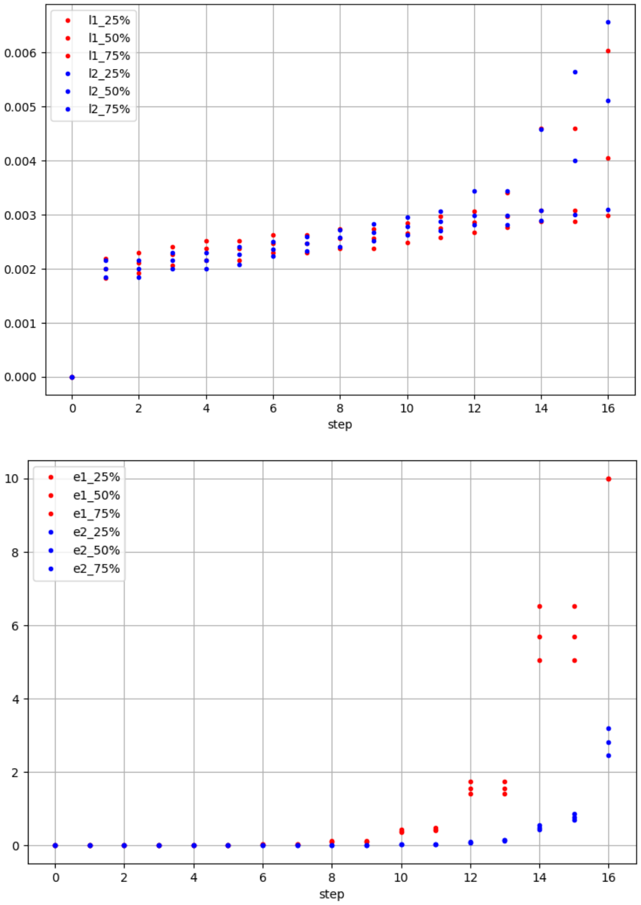

Figure 6. Quartiles associated to degradation evolution for each failure mode of interest (MCTSUP-based solution). Top and bottom plots present respectively evolution of booster leak for engines 1 and 2 and eccentricity for motors 1 and 2.

5. Conclusion

This paper presented a sequential decision-making methodology for predictively reconfiguring a system in order to delay the evolution of equipment degradation. The goal is to considerably reduce the risk of mission failure due to equipment failure when compared to operation using the standard system configuration. The methodology includes a novel adaptation of MCTS method which adds efficient means for propagating the uncertainty associated to estimation of states and parameters within a MDP framework.

The proposed methodology is tested based on a use case of an autonomous surface ship, taking into consideration failure modes that would affect its propulsion system. Results successfully demonstrate considerable reduction in risk associated with not fulfilling mission requirements when compared to the standard system configuration.

Future work includes benchmarking the proposed methodology with alternative state-of-the-art sequential decision-making approaches and its application to additional operational profiles and more complex and realistic use cases. Improvements to the methodology itself are also envisioned, such as taking into consideration constraints associated to reconfiguration options, additional sources of uncertainty and alternative means for propagating uncertainty. Extensions to the problem may also be considered, such as proposing alternative mission profiles in cases where reconfiguration of the system is not sufficient to assure success of the mission.

References

Balaban, E., Johnson, S.B. & Kochenderfer, M.J., (2019). Unifying system health management and automated decision making. *Journal of Artificial Intelligence Research*, vol. 65, pp.487-518. doi: 10.1613/jair.1.11366

Bougacha, O., Varnier, C. & Zerhouni, N., (2020). A review of post-prognostics decision-making in prognostics and health management. *International Journal of Prognostics and Health Management*, 11(15), pp. 31.

Hu, Y., Miao, X., Si, Y., Pan, E. & Zio, E., (2022). Prognostics and health management: A review from the perspectives of design, development and decision. *Reliability Engineering & System Safety*, 217. doi:10.1016/j.ress.2021.108063

Silver, D. & Veness, J., (2010). Monte-Carlo planning in large POMDPs. *Advances in neural information processing systems* (NeurIPS), 23.

Ye, N., Somani, A., Hsu, D. & Lee, W.S., (2017). DESPOT: Online POMDP planning with regularization. *Journal of Artificial Intelligence Research*, 58, pp.231-266.

Bellman, R., (1957). A Markovian decision process. *Journal of mathematics and mechanics*, pp.679-684.

Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. & Colton, S., (2012). A survey of Monte Carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), pp.1-43.

Couëtoux, A., Hoock, J.B., Sokolovska, N., Teytaud, O. & Bonnard, N., (2011). Continuous upper confidence trees. *International Conference on Learning and Intelligent Optimization* (pp. 433-445). Springer, Berlin, Heidelberg.

Gordon, N. J., Salmond, D. J. & Smith, A. F., (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F-radar and signal processing* 140(2), pp. 107-113.

Julier, S. J. & Uhlmann, J. K., (2002). Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. *American Control Conference*, pp. 887-892.

Van Der Walt, S., Colbert, S. C. & Varoquaux, G., (2011). The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2), pp.22-30.

Hou, J., Sun., J., & Hofmann, H. F., (2018). “Mitigating power fluctuations in electric ship propulsion with hybrid energy storage system: design and analysis,” IEEE Journal of Oceanic Engineering, vol. 43, no. 1.

Hou, J., Song, Z., Hofmann, H. F., & Sun (2021), J., “Control strategy for battery/flywheel hybrid energy storage in electric shipboard microgrids,” IEEE Trans. Industrial Informatics, vol. 17, no. 2.

Heywood, J. B. (1988). *Internal Combustion Engine Fundamentals*. McGraw-Hill Book Co.

Naval Surface Warfare Center Carderock Division. (1992). (rep.). *Handbook of Reliability Prediction Procedures for Mechanical Equipment*. Bethesda, Maryland