

## A Multi-Sensory Switching-stable Architecture for Distributed Fault Tolerant Propulsion Control of Marine Vessels

Kougiatsos, N.; Negenborn, R.R.; Reppa, V.

**DOI**

[10.24868/10728](https://doi.org/10.24868/10728)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

Proceedings of the International Ship Control Systems Symposium

**Citation (APA)**

Kougiatsos, N., Negenborn, R. R., & Reppa, V. (2022). A Multi-Sensory Switching-stable Architecture for Distributed Fault Tolerant Propulsion Control of Marine Vessels. *Proceedings of the International Ship Control Systems Symposium, 16*, Article 27. <https://doi.org/10.24868/10728>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# A Multi-Sensory Switching-stable Architecture for Distributed Fault Tolerant Propulsion Control of Marine Vessels

N Kougiatsos\*, R R Negenborn, V Reppa

*Department of Maritime and Transport Technology, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, 2628CD, The Netherlands*

\*Corresponding author. Email: n.kougiatsos@tudelft.nl

## Synopsis

Nowadays, marine vessels constitute safety-critical assets facilitating the transport of millions of passengers and tons of cargo worldwide. As such, they require a large number of heterogeneous sensors dispersed in the various on-board machinery for operational and condition monitoring of their vital systems, such as the propulsion system. Despite the vast availability of data from on-board sensors, there is hardly any collaboration between the spatially distributed sensor devices to boost vessel performance. Up to this day, physical redundancy has been mostly discussed in maritime literature and has also been required by certain ship system design regulations. The use of virtual sensors (software-based) has not been properly investigated yet for maritime applications, despite their successful application in other fields like aircraft control and process control. This paper proposes a novel switching mechanism to alternate between physical and virtual sensors used in the primary propulsion control layer of marine vessels aiming to compensate for the effects of sensor faults. The switching mechanism focuses on ensuring the safe performance of the propulsion grid after the sensor faults occur. The software sensors are constructed using mathematical models describing the nonlinear dynamics of the propulsion system and the input and sensor output data. Simulation results are used to illustrate the switching mechanism's performance in the case of a hybrid propulsion system, where the different subsystems are controlled in a distributed configuration.

*Keywords:* Hybrid propulsion; Sensor faults; Software-based sensors; Fault-tolerant control systems; Marine systems

## 1 Introduction

Maritime safety is a prerequisite for current vessels operating at sea and a basic pillar for the development of future ships. Despite this, recent statistics from the European Maritime Safety Agency (2020) indicate that 22 % of all vessel casualties for the year 2019 can be attributed to loss of propulsion, and this percentage is constantly increasing. The propulsion system can thus be considered safety-critical for the vessel operation.

A vessel can be considered as a large network of subsystems, interconnected physically (e.g. pipes) or in a cyber manner (e.g. communication networks). Due to their inherent complexity in design, marine vessels require numerous heterogeneous hardware sensors installed to monitor and control their various subsystems. It is estimated that only for one of the ship's main fuel engines, 15-17 sensors are installed onboard [Raptodimos et al. (2016)]. These sensor devices may be subject to faults during the vessel's lifecycle and are essential to the operators in order to make decisions.

In case of sensor faults, utilizing hardware redundancy has been proposed in literature, meaning that multiple copies of a sensor are installed in the physical plant [Wu et al. (2006)]. This approach is also followed in many cases in practice due to the existing design regulations of some safety-critical vessel systems such as the Dynamic Positioning system [DNV (2015)]. However, this approach entails higher cost for the shipowner, taking into consideration both installation and the maintenance costs of the extra hardware devices. Moreover, the physically redundant sensors can potentially fail as a unit due to the same cause (e.g. manufacturing defects), thus helping to conceal the fault. Alternatively, the vast availability of heterogeneous sensors onboard the vessel and the model information can be combined to construct software-based sensors instead [Blanke et al. (2016); Darvishi et al. (2021)]. Hardware and software-based sensors can then be combined to create a multi-sensory monitoring and control architecture. This way, the operational awareness inside the ship can be improved significantly.

In maritime literature there has been some research on sensor networks for propulsion systems. In [Li et al. (2012)], a fusion technique of vibration and wear particle analyses is proposed for the diagnosis of the tribo-system marine fuel engines. To this end, information from multiple hardware vibration sensors oriented in different

---

### Authors' Biographies

**Nikos Kougiatsos, MSc** is currently a PhD researcher at the Section Transport Engineering & Logistics of Department Maritime & Transport Technology, Delft University of Technology. His research involves scalable, modular and fault-tolerant control strategies for marine vessels. During his Master's in Naval Architecture and Marine Engineering from the National Technical University of Athens, he has been awarded two scholarships from the American Bureau of Shipping (ABS).

**Prof. dr. Rudy Negenborn** is professor in automatic control & coordination of transport technology at the Section Transport Engineering & Logistics of Department Maritime & Transport Technology, Delft University of Technology. Prof. Negenborn's research interests include multi-agent systems, distributed control, model predictive control for applications in (waterborne) networked transport systems.

**Dr. Vasso Reppa** is a tenure-track Assistant Professor with the Transport Engineering and Logistics Section of the Maritime and Transport Technology Department, Delft University of Technology. Dr. Reppa's research interests include fault tolerant control, fault diagnosis, interconnected uncertain systems, multi-agent control, autonomous shipping operations and waterborne transport.

directions and attached to the engine is used. Li et al. (2019) on the other hand construct a neural approximator for the temperature forecast of marine propulsion systems using the values of different hardware heterogeneous sensors at the input layer. The forecast is then compared to the actual sensor measurement to identify sensor faults. Then, in [Kerrigan and Maciejowski (2015)], the authors deal with both process and sensor faults in marine propulsion systems. For the latter, a Kalman estimator technique is used to reconstruct the measurements of the faulty sensors. However, all of these papers either focus on small parts of the propulsion system or consider a simplified version of the system model to design the virtual sensors. In addition, the stability of the system based on operational and control performance criteria is disregarded when switching from hardware to software-based sensors. Considering data-driven approaches, in [Darvishi et al. (2021)] a machine-learning-based framework is proposed for sensor validation considering a multilayer perceptron neural network architecture for vessels. In [Campa et al. (2008)], a sensor validation scheme for heavy-duty diesel engines is proposed using a hybrid scheme composed of Adaptive Linear Neural Networks for linear engine operating conditions as well as Minimal Resource Allocating Networks for non-linear engine conditions to create virtual sensors. However, all the aforementioned model-free approaches require a high number of neurons to calculate the output due to the high system nonlinearity and the detectability of sensor faults depends on the accuracy of the used training sets. Moreover, the generalization ability of the produced results is low.

Previous work of the authors [Kougiatsos et al. (2022)] focused on providing a distributed sensor fault diagnosis framework for marine fuel engines and other similar systems described by nonlinear Differential-Algebraic Equations. In [Kougiatsos and Reppa (2022)], three types of software-based sensors were proposed using the considerable number of heterogeneous hardware sensors onboard the vessel and model information; dynamic, static and Set Inversion via Interval Analysis (SIVIA) software-based sensors. Each of them can be assessed based on the time needed to reconstruct the faulty measurements and the final achieved accuracy. Insights on those characteristics have already been provided by the simulation results in [Kougiatsos and Reppa (2022)].

This paper proposes a multi-sensory switching mechanism for the propulsion control of marine vessels. The mechanism itself is composed of both hardware and software-based sensors alongside a switching logic between them. The sensor fault accommodation methodology and the design of software-based sensors for the general case of Differential-Algebraic systems (such as the marine fuel engine) has already been presented in [Kougiatsos and Reppa (2022)]. The focus of the present paper is the formulation of the logic to switch between hardware and software-based sensors while maintaining system stability in marine hybrid propulsion control architectures. To this end, both control performance and operational criteria are considered.

