

# Collaborative prognostics in Social Asset Networks

Adrià Salvador Palau<sup>a,\*</sup>, Zhenglin Liang<sup>a</sup>, Daniel Lütgehetmann<sup>b</sup>,  
Ajith Kumar Parlikad<sup>a</sup>

<sup>a</sup>*Department of Engineering, Institute for Manufacturing, University of Cambridge,  
Cambridge, CB3 0FS, UK*

<sup>b</sup>*Institut für Mathematik, AG Topologie, Arnimallee 7, Raum 212, 14195 Berlin, Germany*

---

## Abstract

With the spread of Internet of Things (IoT) technologies, assets have acquired communication, processing and sensing capabilities. In response, the field of Asset Management has moved from fleet-wide failure models to individualised asset prognostics. Individualised models are seldom truly distributed, and often fail to capitalise the processing power of the asset fleet. This leads to hardly scalable machine learning centralised models that often must find a compromise between accuracy and computational power. In order to overcome this, we present a novel theoretical approach to collaborative prognostics within the Social Internet of Things. We introduce the concept of Social Asset Networks, defined as networks of cooperating assets with sensing, communicating and computing capabilities. In the proposed approach, the information obtained from the medium by means of sensors is synthesised into a Health Indicator, which determines the state of the asset. The Health Indicator of each asset evolves according to an equation determined by a triplet of parameters. Assets are given the form of the equation but they ignore their parametric values. To obtain these values, assets use the equation in order to perform a non-linear least squares fit of their Health Indicator data. Using these estimated parameters, they are interconnected to a subset of collaborating assets by means of a similarity metric. We show how by simply interchanging their estimates, networked assets are able to precisely determine their Health Indicator dynamics and reduce maintenance costs. This is done in real time, with no centralised library, and without the need for extensive historical data. We compare Social Asset Networks with the typical self-learning and fleet-wide approaches, and show that Social Asset Networks have a faster convergence and lower cost. This study serves as a conceptual proof for the potential of collaborative prognostics for solving maintenance problems, and can be used to justify the implementation of such a system in a real industrial fleet.

*Keywords:* Maintenance; Prognostics; Networks; SIoT; Asset Management.

---

\*Corresponding author

*Email address:* [as2636@cam.ac.uk](mailto:as2636@cam.ac.uk) (Adrià Salvador Palau)

## 1. Introduction

Asset health and performance management can be summarised as the task of managing assets in order to reduce the risk of failures and performance reduction while minimising cost [1]. In recent decades, this field of study has received attention due to the increased focus on services undertaken by manufacturing firms [2], and the rapid spread of Internet of Things technologies in industrial systems [3, 4, 5, 6, 7]. Such technologies allow assets to record, store and process data in situ, and in some cases also allow them to undertake independent actions [8]. The connection between asset management and the Social Internet of Things can be drawn on previous work by our group, where we presented the fundamental building blocks for the SIOIT (Social Internet of Industrial Things). We named one of these blocks “Social Assets”, defined as assets with an assigned software agent augmenting them with intelligent and social behaviours [7]. In this paper we present a novel approach for collaborative prognostics by extending the concept of Social Assets to Social Asset Networks which correspond to the study of the dynamics of a network of such assets coupled to an asset health management policy.

A prolific field of health management is Prognostics, which is defined as “Predictive Diagnostics which includes determining the remaining useful life or time span of useful operation for a component” [9]. Effective Prognostics depend on a combination of precise evaluation of the current asset state together with reliable prediction models [10]. Prognostics have become increasingly feasible over the last decade, in which deployment of networked sensors has become the norm in many maintenance-intensive industries [11, 12]. This has enabled the development of Condition Based Maintenance, where the sensor data is gathered and studied in order to obtain a continuous assessment of the state of the asset [10]. Such state is often characterised using a synthetic variable known as Health Indicator, which is formed by a combination of inputs of the many sensors installed in the asset [13, 14].

The definition of a Health Indicator, as well as the choice of the model used for prognostics, is tied to multiple layers of error and uncertainty. For example, a typical method for sensor selection, multiple regression, relies heavily on assumptions such as linear independence of the variables, and always carries associated uncertainties. Prognostics models often rely on a limited sample which is taken to be representative of the population, and from which a general model is inferred. Regardless of these limitations, for assets with a well-understood deterioration process, it is possible to find a parametric model that can be used to predict the future state of the Health Indicator [15]. In this paper, we study assets of this kind, where the time series of the Health Indicator is defined by an inverse exponential model. This is a popular model, as it presents an initially slow decay followed by a period of faster deterioration [16].

Systems composed of several assets are referred to as multi-unit or multi-component systems. Multi-component systems are those treated as a multiplicity of assets forming a larger asset (for example, airplanes). Alternately, multi-unit systems usually refer to fleets of several assets of the same kind,

sharing a subset of common failure modes. In this paper we study the latter. Usually, multi-unit systems are reduced to systems where all units are assumed to be fundamentally equal [17, 18]. However, this assumption is known in many cases to not be true and different maintenance policies may apply better to each individual asset. A novel approach to deal with multi unit systems is the Social Internet of Things [7], which provides a paradigm for objects to connect with each other and exploit asset individualities to improve global performance [19].