From the application point of view, the main contribution of this paper is the development of a multi-sensory switching-stable methodology that has not been proposed in maritime literature. In addition, compared to the commonly referred sensor networks where only hardware redundancy is used, the use of analytical redundancy is also explored. This work also considers a more complete hybrid propulsion model compared to the ones found in relevant literature, incorporating a Mean Value First Principle (MVFP) model for the fuel engine [Geertsma et al. (2017b)] coupled with the shaft and gearbox dynamics [Kalikatzarakis et al. (2018); Planakis et al. (2021)] and a static model for the electric motor [Wildi (2002)]. From the viewpoint of multi-sensory systems, the present work considers highly nonlinear Differential-Algebraic systems to design the virtual sensors.

The paper is organised as follows. In Section 2, the problem statement, the general mathematical formulation and the design of software-based sensors are provided. Section 3 then discusses the system models used for the paper. The sensor switching logic between multiple software and hardware-based sensors is discussed in Section 4 based on both control performance and operational criteria. Simulation results for a hybrid propulsion architecture are shown in Section 5, followed by concluding remarks and directions for future research in Section 6.

## 2 Problem Formulation

### 2.1 System decomposition & modelling

In this paper we consider a hybrid propulsion system, such as the one shown in Figure 1. The system has the standard configuration [Geertsma et al. (2017a)] and is composed of a marine fuel engine and an induction motor coupled to the same propeller via a gearbox. In the same figure, the sensors that are used for control are also highlighted. As can be seen in Figure 2, the system is partitioned in two parts: one containing the fuel engine, the gearbox, the shaftline and the propeller dynamics ( $\Sigma^{(1)}$ ) and another including only the induction motor ( $\Sigma^{(2)}$ ).

Each system can be internally composed of one or more subsystems. Let  $\Sigma^{(I,j)}$ ,  $I = \{1, 2\}$ ,  $j = \{1, \dots, j_I\}$  denote the  $j$ -th subsystem of system  $\Sigma^{(I)}$ . Each system  $\Sigma^{(I)}$  is described by the following nonlinear Differential-Algebraic dynamics:

$$\Sigma^{(I)} : \begin{cases} \dot{x}^{(I)}(t) = A^{(I)}x^{(I)}(t) + \gamma^{(I)}(x^{(I)}(t), z^{(I)}(t), u^{(I)}(t)) + h^{(I)}(x^{(I)}(t), z^{(I)}(t), \chi^{(I)}(t), u^{(I)}(t)), & (1a) \\ 0 = \xi^{(I)}(x^{(I)}(t), z^{(I)}(t), \chi^{(I)}(t), u^{(I)}(t)), & (1b) \end{cases}$$

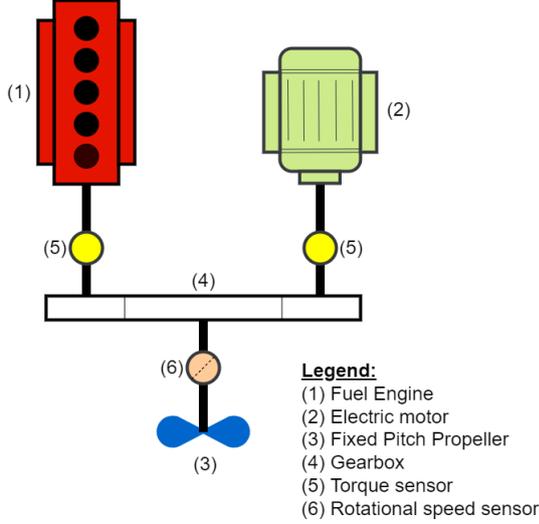


Figure 1: Marine Hybrid propulsion system

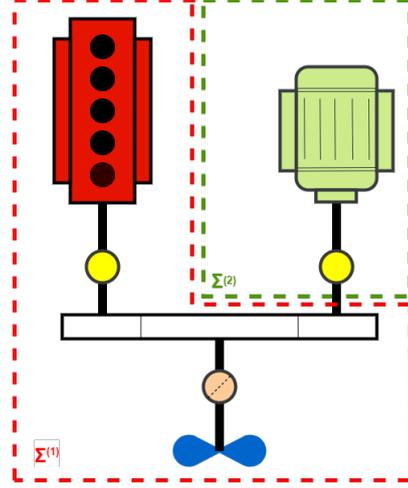


Figure 2: System partitioning

where  $x^{(l)} \in \mathbb{R}^{n_l-r_l}$  is the state variable vector,  $z^{(l)} \in \mathbb{R}^{r_l}$  is the algebraic variable vector,  $\chi^{(l)} \in \mathbb{R}^{k_l}$  are the interconnection variables from the neighbouring subsystems,  $u^{(l)} \in \mathbb{R}^{l_l}$  is the control input vector,  $\gamma^{(l)} : \mathbb{R}^{n_l-r_l} \times \mathbb{R}^{l_l} \mapsto \mathbb{R}^{n_l-r_l}$  represents the known nonlinear system dynamics,  $h^{(l)} : \mathbb{R}^{n_l-r_l} \times \mathbb{R}^{r_l} \times \mathbb{R}^{k_l} \times \mathbb{R}^{l_l} \mapsto \mathbb{R}^{n_l-r_l}$  represents the known interconnection dynamics with the neighbouring subsystems,  $\xi^{(l)} : \mathbb{R}^{n_l} \times \mathbb{R}^{k_l} \times \mathbb{R}^{l_l} \mapsto \mathbb{R}^{n_l-r_l}$  is a smooth vector field. The term  $A^{(l)}x^{(l)}$  represents the linear part of the system's  $\Sigma^{(l)}$  dynamics, where  $A^{(l)} \in \mathbb{R}^{(n_l-r_l) \times (n_l-r_l)}$  is assumed known.

Each subsystem incorporates a set of hardware sensors  $S^{(l)} = \bigcup_{k=1}^{n_l} S^{(l)}\{k\}$  described as:

$$S^{(l)} : \begin{cases} y_x^{(l)}(t) = x^{(l)}(t) + d_x^{(l)}(t) + f_x^{(l)}(t) \\ y_z^{(l)}(t) = z^{(l)}(t) + d_z^{(l)}(t) + f_z^{(l)}(t), \end{cases} \quad (2)$$

where  $y_x^{(l)} \in \mathbb{R}^{n_l-r_l}$  denotes the hardware sensor values corresponding to state variables,  $y_z^{(l)} \in \mathbb{R}^{r_l}$  denotes the sensor values corresponding to algebraic variables,  $d_x^{(l)} \in \mathbb{R}^{n_l-r_l}$ ,  $d_z^{(l)} \in \mathbb{R}^{r_l}$  are the measurement noise vectors and  $f_x^{(l)} \in \mathbb{R}^{n_l-r_l}$ ,  $f_z^{(l)} \in \mathbb{R}^{r_l}$  are sensor fault vectors. Each fault vector is given by  $f^{(l)}(t) = [f_x^{(l)}(t) \ f_z^{(l)}(t)]^\top = [f_1^{(l)}(t), \dots, f_{n_l}^{(l)}(t)]^\top$ , where  $f_k^{(l)}(t), k \in \{1, \dots, n_l\}$  denotes the change in the output due to a fault in the  $k$ -th hardware sensor. Permanent abrupt offset faults can be modelled as follows [Reppa et al. (2016)]:

$$f_k^{(l)}(t) = \begin{cases} 0, & t < T_{f_k}^{(l)} \\ \hat{\phi}_k^{(l)}, & t \geq T_{f_k}^{(l)}, \end{cases} \quad (3)$$

where  $T_{f_k}^{(l)}$  is the time instant of occurrence of the  $k$ -th fault and  $\hat{\phi}_k^{(l)}$  is its constant fault signature.