In the Social Internet of Things, the fleet of connected objects form a network. Research exists using the technologies enabling the Social Internet of Things to conceptualise a network of assets in order to perform diagnostics. For instance, Edzel Lapira developed clustering techniques in order to obtain individualised fault-detection in a network of similar machines [20]. Lapira’s approach, despite being adaptable to nonstationary regimes, does not consider a truly distributed architecture as most computations are done centrally. Other research exists where the concept of cluster-informed diagnostics is exploited in real-life experiments [21].

Despite these advances in the field of diagnostics, asset networks for prognostics have remained largely unexplored. Most existing multi-unit health prognostics methods are centralised and dependent on a rich set of historical data. Typically, the time series measurements of the Health Indicator are registered and saved in a database. Then, the trajectory of each asset is compared with previously recorded trajectories through a similarity metric, and the closest matches are used to obtain reliable prognostics [16, 22]. In such an approach it is common to use the  $k$  nearest trajectories and weight them in order to increase precision [23]. Other approaches consist of using machine learning to classify and predict failures, providing an estimate of the useful time to failure [24].

Multi-unit systems have also been studied in the field of Distributed Decision Making, which deals with problems involving multiple decisions aimed to optimise an objective function [25]. However, Distributed Decision Making is only scarcely incorporated into maintenance, usually coupled to human decisions [26], and artificial immune systems [27]. A largely unexplored field of Distributed Decision Making is collaborative health management, aimed to study how assets could collaborate with each other in order to improve maintenance and operational policies. This paper aims to help bridging this gap, providing the first theoretical treatment of the mathematical implications of distributed collaborative prognostics.

Multi-unit centralised systems depending on large amounts of data are likely to encounter computational limitations, and can only be implemented once a comprehensive data set is available. So far, existing studies have largely overlooked this issue by utilising increasingly powerful centralised computers, and providing prognostic results from already established failure data libraries. Therefore, no theoretical computation of the cost drawbacks and benefits of decentralising predictive maintenance decision exists in literature. In this paper we explore how through collaborative prognostics, the Social Internet of Things can provide a solution for these known drawbacks. Thus providing theoretical grounding to implement the proposed approach in a real industrial system.

In Section 2 we describe the theoretical foundation of the problem. We begin by describing the system in Section 2.1, and continue by discussing the collaborative prognosis algorithm in Section 2.2. A detailed description of the maintenance policy follows in Section 2.3, followed with a discussion of the importance of uncertainty and error in the system (Section 2.4). In Section 3 we describe a simulation of the proposed approach for the case of a fleet of assets corresponding to three different families of assets, the results of the simulation are presented in Section 3.1. We show how the cost per time depends on the number of collaborating assets, the weighting of the data, the noise of the system and the maintenance policy in a non-trivial way. We also propose a small modification of the approach for large time steps (Section 3.2). In Section 4 we discuss the findings resulting from the simulation and the limitations of our approach. The conclusion follows in Section 5. The paper concludes with a discussion of future work (Section 6), suggesting the computation of the case of multiple families of assets separated by different parametric distances.

## 2. Theoretical foundation

We propose a distributed collaborative method which considers asset similarities in order to improve the reliability of prognostics. In order to do so, we introduce the concept of Social Asset Networks: a Social Asset Network is a set of assets with local computational and sensing capabilities which can communicate with each other through an IoT platform. In the system studied in this paper, each asset has a digital replica, a software agent. From now on, when we refer to an *asset* performing an action, we actually mean the software *agent* associated with that given asset.

### 2.1. Description of the system

The system consists of a set of  $N$  assets connected to a network and able to communicate with each other. In practice, assets communicate through a Social Network Platform, which can be a single or several coordinated servers. Apart of communication, the Social Network Platform takes the role of system configuration, and allows for assets to join and leave the system at any time (see Fig. 1).

Inter-asset communications can be performed using any of the several existing IoT protocols [3, 28]. In the proposed system most of the data is treated locally by the assets and the data sent through communication is only a synthesised extraction of the sensor data. Therefore, communication costs are deemed negligible with respect of the cost optimisation presented in this paper. In practice, this assumption must be carefully evaluated by studying the connection policy of each particular Asset Network. In a distributed system, processing costs only weakly depend on the utilisation of the distributed processing power of the assets through their power consumption. For the system proposed in this section, all assets are assumed to incur in a constant processing cost, regardless of the number of neighbours they are linked to. Therefore, such costs are not weighted in the calculations either.