## 2.2 Design of software-based sensors

Based on previous work [Kougiatsos and Reppa (2022)], three types of software-based sensors can be constructed based on model-information; dynamic software-based sensors from differential equations, static software-based sensors from explicit algebraic equations and SIVIA software-based sensors from implicit algebraic equations. More precisely, the software-based sensors are modelled inside the modules of the monitoring agents as shown in Figure 3 and are expressed as follows:

**Dynamic software-based sensors** are designed based on nonlinear estimators as follows:

$$\begin{cases} \hat{\dot{x}}^{(l,q)} = A^{(l)}\hat{x}^{(l,q)} + \gamma^{(l)}(\hat{x}^{(l,q)}, y_{zH}^{(l)}, u^{(l)}) + h^{(l)}(\hat{x}^{(l,q)}, y_{zH}^{(l)}, y_{\chi H}^{(l)}, u^{(l)}) + L^{(l,q)}(y_x^{(l,q)} - \hat{y}_x^{(l,q)}) + \Omega^{(l,q)}\hat{f}_x^{(l,q)} & (4a) \\ \Omega^{(l,q)}\hat{f}_x^{(l,q)} & (4b) \\ \dot{\Omega}^{(l,q)} = A_L^{(l,q)}\Omega^{(l,q)} - L^{(l,q)} & (4c) \\ \hat{y}_x^{(l,q)} = \hat{x}^{(l,q)} + \hat{f}_x^{(l,q)} & (4d) \\ \dot{\hat{f}}_x^{(l,q)} = \Gamma^{(l,q)}(\Omega^{(l,q)} + 1)\mathcal{D}[\epsilon_{y_x}^{(l,q)}], & (4e) \end{cases}$$

where  $L^{(I,q)}$  is the estimator gain such that  $A_L^{(I,q)} \triangleq A^{(I)} - L^{(I,q)}$  is Hurwitz,  $\Gamma^{(I,q)}$  is the learning rate of the adaptive law in (4e) and  $\Omega^{(I,q)}$  is a filtering term to ensure the stability of the state-equation adaptive scheme. The healthy sensor measurements  $y_{xH}^{(I,q)}, y_{\chi H}^{(I,q)}$  are defined as follows [Kougiatsos and Reppa (2022)]:

$$\begin{cases} y_{zH}^{(I,q)} = y_z^{(I,q)} - \hat{f}_z^{(I,q)} \\ y_{\chi H}^{(I,q)} = y_\chi^{(I,q)} - \hat{f}_\chi^{(I,q)}, \end{cases} \quad (5)$$

where  $\hat{f}_z^{(I,q)}, \hat{f}_\chi^{(I,q)}$  are the estimations of the sensors faults. If no sensor faults are diagnosed, then  $y_{zH}^{(I,q)} = y_z^{(I,q)}$  and  $y_{\chi H}^{(I,q)} = y_\chi^{(I,q)}$ . Finally,  $\mathcal{D}[\cdot]$  is the dead-zone operator, used to activate the sensor fault identification, given as:

$$\mathcal{D}[\varepsilon_{y_x^{(I,j,q)}}] = \begin{cases} 0, & \text{if } D^{(I,j,q)}(t) = 0 \\ \varepsilon_{y_x^{(I,j,q)}}, & \text{if } D^{(I,j,q)}(t) = 1, \end{cases} \quad (6)$$

where  $\varepsilon_{y_x^{(I,j,q)}} \triangleq y_x^{(I,j,q)} - \hat{x}^{(I,j,q)} - \hat{f}_x^{(I,j,q)}$  and  $D^{(I,j,q)}$  denotes the binary decision of the monitoring module  $\mathcal{M}^{(I,j,q)}$  on the occurrence of sensor faults as part of the diagnosis process.

**Static software-based sensors:** If (1b) can be written in an explicit form  $z^{(I)}(t) = \xi_z^{(I)}(x^{(I)}(t), \chi^{(I)}(t), u^{(I)}(t))$ , the following nonlinear estimator can be used:

$$\begin{cases} \hat{z}^{(I,q)} = \hat{\xi}_z^{(I)}(y_{xH}^{(I,q)}, y_{\chi H}^{(I,q)}, u^{(I,j)}) \\ \hat{f}_z^{(I,q)} = (y_z^{(I,q)} - \hat{z}^{(I,q)})D^{(I,q)}, \end{cases} \quad (7a)$$

$$(7b)$$

where  $y_{xH}^{(I,q)} = y_x^{(I,q)} - \hat{f}_x^{(I,q)}$ .

**SIVIA software-based sensors:** For all other cases (implicit algebraic equations), the use of SIVIA [Jaulin and Walter (1993)] is proposed. The rationale behind SIVIA is the identification of sets of nonlinear functions with the guaranteed property of convergence. Using (1b), the following nonlinear estimator can be constructed:

$$0 = \xi^{(I,q)}(x^{(I,q)}, \hat{z}^{(I,q)}, \chi^{(I,q)}, u^{(I)}) \quad (8)$$

If the intervals of values for the parameters  $x^{(I,q)}, \chi^{(I,q)}, u^{(I)}$  are known, SIVIA can calculate the unknown interval  $[\hat{z}^{(I,q)}]_s \triangleq [\underline{\hat{z}}_s^{(I,q)}, \bar{\hat{z}}_s^{(I,q)}] = y_z^{(I,q)} + [\hat{f}_z^{(I,q)}] + [d_z^{(I,q)}]$  through numerical iterations with proved convergence [Jaulin and Walter (1993)]. After some mathematical manipulations, the unknown interval  $[\hat{f}_z^{(I,q)}] = [\underline{\hat{f}}_z^{(I,q)}, \bar{\hat{f}}_z^{(I,q)}]$  can be approximated as follows:

$$\begin{cases} \underline{\hat{f}}_z^{(I,q)} = y_z^{(I,q)} - \bar{d}_z^{(I,q)} - \bar{\hat{z}}_s^{(I,q)} \\ \bar{\hat{f}}_z^{(I,q)} = y_z^{(I,q)} - \bar{d}_z^{(I,q)} - \underline{\hat{z}}_s^{(I,q)}. \end{cases} \quad (9)$$

The estimates of the algebraic state and sensor fault are finally obtained as:

$$\begin{cases} \hat{z}^{(I,q)} = y_{zH}^{(I,q)} = y_z^{(I,q)} - \hat{f}_z^{(I,q)} \\ \hat{f}_z^{(I,q)} = \frac{\bar{\hat{f}}_z^{(I,q)} + \underline{\hat{f}}_z^{(I,q)}}{2} D^{(I,q)}. \end{cases} \quad (10a)$$

$$(10b)$$

The goal of this paper is to provide a switching methodology between hardware and software-based sensors in the case of one or multiple faults affecting the former. The proposed logic is designed so that certain control performance and operational criteria are met.

### 3 Mathematical modelling

Based on the general problem formulation (see Section 2) and the system partitioning shown on Figure 2, the exact models used on this paper are provided in the following subsections. We assume that the engines run in parallel [Geertsma et al. (2017c)] with speed control used for  $\Sigma^{(1)}$  and torque control used for  $\Sigma^{(2)}$ , as can be also seen in Figure 4.

### 3.1 Diesel engine-Gearbox & Shaftline-Propeller set ( $\Sigma^{(1)}$ )

The model of the diesel engine has already been described in [Kougiatsos et al. (2022)] and will thus be omitted. In summary, the fuel engine has already been partitioned in four subsystems; the intake manifold incorporating the fuel pump ( $\Sigma^{(1,1)}$ ), the thermomechanical process incorporating the dynamics of the engine shaft and the thermodynamics of the process inside the engine cylinders ( $\Sigma^{(1,2)}$ ), the exhaust gas path including the turbocharger and the exhaust manifold ( $\Sigma^{(1,3)}$ ) and the air path including the engine compressor and intercooler ( $\Sigma^{(1,4)}$ ). In order to account for the addition of the propeller and its additional shaft speed sensor, subsystem  $\Sigma^{(1,2)}$  of the engine will be appended with an extra state  $x^{(1,2)} = n_p$ , the propeller shaft rotational speed (in rps). The additional differential equation is the following [Kalikatzarakis et al. (2018)]:

$$\dot{x}^{(1,2)} = \begin{bmatrix} \frac{i_{gb}\eta_{sh}}{2\pi J_{tot}} & \frac{i_{gb}\eta_{sh}}{2\pi J_{tot}} \end{bmatrix} \cdot \begin{bmatrix} z_3^{(1,2)} \\ z_1^{(2,1)} \end{bmatrix} - \frac{c}{2\pi J_{tot}} \cdot (x^{(1,2)})^2, \quad (11)$$

where  $\eta_{sh}$  denotes the shaftline and gearbox mechanical efficiency,  $J_{tot}$  is the total moment of inertia describing the shaft line system in [kgm<sup>2</sup>],  $c$  is the propeller law constant and  $i_{gb}$  is the gearbox ratio. The propeller law can be expressed as:

$$x^{(1,2)} = \frac{1}{i_{gb}} \left( \frac{2\pi}{c} (z_3^{(1,2)} + z_1^{(2,1)}) \right)^{1/2}. \quad (12)$$

*Remark 1:* Equations (11) and (12) both serve as analytical redundancy relations for the shaft speed and can be used to create different types of software sensors (dynamic and static respectively).