In the proposed system, assets are able to measure the real value of their Health Indicator, which corresponds to a synthetic indicator formed by a compilation of sensor data as in [16]. When this Health Indicator reaches 0, the asset fails. The assets are assumed to know the form of the equation governing their Health Indicator, but not the concrete values of its parameters. The aim of the assets is to determine these parameters by collaborating with each other, in order to obtain precise prognosis and implement maintenance strategies based on such prognosis (see Fig. 1). Several methods can be used to obtain the Health Indicator value from the multi-sensor time-series data, some of the most common being logistic regression [29], and multi-linear regression. The Health Indicator model used in this paper is characteristic of datasets conducive to multi-linear regression, as the logistic curve has asymptotes at 0 and 1 (we prefer a Health Indicator that crosses the abscissa axis in order to define a clear failure boundary).

The objective of any asset-tailored prognostics approach is to determine with minimal error and uncertainty the time of failure of the asset. Once this is known, the maintenance policy can be optimised [30]. Each asset in the

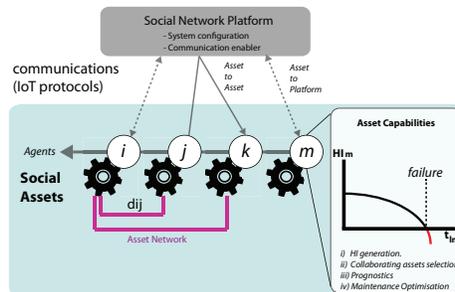


Figure 1: Diagram of the proposed system, featuring the Social Assets, its embedded agents, the Social Network Platform, the different kinds of connections between the Assets and the platform, and the Assets' capabilities.

network is identified by a unique discrete number between 1 and N. Although all the assets are connected to the network, at any given time each asset creates a list of  $N_i$  collaborating assets. These collaborations are represented using directed links (see Fig. 2). Such links are identified by binary sets of asset identifiers  $\{i, j\}$ ,  $i$  is used to identify the asset on the receiving end of the link, and  $j$  to the asset on the sending end of the link.

The Health Indicator of each asset  $i$ ,  $HI_i$  is chosen to follow the following equation (as in [16]):

$$HI_i(t_{li}) = a_i \left( 1 - e^{-b_i(t_{fi} - t_{li})} + \epsilon_{0,\sigma} \right). \quad (1)$$

Where  $t_{li}$  is the local time of the asset: the time since the last repair or since the asset was installed.  $(a_i, b_i, t_{fi})$  are assumed to be the parameters that govern the health of a particular asset.  $a_i$  is an amplitude parameter, determining the expected value of the Health Indicator at  $t_{li} = 0$ .  $t_{fi}$  is the expected time of

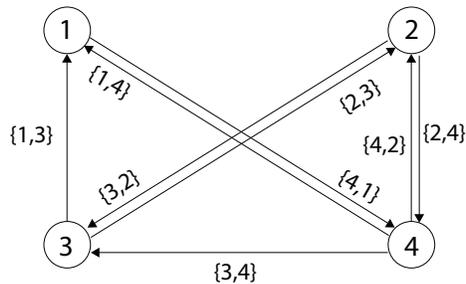


Figure 2: Diagram of an Asset Network, featuring four assets, each linked to  $N_i = 2$  collaborating assets. The directed links representing these collaborations are represented visually and through their binary identifiers  $i, j$ .

failure.  $b_i$  is a curvature parameter: in practice the smaller  $b_i$  is the quicker the asset is deteriorating. This has no influence on  $t_{fi}$  but it affects the probability of failure at smaller values of  $t_{fi}$ , which for smaller  $b_i$  increases significantly.  $\epsilon_{0,\sigma}$  is a Gaussian noise term with mean 0 and standard deviation  $\sigma$  which is included to represent realistic measurement limitations.

Assets can be in two different states: functioning and failure. An asset has failed if at any time its Health Indicator is smaller or equal to 0. For any other value of the Health indicator the asset is functioning. Therefore, at any time, the real state of the asset is assumed to be solely determined by the value of the Health Indicator.

At all times, the value of the Health Indicator is known by the manager and by the asset (supposedly through sensors installed within the asset). However, the values of  $(a_i, b_i, t_{fi})$  are unknown to the assets. Instead, the assets compute at each time step a triplet of estimated values  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e)$  using the collaborative prognosis algorithm described in Section 2.2. These values can be used to predict the future state of the asset and implement maintenance strategies, therefore determining the *prognostics* model.

## 2.2. Collaborative prognosis algorithm

The agent associated with each asset is programmed to guess the real values from its own Health Indicator data and from the information conveyed by its asset neighbours. The agents execute the following steps:

1. During the first  $n$  time steps since the first installation of the asset, each asset  $i$  is connected to  $N_i$  collaborating assets as follows: first, a triplet of initial estimated parameters  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e)$  is randomly initialised. Second, the asset's collaborating assets are selected as the  $j$  elements giving the  $N_i$  smallest distances  $d_{ij}$ , defined as in [31]:

$$d_{ij} = \frac{1}{\sqrt{3}} \sqrt{\frac{(\bar{a}_i^e - \bar{a}_j^e)^2}{\bar{a}_{\max}} + \frac{(\bar{b}_i^e - \bar{b}_j^e)^2}{\bar{b}_{\max}} + \frac{(\bar{t}_{fi}^e - \bar{t}_{fj}^e)^2}{\bar{t}_{f\max}}}. \quad (2)$$