The output value of the shaft speed sensor is then appended to the sensor outputs of the second subsystem in system 1 ( $\Sigma^{(1,2)}$ ) as:

$$S^{(1,2)} : y^{(1,2)} = \begin{bmatrix} x^{(1,2)} \\ z^{(1,2)} \end{bmatrix} + d^{(1,2)} + f^{(1,2)}, \quad (13)$$

where  $y^{(1,2)} \in \mathbb{R}^4$ .

Speed control of the marine fuel engine is done using a PI controller expressed as:

$$u^{(1,1)} = \min \left( X_{lim}, K_p^{(1,1)} e^{(1,2)}(t) + K_I^{(1,1)} \int_0^t e^{(1,2)}(\tau) d\tau \right), \quad (14)$$

where  $e^{(1,2)} = n_p^{ref} - y_1^{(1,2)}$  is the shaft speed reference tracking error,  $X_{lim}$  is the fuel injection limiter as mentioned in [Geertsma et al. (2017c)] and  $K_p^{(1,1)}, K_I^{(1,1)}$  are the gains of the fuel engine speed controller.

### 3.2 Electric motor ( $\Sigma^{(2)}$ )

A suitable algebraic model describing the output torque  $z^{(2,1)}$  in [Nm] of the induction motor is the following [Wildi (2002)]:

$$\Sigma^{(2,1)} : z_1^{(2,1)} = \frac{p}{4\pi i_{gb} x^{(1,2)}} \frac{\left( u^{(2,1)} \right)^2}{\left( \frac{R_2}{s} \right)^2 + \left( \frac{i_{gb} x^{(1,2)}}{2\pi} (H_s + H_r) \right)^2} \frac{R_r}{s}, \quad (15)$$

where  $p$  denotes the number of poles,  $n_{IM}$  is the rotational speed of the motor shaft in [rps],  $u^{(2,1)}$  is the control value expressing the input voltage,  $s$  is the slip,  $R_r$  is the rotor resistance in [Ohm] and  $H_s, H_r$  are the stator and rotor reluctance in [H] respectively. The output of the induction motor torque sensor  $y^{(2,1)} \in \mathbb{R}$  is described by:

$$S^{(2,1)} : y^{(2,1)} = z^{(2,1)} + d^{(2,1)} + f^{(2,1)}. \quad (16)$$

Torque control of the induction motor is accomplished using a PI controller scheme as follows [Vahedpour et al. (2015)]:

$$u^{(2,1)} = C_{V/f} \left( 2\pi i_{gb} y^{(2,1)} + K_p^{(2,1)} e^{(2,1)}(t) + K_I^{(2,1)} \int_0^t e^{(2,1)}(\tau) d\tau \right), \quad (17)$$

where  $e^{(2,1)} = Q^{ref} - y^{(2,1)}$  is the torque reference tracking error,  $K_p^{(2,1)}, K_I^{(2,1)}$  are the gains of the induction motor torque controller and  $C_{V/f}$  is a constant.

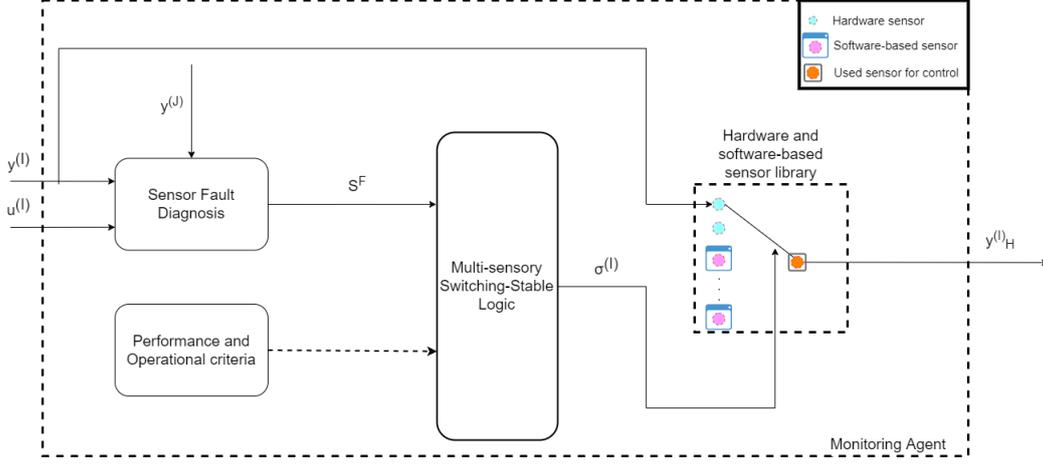


Figure 3: Internal structure of monitoring agents

#### 4 Multi-sensory switching-stable mechanism

In Figure 3, the internal structure of the monitoring agents is shown. The measurements (local and transmitted from neighboring systems) from the available hardware sensors are used to perform sensor fault diagnosis and determine the healthy and faulty sensor sets. These sets are then taken into account by the multi-sensory switching logic alongside performance and possible operational constraints. The output of the switching logic is a signal  $\sigma^{(I)}$  which denotes the index of the sensor from a library containing both hardware-based sensors physically present in the system  $\Sigma^{(I)}$  and software-based sensors that are realised internally in the monitoring agents.

Then, in Figure 4, the proposed distributed parallel control architecture is shown. The monitoring agents of each system  $\Sigma^{(I)}$ ,  $I = \{1, 2\}$  provide decisions on which sensor to use from the set of available hardware and software-based sensors belonging to the system. The monitoring agents can communicate with each other in a distributed configuration, mimicking the physical interconnections between the two subsystems. The controllers of the fuel engine and the electric motor are realised in a decentralised configuration and use the feedback sensors dictated by the monitoring agents. The next section discusses the switching logic between hardware and software-based sensors during the propulsion plant operation.

##### 4.1 Control performance criteria

Based on the decisions of the designed monitoring agents for the fuel engine and electric motor subsystems [Kougiatsos et al. (2022)], we can divide the sensors used in the system in two sets: the healthy ( $S^H$ ) and the faulty sensor set ( $S^F$ ). These sets are in general time-varying during the sensor fault diagnosis operation.

The switching mechanism makes a decision on which sensor to use, expressed by the switching signals  $\sigma^{(1)} \in \mathbb{Z}_+^{(1,2)}$  and  $\sigma^{(2)} \in \mathbb{Z}_+^{(2,1)}$  respectively. The hardware sensors correspond to  $\sigma^{(1)} = \sigma^{(2)} = 1$ . In case  $S^{(1,2)}\{1\} \in S^F$  or  $S^{(2,1)}\{1\} \in S^F$  the switching signals  $\sigma^{(1)}, \sigma^{(2)}$  should change. The switching laws are designed to achieve the minimum tracking error and can be expressed as follows [Kodakkadan et al. (2016)]:

$$\sigma^{(1)}(t) = \begin{cases} \arg \min_{k \in \mathbb{Z}_+^{(1,2)}} |y_{(1,k)}^{(1,2)}(t) - n_p^{\text{ref}}|, & S^{(1,2)}\{1\} \in S^F \\ 1, & S^{(1,2)}\{1\} \notin S^F, \end{cases} \quad (18)$$

$$\sigma^{(2)}(t) = \begin{cases} \arg \min_{k \in \mathbb{Z}_+^{(2,1)}} |y_{(1,k)}^{(2,1)}(t) - Q^{\text{ref}}|, & S^{(2,1)}\{1\} \in S^F \\ 1, & S^{(2,1)}\{1\} \notin S^F, \end{cases} \quad (19)$$

where  $n_p^{\text{ref}}$  is the shaft speed reference signal,  $Q^{\text{ref}}$  is the electric motor torque reference, both provided by a higher secondary level controller (e.g., a power split controller) [Geertsma et al. (2017a)],  $y_{(1,k)}^{(1,2)}$ ,  $k \in \mathbb{Z}_+^{(1,2)} = \{1, 2, 3\}$  denotes the available hardware sensor ( $k = 1$ ) based on (13), dynamic software sensor ( $k = 2$ ) based on (11) and (4b) and static software sensor ( $k = 3$ ) based on (12) and (7a) that can be used for the shaft speed measurement and  $y_{(1,k)}^{(2,1)}$ ,  $k \in \mathbb{Z}_+^{(2,1)} = \{1, 2\}$  denotes the available hardware ( $k = 1$ ) based on (16) and static software sensor ( $k = 2$ ) based on (15) and (7a) that can be used for the electric motor torque measurement.