Where  $(\bar{a}_{\max}, \bar{b}_{\max}, \bar{t}_{f\max})$  are the maximum values expected for their corresponding parameters (taken from the maxima of the randomly generated values of  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{f_i}^e)$ ). Therefore, each asset has a vector of collaborating assets determined by random discrete numbers  $\vec{N}_i = \vec{k}_i$  where  $\vec{k}_i$  is a discrete vector composed by the indexes  $j$  indicating the identity of each near neighbour. Note how during this first  $n$  time steps, there is no learning. This is done in order to gather enough initial data to fit Eq. (1) in later steps.  $n$  has to be bigger or equal than the degrees of freedom of Eq. (1). In our algorithm, assets calculate their similarity to all other assets in the fleet individually. However, platform-based cluster methods can also be used, and have been shown to be scalable to large asset fleets [21, 20]. As long as the model used to calculate inter-asset similarities remains the same and operates in real time, the results obtained in this work are applicable to both cases.

2. At time  $n$  or higher, assets guess the real values from the available Health indicator data points by fitting Eq. (1) without the noise term using a trust-region-reflective nonlinear least squares fitting algorithm, for example MATLAB's `lsqnonlin` function [32]. This returns a triplet of estimated values  $(a_i^e, b_i^e, t_{f_i}^e)$ .
3. Assets receive the triplets  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{f_i}^e)$  from the  $N_i$  nearest neighbours given by the vector  $\vec{N}_i$ , and re-evaluate their estimates using a weighted average from themselves and their neighbours:

$$\begin{aligned} \bar{a}_i^e &= a_i^e w_{ii} + \sum_{\vec{N}_i} w_{ij} \bar{a}_j^e, & \bar{b}_i^e &= b_i^e w_{ii} + \sum_{\vec{N}_i} w_{ij} \bar{b}_j^e, \\ \bar{t}_{f_i}^e &= t_{f_i}^e w_{ii} + \sum_{\vec{N}_i} w_{ij} \bar{t}_{f_j}^e. \end{aligned} \quad (3)$$

Where the weights are chosen to depend exponentially with the inverse of the distance:

$$w_{ij} = \frac{e^{-d_{ij}\gamma_j}}{\sum_{\vec{N}_i} e^{-d_{ij}\gamma_j}}. \quad (4)$$

This is set in order to enable a certain control of the influence of collaborating assets in the estimates of the asset itself. The parameter  $\gamma_j$  determines how strict the weighting is with respect to the estimated distance between assets. A system where all neighbours would be weighted equally corresponds to  $\gamma_j = 0$ . As  $\gamma_j$  increases, the information from collaborating assets is weighted with decreasing weights.

4. The assets send their guessed triplets to all the other assets in the system and calculate pairwise euclidean distances using Eq. (2).
5. Assets are linked to their  $N_i$  nearest neighbours, corresponding to the  $N_i$  smallest values of  $d_{ij}$  given by Eq. (2). These are the collaborating assets

- . Each asset stores the identifiers of his collaborating assets in a vector  $\vec{N}_i$  formed by the indexes  $j$  of all near neighbours.
- 6. At each time step the assets are maintained according to the time-based maintenance policy described in Section 2.3.
- 7. Steps 2-6 are repeated iteratively at each time step.

The Social Network Platform featured in Fig. 1 takes the role of system configuration, setting system parameters such as  $\gamma_i$ ,  $N_i$ , and allowing agents to join and leave the network. The platform also keeps track of the costs incurred upon implementation of the maintenance policy described in next section.

### 2.3. Maintenance policy

We implement a Predictive Maintenance policy. Assets are preventively repaired when the asset's local time (or time since last repair),  $t_{li}$ , is bigger or equal than the estimated time of failure multiplied by a factor  $\eta_i$ , set by the Social Network Platform.

$$\frac{t_{li}}{\bar{t}_{fi}^e} \geq \eta_i, \quad 0 \leq \eta_i \leq 1. \quad (5)$$

Alternately, assets are correctively repaired immediately if  $\text{HI}_i \leq 0$ . The cost of a preventive repair,  $C_p$  is much lower than the cost of a corrective repair,  $C_c$ . When an asset is repaired its age  $t_{li}$ , is set back to 0, and its Health Indicator is reset.

The cost per unit of time is optimised at every time step by assuming that the values of  $\bar{t}_{fi}^e$  correspond to the real values of  $t_{fi}$ . Under such conditions, the problem reduces to a time based replacement policy problem. Under stability of the parameters  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e)$ , optimising the system for one cycle would correspond to the long-term optimal solution [30]. However, in our system the optimisation is continuously updated as  $\bar{t}_{fi}^e$  changes over time as the assets revise their estimates by learning from new data. For each asset with a triplet  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e)$ , the total expected cost per unit time is given by:

$$C(\eta_i \bar{t}_{fi}^e) = \frac{C_p + C_c H(\eta_i \bar{t}_{fi}^e)}{\eta_i \bar{t}_{fi}^e}. \quad (6)$$

Where  $H(\eta_i \bar{t}_{fi}^e)$  is the expected number of failures in the interval  $(0, \eta_i \bar{t}_{fi}^e)$ . Differentiating Eq. (6) with respect to  $\eta_i \bar{t}_{fi}^e$  and equating it to 0, we obtain:

$$\eta_i \bar{t}_{fi}^e h(\eta_i \bar{t}_{fi}^e) - H(\eta_i \bar{t}_{fi}^e) = \frac{C_p}{C_c}. \quad (7)$$

Assuming that the time step is arbitrarily small, the expected number of failures per cycle corresponds to the probability of failures during one cycle. First we calculate the probability of failure at time  $t$ .

$$P(t) = \int_{-\infty}^0 N(\bar{\text{HI}}_i(t), \sigma, x) dx = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{-\bar{\text{HI}}_i(t)}{\sigma \sqrt{2}} \right) \right). \quad (8)$$

Here  $\bar{H}I_i$  is the Health Indicator without the random term, i.e. :

$$\bar{H}I_i(t_{li}) = \bar{a}_i^e \left( 1 - e^{-\bar{b}_i^e(\bar{t}_{fi}^e - t_{li})} \right) \quad (9)$$

The expected number of failures during one cycle corresponds to integrating  $P(t)$  over time and normalising to the time period.

$$\begin{aligned} H(\eta_i \bar{t}_{fi}^e) &= \frac{1}{\eta_i \bar{t}_{fi}^e} \int_0^{\eta_i \bar{t}_{fi}^e} P(t) dt = \\ &= \frac{1}{2\eta_i \bar{t}_{fi}^e} \left( \eta_i \bar{t}_{fi}^e + \int_0^{\eta_i \bar{t}_{fi}^e} \operatorname{erf} \left( \frac{-\bar{H}I_i(t)}{\sigma\sqrt{2}} \right) dt \right) = \\ &= \frac{1}{2\eta_i \bar{t}_{fi}^e} \left( \eta_i \bar{t}_{fi}^e + \int_0^{\eta_i \bar{t}_{fi}^e} \operatorname{erf} \left( \frac{-\bar{a}_i^e \left( 1 - e^{-\bar{b}_i^e(\bar{t}_{fi}^e - t)} \right)}{\sigma\sqrt{2}} \right) dt \right). \end{aligned} \quad (10)$$

Using the fundamental theorem of calculus, one can obtain  $h(\eta_i \bar{t}_{fi}^e)$ :

$$h(\eta_i \bar{t}_{fi}^e) = P(\eta_i \bar{t}_{fi}^e) = \frac{1}{2} \left( 1 + \operatorname{erf} \left( \frac{-\bar{H}I_i(\eta_i \bar{t}_{fi}^e)}{\sigma\sqrt{2}} \right) \right). \quad (11)$$

Eq. (7) then follows:

$$\begin{aligned} \eta_i \bar{t}_{fi}^e h(\eta_i \bar{t}_{fi}^e) - \\ \frac{1}{2\eta_i \bar{t}_{fi}^e} \left( \eta_i \bar{t}_{fi}^e + \int_0^{\eta_i \bar{t}_{fi}^e} \operatorname{erf} \left( \frac{-\bar{a}_i^e \left( 1 - e^{-\bar{b}_i^e(\bar{t}_{fi}^e - t)} \right)}{\sigma\sqrt{2}} \right) dt \right) \\ = \frac{C_p}{C_c}. \end{aligned} \quad (12)$$

Which can be solved for  $\eta_i$  numerically.

#### 2.4. On the trade-off between self-learning and collaborative learning

The problem of how to use information from other assets in order to predict the behaviour of a particular asset is not trivial. The fundamentals of this question can be traced back to the basic laws of measurement uncertainty. In effect, multiple assets can be viewed as multiple *similar* experiments with different degrees of variation. According to these laws, each asset will estimate its triplet of parameters  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e)$  with a level of uncertainty determined by its systematic and random components [33]:

$$(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{fi}^e) = (a_i^e, b_i^e, t_{fi}^e) \pm \sqrt{U_{si}^2 + U_{ri}^2}. \quad (13)$$

Each estimation will also carry some *Error*, given by the difference between the estimated value and the true value of the parameter triplet.

$$(Err_i^a, Err_i^b, Err_i^{t_f}) = (|\bar{a}_i^e - a_i|, |\bar{b}_i^e - b_i|, |\bar{t}_{fi}^e - t_{fi}|). \quad (14)$$