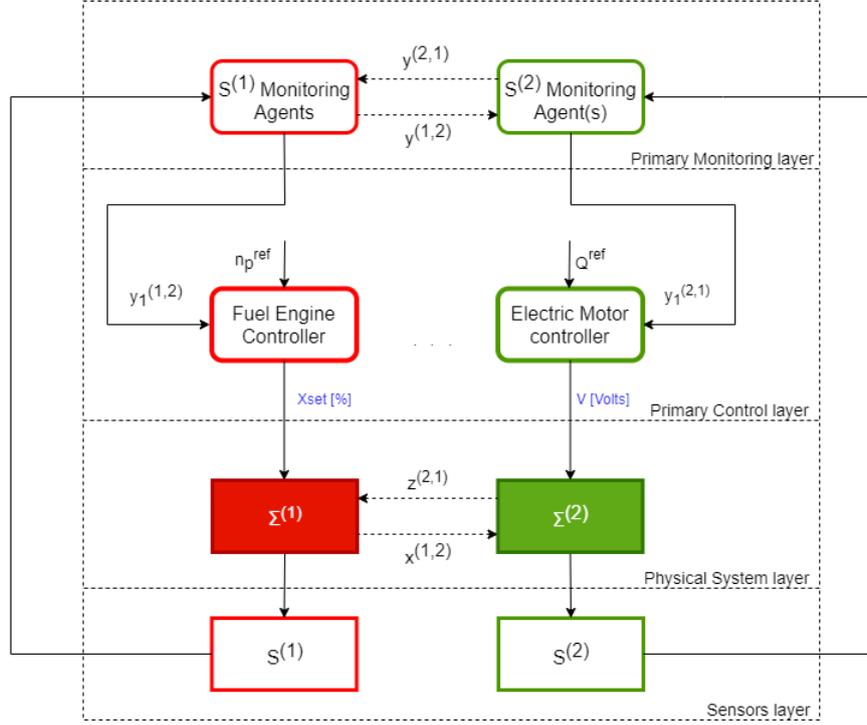


Figure 4: Multisensory switching hybrid propulsion control architecture

*Remark 2:* The control performance criteria expressed in (18), (19) may lead to control instability due to frequent switching of the used sensor channel. To avoid those cases, a dwell time between consecutive switches [Stoican et al. (2014)] is used in this paper. The 'design' of this dwell time will be investigated in future research.

## 4.2 Operational criteria

In the case of a marine fuel engines, certain regions of performance (most commonly referred to as Ranges of Operation) can be defined based on the operational envelope of the engine. For the specific engine model, the operational envelope is shown in Figure 5. The horizontal axis corresponds to the engine shaft speed measurement ( $i_{gb} \cdot y_1^{(1,2)}$ ) in [RPM] while the vertical axis is the produced power of the engine per cylinder ( $i_{gb} \cdot y_1^{(1,2)} \cdot y_4^{(1,2)} / i_e$ ) in [kW/cyl], where  $i_e$  is the number of cylinders. According to the project guide [SE (2008)] the engine operates in Range I under normal conditions. In case the operational point belongs to Range II, it is strongly advised to only operate the engine there for a maximum of 1 min and only for acceleration or manoeuvring operations. Finally, in Range III, operation is permitted up to 12 h. After this maximum allowable time durations have elapsed, the fuel engine can become unstable and multiple system failures can occur.

Regarding the design efficiency of software-based sensors, some preliminary results have already been extracted from previous work [Kougiatsos and Reppa (2022)]. More precisely, the use of dynamic software-based sensors produces relatively accurate results with minimal noise effect due to the filtering ability of the adaptive observers. However, the adaptive nature of the underlying nonlinear observers entails a time-lag for the convergence of the fault estimation process. On the other hand, in the case of static software-based sensors, convergence is fast but the healthy measurements are affected by noise which could deteriorate the system performance due to the use of PI controllers in the multi-sensory control scheme. Finally, for SIVIA software-based sensors, convergence of the numerical method requires more time than the dynamic case, with overshoots and undershoots that continuously diminish as the estimation process continues.

Using the above information, we propose the following guidelines for the use of the software-based sensors instead of their hardware counterparts:

- When the hardware shaft speed ( $S^{(1,2)}\{1\}$ ) and fuel engine torque ( $S^{(1,2)}\{4\}$ ) sensors indicate operation in Regions I or III and one or more of them belong to  $S^F$ , accuracy of the estimation should be prioritised (e.g., using dynamic software-based sensors)
- When the hardware sensors  $S^{(1,2)}\{1\}$ ,  $S^{(1,2)}\{4\}$  indicate operation in Regions II and one or more of them belong to  $S^F$ , the time needed for the estimation should be minimal to satisfy the operational time constraints (e.g., using static software-based sensors)

Combining the operational with the performance criteria, the complete switching logic is summarised in Algorithm 1.

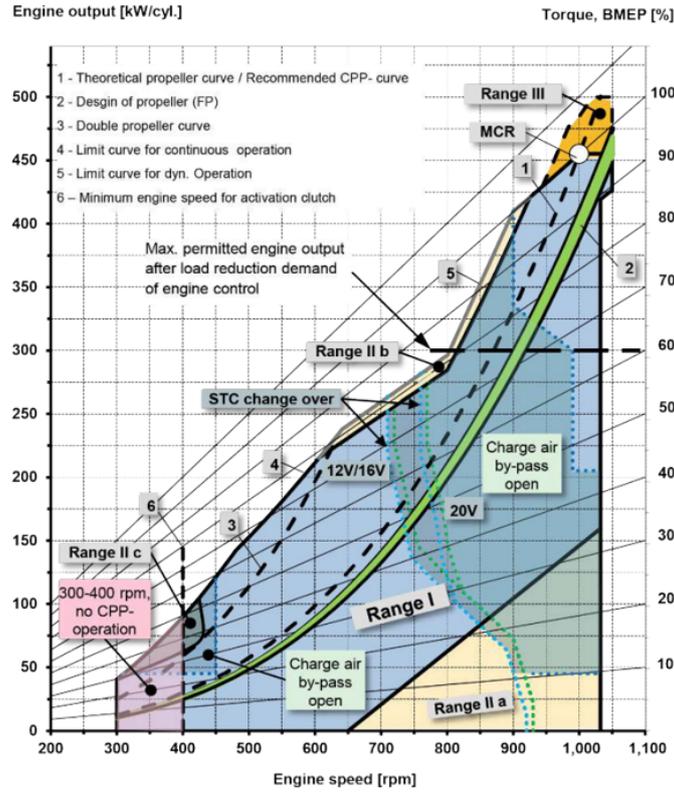


Figure 5: MAN V28/33D STC Fuel engine operational envelope [SE (2008)]

## 5 Simulation experiments and discussion

In this section, we apply the multi-sensory switching methodology described in Section 4 to a hybrid marine propulsion system using the parallel control approach described in [Geertsma et al. (2017c)]. The parameters for the fuel engine were extracted from [Geertsma et al. (2017c)] and for the electric motor, shaftline and gearbox from [Geertsma et al. (2017b)]. The dwell time for the switching logic is set at 0.5 s. The rest of parameters are assumed as follows:  $\eta_{sh} = 0.95$ ,  $c = 8 \cdot 10^5$ ,  $K_p^{(1,1)} = 2$ ,  $K_I^{(1,1)} = 40$ ,  $K_p^{(2,1)} = 1$ ,  $K_I^{(2,1)} = 0.2$ ,  $C_{V/f} = 2$ .