Typically, accumulating equivalent measurements governed by Gaussian uncertainty distributions, reduces the statistical uncertainty by a factor of  $1/\sqrt{N_S}$  where  $N_S$  is the number of samples. Therefore, we expect the statistical uncertainty associated to the estimates to reduce by a factor of  $1/\sqrt{N_i}$ . Uncertainty reduction notwithstanding, if the near neighbours of an asset  $i$  correspond to very different asset types, weighting their data in the estimators of such asset will not always be beneficial, leading to an increase of  $Err_i$  larger to the reduction of the associated uncertainty. This phenomena can be observed in collaborative learning for high values of  $d_{ij}$ , and small values of  $\gamma_i$ . In other words, more data does not always lead to smaller errors (better estimates). An example of this is seen in Fig. 6 where high  $\sigma$  values imply suboptimal choice of the collaborating assets. This drives the optimal number of collaborating assets to be  $N_i = 2$ , instead of  $N_i = 9$  as in the case of  $\sigma = 0$ .

### 3. Simulation model

In order to simulate the behaviour of the system, we use the agent simulation software NetLogo [34]. NetLogo is an agent-base programming language used to simulate complex systems and emergent phenomena. In our simulations we connected NetLogo with MATLAB using MatNet, an open source extension [35]. MATLAB is a commercial numerical computation software that we use to perform the non linear fit of the health indicator described in Section 2.2 .

In order to study the system we adopt the following assumptions:

- All assets have a common weight parameter  $\gamma_i = \gamma$ .
- The number of time steps run as step. 1 in Section 2.2 is set at  $n = 10$ .
- We simulate only three families of assets chosen represented by the triplets  $F : (a, b, t_f) \rightarrow \{(0.25, 0.025, 25), (0.5, 0.05, 50), (0.75, 0.075, 75)\}$ .
- Nine assets are simulated for each family, totalling 27 assets. This means that for  $N_i > 9$ , there will always be an asset forcibly connected to neighbours belonging to a different family.
- The cost of a preventive repair is set at  $C_p = 1$  and the cost of corrective repair at  $C_c = 100$ .
- Due to computational constraints, the maintenance policy of individual assets is not optimised in real time. Instead, all assets share the same  $\eta_i = \eta$ , which is varied in steps.

System convergence is defined by the average difference between the estimated time of failure  $t_{fi}^e$  and the real time of failure of the family the asset belongs to:

$$\langle |t_{fi}^e - t_{fi}| \rangle < \kappa. \quad (15)$$

Where  $\kappa$  is chosen to be the smallest time step of the system,  $\kappa = 1$ .

The simulation focuses in exploring the dependence of maintenance cost with the number of collaborating assets  $N_i$  and the noise in the system  $\sigma$ . Additionally, we aim to explore the dynamics of the system coupled with the replacement policy (determined largely by  $\eta$ ) and the weighting of the collaboration data (given by  $\gamma$ ).

### 3.1. Results

The system parameters  $\eta, N_i, \sigma, \gamma$  are varied in steps. We examine the cost per unit time  $K$ , with respect to the different variables of the system:

$$K = \frac{\sum_T C_T}{T}. \quad (16)$$

Where  $T$  is the total time of the simulation (the number of steps since initialisation). We see that the system follows a non-trivial behaviour with respect to all the studied variables, and results in different local minima for  $K$ . For a given  $\eta$ , a higher value of  $K$  implies a higher prediction error in the assets of the system, as the estimated time to failure  $t_{fi}^e$  is more likely to be higher than the true time to failure, leading to unwanted corrective repairs.  $K$  can then be seen in this paper as a proxy for model accuracy. For example, for the case of  $\sigma = 0.01$ ,  $\gamma = 0.1$  and  $\eta = 0.8$ , increasing the number of collaborating assets results in smaller costs up until  $N_i = 3$ . For  $N_i = 4$ , the tendency reverses. In this case, assets that are too different from each other are allowed to exchange information, leading to a sub-optimal solution (see Fig. 3 and Fig. 4).

Fig. 5 shows the dependence of  $K$  on  $\eta$  and  $N_i$ . The results suggest that  $N_i = 7$ ,  $\eta = 0.3$  is the best solution, given  $\sigma = 0.1$ .  $\gamma = 0.1$ . The dependence with respect to  $\eta$  follows the expected behaviour, with the solution being suboptimal for too large or too small  $\eta$ . For large  $\eta$  the preventive repairs are done too close to the point of failure, leading to unwanted corrective repairs. For small  $\eta$ , the frequency of preventive repairs is too high and the assets in the system do not accumulate sufficient health indicator data to learn from themselves.

Fig. 6 shows the dependence of  $K$  on  $N_i$  and  $\sigma$  for  $\eta = 0.5$  and  $\gamma = 0.1$ . As expected, higher  $\sigma$  imply higher costs, and the optimal number of collaborating assets  $N_i$  will depend on the noise present in the system. It is interesting to see that for no noise, the dependency of  $K$  with  $N_i$  is monotonic (higher  $N_i$  means a better solution). However, when noise is introduced to the system the ideal number of collaborating assets becomes non-trivial and the global minimum shifts to smaller values of  $N_i$ . In practice, this means that for a noisy or uncertain system collaborative learning has to be constrained in order to avoid an increase in prediction errors (see Section 2.4).