It is assumed that the measurements of each sensor of the two systems are corrupted by uniformly distributed noise with  $d_j^{(l)}$  being 3 % of the amplitude of the noiseless measurements of the sensor. We have simulated permanent abrupt offset sensor faults affecting the shaft speed sensor  $S^{(1,2)}\{1\}$  and the electric motor torque sensor  $S^{(2,1)}\{1\}$ . For the diagnosis of sensor faults, the distributed methodology described in [Kougiatsos et al. (2022)] is employed. Moreover, a dynamic software-based sensor for the shaft speed and two static software-based sensors for the shaft speed and electric motor's torque output are created based on [Kougiatsos and Reppa (2022)]. The design gains of the various monitoring modules corresponding to the marine fuel engine are selected as  $L^{(1,1)} = 1.16$ ,  $L^{(1,2)} = 10$ ,  $L^{(1,3)} = 445$ ,  $L^{(1,4)} = 319.98$  while the learning rates for the design of the dynamic software-based sensors are selected as  $\Gamma^{(1,1)} = 0.5$ ,  $\Gamma^{(1,2)} = 100$ ,  $\Gamma^{(1,3)} = 8.7$ ,  $\Gamma^{(1,4)} = 5$ . As for the electric motor, since it's described by a static model and only a static software-based sensor is created, no gains or learning rates are applicable.

In the experiment, the system has a control objective to achieve a reference power profile with a magnitude of  $P_D = 9400$  kW. The initial conditions for the fuel engine are set to the maximum continuous rating (MCR) point while for the electric motor, a starting speed equal to 16.71 rps is assumed. As the derivation of a suitable power split strategy is out of the scope of this paper, it is assumed that the fuel engine delivers a maximum of 5400 kW (corresponding to the maximum allowable power level to prevent overloading) while the electric motor is expected to deliver the rest of the requested propulsion power. This results to the following reference signals for  $\Sigma^{(1)}$  and  $\Sigma^{(2)}$ :

$$n_p^{ref} = \left( \frac{P_D}{c} \right)^{1/3} \quad (20)$$

---

**Algorithm 1** Multi-sensory switching-stable logic for control of hybrid marine vessels

---

**Input:**  $S^F$  ▷ From sensor fault diagnosis agents  
**Output:**  $\sigma^{(1)}, \sigma^{(2)}$  ▷ Indices of the 'to be used' sensors

- 1:  $y_1^{(1,2)} \leftarrow$  Shaft speed measurement (rps)
- 2:  $y_4^{(1,2)} \leftarrow$  Fuel engine torque measurement (Nm)
- 3:  $y_1^{(2,1)} \leftarrow$  Electric motor torque measurement (Nm)
- 4:  $i_e \leftarrow$  Number of engine cylinders
- 5:  $M \leftarrow (i_{gb} \cdot y_1^{(1,2)}, i_{gb} \cdot y_1^{(1,2)} \cdot y_4^{(1,2)} / i_e)$  ▷ Fuel engine operational point
- 6:  $\sigma^{(1)} \leftarrow 1$  ▷ Hardware sensors in use
- 7:  $\sigma^{(2)} \leftarrow 1$
- 8: **if**  $S^{(1,2)}\{1\} \in S^F$  **then**
- 9:     **if**  $M \notin \text{Region II}$  **then**
- 10:         Compute  $\sigma^{(1)}$  from (18) ▷ Control Performance criteria
- 11:     **else**
- 12:          $\sigma^{(1)} \leftarrow 3$  ▷ Operational criteria
- 13:     **end if**
- 14: **end if**
- 15: **if**  $S^{(2,1)}\{1\} \in S^F$  **then**
- 16:     Compute  $\sigma^{(2)}$  from (19) ▷ Control Performance criteria
- 17: **end if**

---

$$Q^{ref} = \frac{P_D - 5400}{2\pi i_{gb} n_p^{ref}} \quad (21)$$

Two fault scenarios are simulated in order to showcase the use of both control performance (scenario 1) and operational (scenario 2) criteria in the switching logic (see Algorithm 1). The magnitude of the sensor faults for the first scenario are chosen as  $\hat{\phi}_1^{(1)} = 2$  rps and  $\hat{\phi}_1^{(2)} = 5 \cdot 10^4$  Nm and their times of occurrence at  $T_{f1}^{(1)} = 20$  s,  $T_{f1}^{(2)} = 50$  s respectively. The results corresponding to this scenario are shown in Figures 6 and 7. As observed in Figure 6(a), the sensor fault injected in  $\Sigma^{(1)}$  is detected as the decision of the corresponding monitoring module becomes 1. As a result, the switching logic is activated, and the hardware sensor corresponding to the index '1' is swapped off with the dynamic and static software-based sensors with indices '2' and '3' respectively. At first, the static software-based sensor is preferred as the dynamic one needs some time to converge to the healthy sensor value. After some time, the dynamic software sensor is used due to better tracking performance and for the same reason an alternating behavior can be seen between the two software sensors. However, the dwell-time is respected during consecutive switches and control stability is maintained. This is also presented in Figure 6(c) where the good tracking behavior of the shaft speed reference signal is shown. The reference shaft speed is achieved with a Mean Absolute Percentage Error (MAPE) of 0.47 %. In addition, Figure 6(b) shows the switching logic assigned to  $\Sigma^{(2)}$ . Again the sensor fault is detected at its time of occurrence and the hardware sensor with index '1' is swapped with the designed static software sensor with index '2'. Control stability in this system is also maintained, with a good resulting motor reference torque tracking performance that can be seen in Figure 6(d) and corresponds to a MAPE of 0.001 %. It can also be observed that the sensor switching in  $\Sigma^{(1)}$  is also propagated in this system especially between 20 and 25 sec. This is attributed to the existing system monitoring agent interconnections. Moreover, in Figure 7 the reference power level tracking performance is presented as a Key Performance Indicator characterising the multi-sensory switching performance with regard to the whole propulsion system. A MAPE of just 1.39 % in power tracking is achieved for the first simulation scenario.

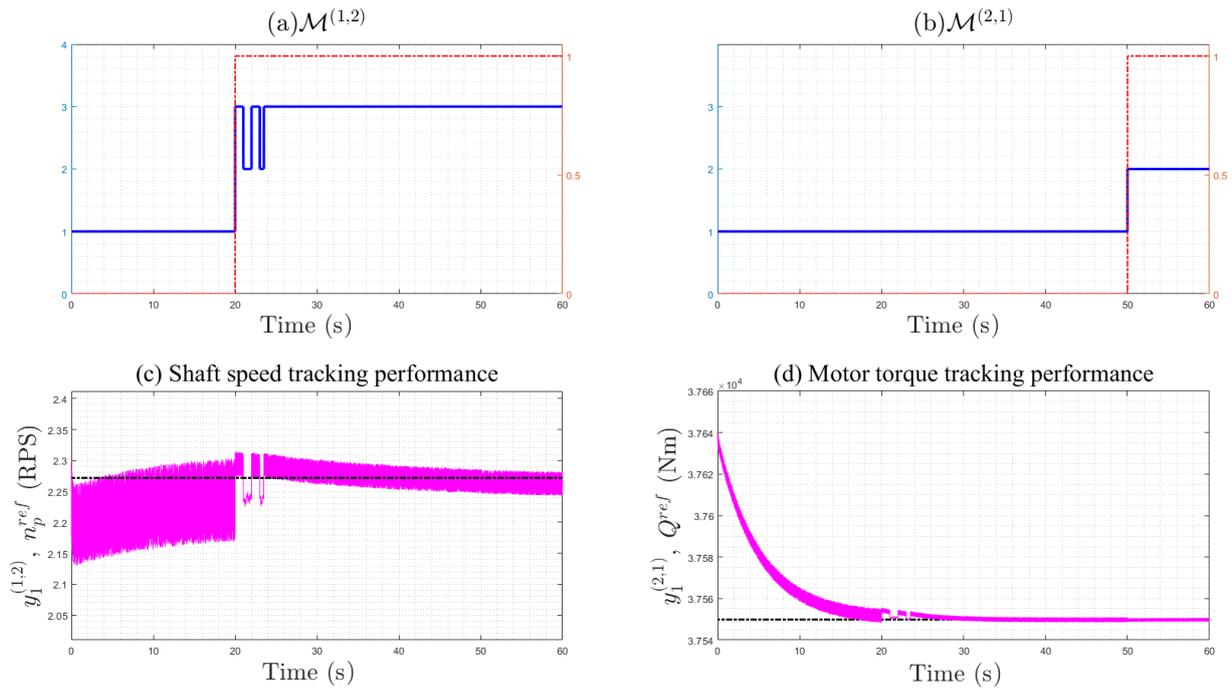


Figure 6: (a),(b) Multi-sensory switching logic (blue: switching signals  $\sigma^{(1)}$  and  $\sigma^{(2)}$ , red/ dash-dotted: sensor fault binary decision logic) and (c),(d) its effect on local system performance (magenta: hardware/ software sensor values, black/ dash-dotted: reference values) under scenario 1

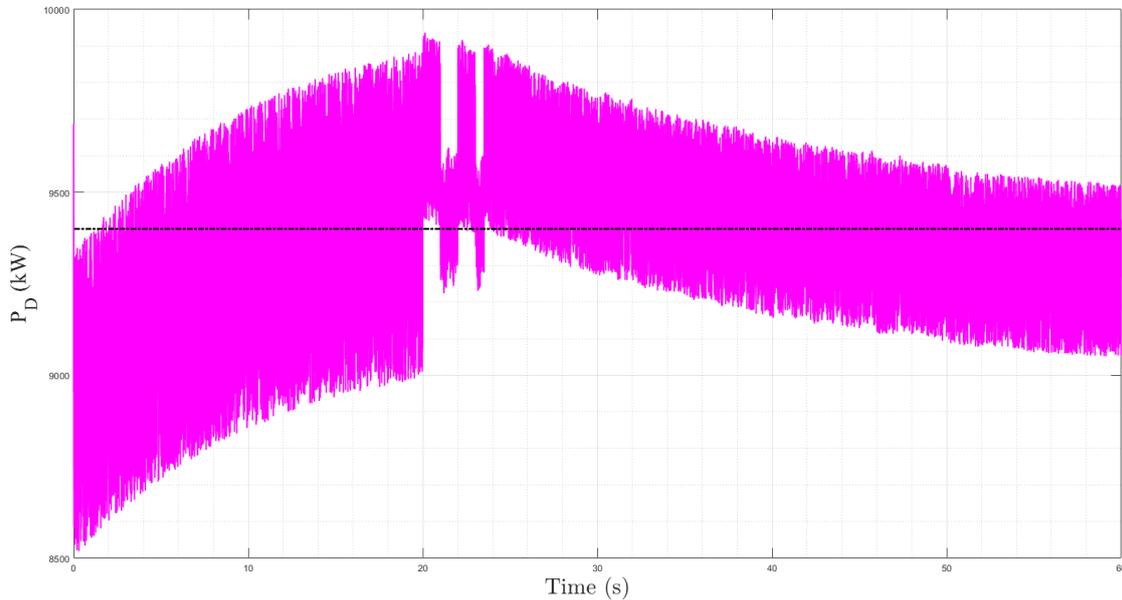


Figure 7: Multi-sensory switching logic effect on global system/ power profile tracking performance under scenario 1 (magenta: measured power, black/ dash-dotted: reference/ requested power level)

In scenario 2, only the magnitude of the sensor fault in the shaft speed sensor changes to  $\hat{\phi}_1^{(1,2)} = -0.5$  rps to allow us to simulate the operation in Region II in the operational envelope (see Figure 5). The simulation results are shown in Figures 8 and 9. As can be seen from Figures 8(a),(b), the sensor faults are detected by the monitoring agents but this time  $\sigma^{(1)}$  changes from the value '1' (hardware sensor) permanently to the value '3' (static shaft speed software sensor) due to the effect of the operational criteria in Algorithm 1, as shown in Figure 8(a). The

tracking performance of the shaft speed and torque reference signals is also satisfactory in this case, with resulting MAPEs of 0.48 % and 0.001% respectively, as shown in Figures 8(c),(d). Finally, in Figure 9, the power reference tracking performance is illustrated. The tracking error in this case corresponds to a resulting MAPE of 1.42 %.

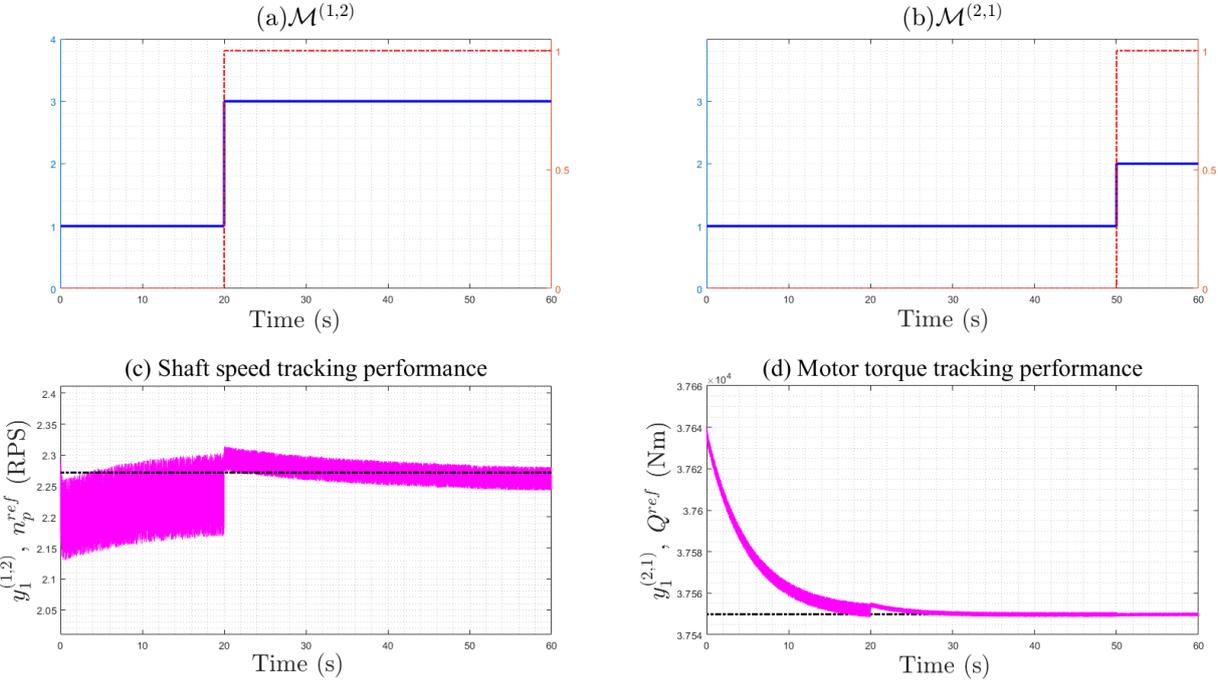


Figure 8: (a),(b) Multi-sensory switching logic (blue: switching signals  $\sigma^{(1)}$  and  $\sigma^{(2)}$ , red/ dash-dotted: sensor fault binary decision logic) and (c),(d) its effect on local system performance (magenta: hardware/ software sensor values, black/ dash-dotted: reference values) under scenario 2 ,