Fig. 7 shows the dependence of  $K$  on  $\gamma$  and  $\sigma$  for  $\eta = 0.5$  and  $N_i = 2$ . As expected, for the case of absence of noise ( $\sigma = 0$ ), the weight parameter is better set at 0 as all information coming from the rest of the assets will be relevant. However, as the noise in the system increases, higher values of  $\gamma$  are more optimal as to constrain the information coming from different assets becomes a sensible idea.

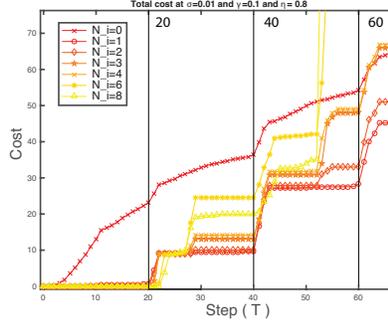


Figure 3: Total cost of the system for different values of  $N_i$  averaged over 50 trajectories. The cost is plotted for  $\sigma = 0.01$ ,  $\gamma = 0.1$  and  $\eta = 0.8$ . Note how the cost function shows clear increases at  $T = 25\eta = 20$ ,  $T = 50\eta = 40$  and  $T = 75\eta = 50$ , points that correspond to the preventive maintenance times upon convergence. Note how the increase at  $T = 25\eta$  is smaller than the increase at  $T = 50\eta$ , this is due to the fact that in the latter case the preventive maintenance time of the second cycle of assets with  $t_{fi} = 25$  coincides with the maintenance time of the assets with  $t_{fi} = 50$ .

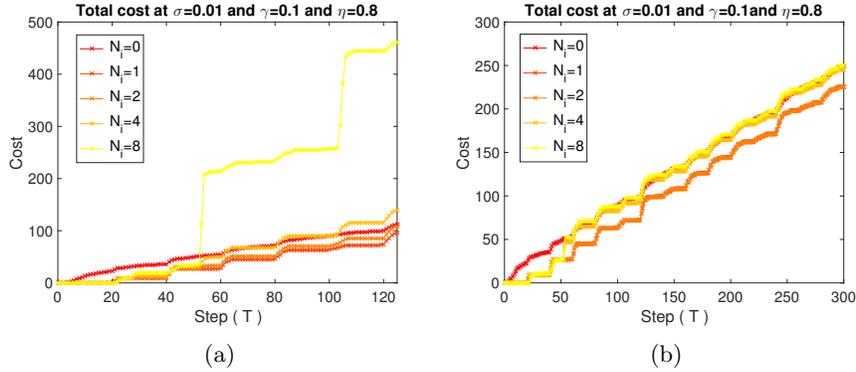


Figure 4: Total cost of the system using collaborative learning after convergence (a), compared to restricting to self learning upon convergence (b) for  $\epsilon = 1$ . The cost is plotted for different values of  $N_i$  averaged over 50 trajectories for  $\sigma = 0.01$ ,  $\gamma = 0.1$  and  $\eta = 0.8$ . Note how in (a) the cost functions for  $N_i > 0$  exceed the values for the self-learning approach ( $N_i = 0$ ) at long values of  $T$ . While in (b), even at long timesteps of the simulation, the cost of collaborative learning is always equal or smaller than the cost of self learning. With the case of  $N_i = 2$  providing minimal costs. The minimum cost solution corresponds to the one reaching fastest convergence, as it implies less corrective repairs in the early phase of the simulation.

### 3.2. System behaviour after convergence

We have shown that collaborative prognostics ( $N_i > 0$ ) reduces the maintenance cost upon convergence (satisfaction of Eq. (15)) for certain levels of system noise  $\sigma$ . However, when the system is run for a long period of time, for small values of  $\gamma$ , collaborative prognostics is costlier than self-learning ( $N_i = 0$ ) (see Fig. 4). This is due to the fact that for the proposed model, self-learning is able to estimate the real values of  $(a_i, b_i, t_{fi})$  with asymptotically

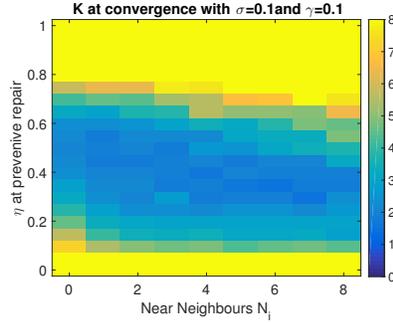


Figure 5: Cost per unit time,  $K$  with  $\sigma = 0.1$  and  $\gamma = 0.1$  for different values of  $N$  and  $\eta$  averaged over 20 trajectories. A global minimum is observed at  $N = 7$ ,  $\eta = 0.3$ .