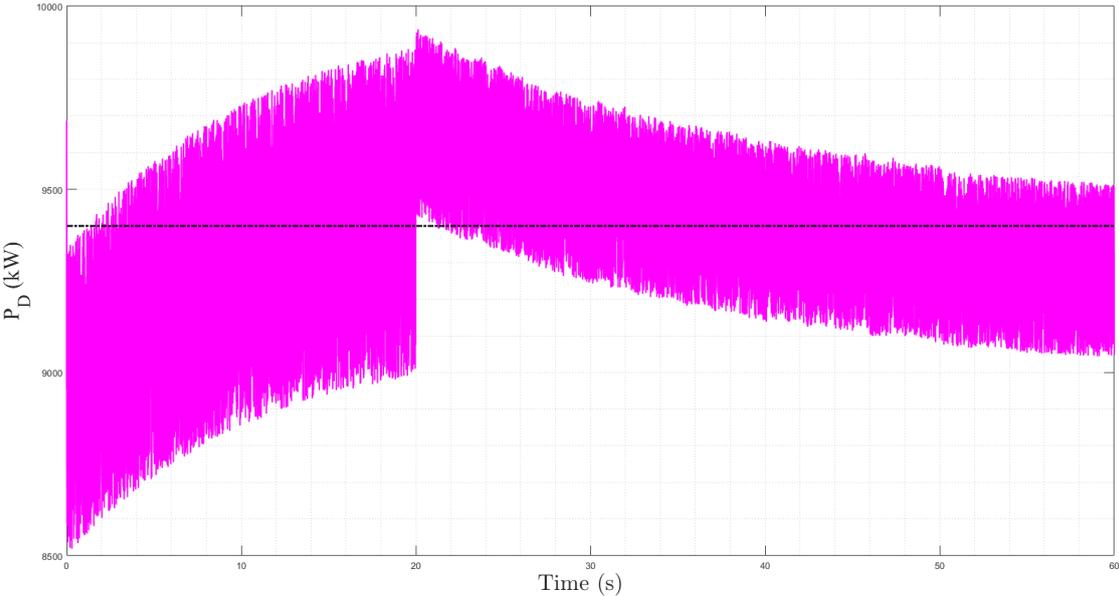


Figure 9: Multi-sensory switching logic effect on global system/ power profile tracking performance under scenario 2 (magenta: measured power, black/ dash-dotted: reference/ requested power level)

Based on the above results, the multi-sensory switching logic is deemed to have a satisfactory performancee

with minimal tracking errors of the reference signals fed in the hybrid propulsion control system. Moreover, multiple sensor faults affecting the system sensors used for control are properly compensated and both system and control stability are maintained.

## 6 Conclusions and further research

This paper proposed a multi-sensory switching methodology for use in hybrid marine vessel propulsion installations. The system consisted of a marine fuel engine running in parallel with an induction motor, both coupled through the shaftline to the gearbox and the propeller. First we decomposed the system in two subsystems, one containing the fuel engine, the propeller and the shaftline/gearbox and the second one only including the induction motor. Parallel control has been assumed in this case study with the marine fuel engine running in speed control and the electric motor running in torque control. Multiple heterogeneous hardware sensors were assumed installed in the examined systems, with two of them used for control purposes. Instead of relying on hardware redundancy in case of sensor faults, this paper proposed the combination of hardware and software-based sensors in a multi-sensory scheme, taking advantage of the analytical redundancy of the coupled fuel engine and electric motor.

The multi-sensory switching logic was then designed to make decisions on which sensor to use so that the reference tracking error can be minimal for both engine controllers, resulting in more stable control of the vessel's engines. Moreover, in the case of the marine fuel engine, the associated operational criteria mainly expressed by the time duration the engine can remain in the different operational ranges, were also considered. The preliminary results of this paper indicate a good performance of the switching logic under multiple faults occurring at sensors used for propulsion control and at different time instants. The methodology can as well be expanded with more software-based sensors as the propulsion plant realisations become more complex.

Future work will address possible control stability issues during consecutive switches in the sensor feedback channel. In addition, the behavior of the switching logic during transients in the propulsion will be investigated.

## Acknowledgement

This publication is part of the project READINESS with project number TWM.BL.019.002 of the research programme "Topsector Water & Maritime: the Blue route" which is partly financed by the Dutch Research Council (NWO).

## References

- Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2016. *Diagnosis and Fault-Tolerant Control* (Third Edition). Springer Berlin, Heidelberg.
- Campa, G., Thiagarajan, M., Krishnamurty, M., Napolitano, M.R., Gautam, M., 2008. A neural network based sensor validation scheme for heavy-duty diesel engines. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME* 130, 0210081–02100810.
- Darvishi, H., Ciunzo, D., Eide, E.R., Rossi, P.S., 2021. Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture. *IEEE Sensors Journal* 21, 4827–4838.
- DNV, 2015. *Dynamic positioning vessel design philosophy guidelines*. DNV GL Recommended Practice, DNVGL-RP-E306, July 2015.
- European Maritime Safety Agency, 2020. *Annual Overview of Marine Casualties and Incidents 2020*, 12–24.
- Geertsma, R.D., Negenborn, R.R., Visser, K., Hopman, J.J., 2017a. Design and control of hybrid power and propulsion systems for smart ships: A review of developments. *Applied Energy* 194, 30–54.
- Geertsma, R.D., Negenborn, R.R., Visser, K., Hopman, J.J., 2017b. Parallel Control for Hybrid Propulsion of Multifunction Ships. *IFAC-PapersOnLine* 50, 2296–2303.
- Geertsma, R.D., Negenborn, R.R., Visser, K., Loonstijn, M.A., Hopman, J.J., 2017c. Pitch control for ships with diesel mechanical and hybrid propulsion: Modelling, validation and performance quantification. *Applied Energy* 206, 1609–1631.
- Jaulin, L., Walter, E., 1993. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica* 29, 1053–1064.
- Kalikatzarakis, M., Geertsma, R.D., Boonen, E.J., Visser, K., Negenborn, R.R., 2018. Ship energy management for hybrid propulsion and power supply with shore charging. *Control Engineering Practice* 76, 133–154.
- Kerrigan, E.C., Maciejowski, J.M., 2015. Fault-tolerant control of a ship propulsion system using model predictive control. In *Proceedings of the 1999 European Control Conference, Karlsruhe, Germany*, 4602–4607.
- Kodakkadan, A.R., Reppa, V., Oлару, S., 2016. Switching-stable control mechanism in the presence of guaranteed detectable sensor faults. In *Proceedings of the conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain*, 93–98.

- Kougiatsos, N., Negenborn, R.R., Reppa, V., 2022. Distributed model-based sensor fault diagnosis of marine fuel engines. *IFAC-PapersOnLine* 55, 347–353. 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022, Paphos, Cyprus.
- Kougiatsos, N., Reppa, V., 2022. A Distributed virtual sensor scheme for marine fuel engines. To appear in *IFAC CAMS 2022*, Kongens Lyngby, Denmark, September, 2022.
- Li, T., Hua, M., Yin, Q., 2019. The temperature forecast of ship propulsion devices from sensor data. *Information* 10(10), 316 .
- Li, Z., Yan, X., Guo, Z., Liu, P., Yuan, C., Peng, Z., 2012. A new intelligent fusion method of multi-dimensional sensors and its application to tribo-system fault diagnosis of marine diesel engines. *Tribology Letters* 47, 1–15.
- Planakis, N., Papalambrou, G., Kyrtatos, N., 2021. Predictive power-split system of hybrid ship propulsion for energy management and emissions reduction. *Control Engineering Practice* 111, 104795.
- Raptodimos, Y., Lazakis, I., Theotokatos, G., Varelas, T., Drikos, L., 2016. Ship sensors data collection & analysis for condition monitoring of ship structures & machinery systems. *RINA, Royal Institution of Naval Architects - Smart Ship Technology 2016, Papers* , 77–86.
- Reppa, V., Polycarpou, M.M., Panayiotou, C.G., 2016. Sensor fault diagnosis. *Foundations and Trends in Systems and Control* 3, 1–248.
- SE, M.D., 2008. Project guide for marine plants diesel engine 28/33d. Technical Report. Augsburg, Germany.
- Stoican, F., Olaru, S., Seron, M.M., De Doná, J.A., 2014. A fault tolerant control scheme based on sensor-actuation channel switching and dwell time. *International Journal of Robust and Nonlinear Control* 24, 775–792.
- Vahedpour, M., Noei, A.R., Kholerdi, H.A., 2015. Comparison between performance of conventional, fuzzy and fractional order pid controllers in practical speed control of induction motor, in: *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 912–916.
- Wildi, T., 2002. *Electrical Machines, Drives, and Power Systems*. Pearson Education Limited.
- Wu, N.E., Thavamani, S., Zhang, Y.M., Blanke, M., 2006. Sensor fault masking of a ship propulsion system. *Control Engineering Practice* 14, 1337–1345.