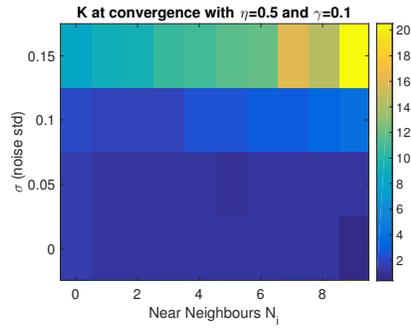


Figure 6: Cost per unit time,  $K$  with  $\gamma = 0.1$  and  $\eta = 0.5$  for different values of  $N$  and  $\sigma$  averaged over 20 trajectories. Note how for  $\sigma > 0.1$ , increasing the number of collaborating assets,  $N_i$ , is detrimental for the system cost.

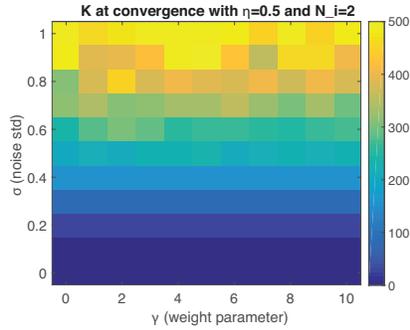


Figure 7: Cost per unit time,  $K$  with  $N = 2$  and  $\eta = 0.5$  for different values of  $\gamma$  and  $\sigma$  averaged over 20 trajectories. Note how several local minima are observed, showing that optimising the value of  $\gamma$  is non-trivial. If the cost is recorded at  $T=150$  instead of at convergence, the variation along  $\gamma$  vanishes, showing that the dynamics of the system upon convergence differ from the dynamics at large times.

small statistical uncertainty, given that the simulation is let run a long enough time. Adversely, collaborative learning always risks weighting information from neighbours from another asset family (see Section 2.4). This caveat is avoided by adding a condition that increases the value of  $\gamma$  upon convergence of the triplet  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{f_i}^e)$ .

In order to do so we add the following condition: if  $|\bar{t}_{f_i}^e(t_{li}) - \bar{t}_{f_i}^e(t_{li} + 1)| < \epsilon$ , during five consecutive time steps, then  $\gamma_i \rightarrow \infty$  (equivalent to  $N_i = 0$ ).  $\epsilon$  is chosen to be small (for example,  $\epsilon = \kappa$ ). In practice this means that once all the information has been extracted from the collaborating assets, the asset is allowed to learn only from itself. Once this condition is incorporated collaborative learning is less costly than self-learning at large values of  $T$ , even if very small values of  $\gamma$  are chosen (see Fig. 4).

### 3.3. Optimisation upon convergence

In this paper, the system has been simulated at global  $\eta = \eta_i$  values, without real-time optimisation of the maintenance policy. In order to show that the network dynamics do not influence the validity of Eq. (12), we simulate the system for a family of assets and plot its cost per second depending on  $\eta$  and  $\sigma$ . In Fig. 8 we show how the theoretical optimised values of  $\eta$  correspond to simulation.

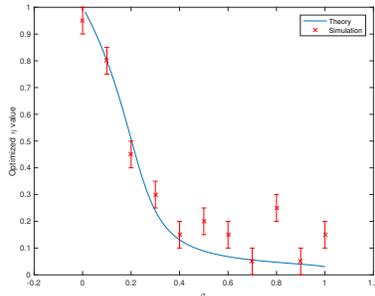


Figure 8: Optimal (minimum cost)  $\eta$  value obtained from simulating a family of 9 assets interchanging information upon convergence, compared to the theoretical  $\eta$  value predicted by Eq. (12). Note how higher values of  $\sigma$  imply higher variance in the system as the estimated values of  $(\bar{a}_i^e, \bar{b}_i^e, \bar{t}_{f_i}^e)$  fluctuate due to the noise.

## 4. Discussion

In this paper, we have introduced the idea of Social Asset Networks. The main difference between the approach we propose and alternative maintenance policies, is that ours does not require to work with a large library of previously recorded Health Indicator trajectories. Other popular multi-unit maintenance treatments rely on extensive datasets of the previous behaviour of the assets,

which are then treated in a centralised computer and used in order to predict future behaviour. This implies a long period where sub-optimal maintenance policies are applied, increasing the total cost.

As does any first attempt, our work has several limitations: the Health Indicator and noise distribution functions studied in this paper are not general. For example, Weibull distributions are often found to describe better real time failures. Additionally, dependencies with the expected asset computational limitations are not considered except in the design of the system. Thus, the presented results only apply to a subset of well-understood industrial asset types, and care must be taken when generalising them. However, the concept of Social Asset Networks can be generalised to other diagnostics and prognostics models by adapting the failure and maintenance rules and the definition of the distance metric.

The results shown in Fig. 3 to 8 clearly indicate that Social Asset Networks are multi-variable optimisation problems, with global and local minima following non-trivial dynamics. However, it is still possible to observe some general behaviours:

- **On the effect of noise:** as expected, an increase of the value of  $\sigma$  causes an increase of the total cost. At low values of  $\sigma$ , the optimal solution belongs to  $N_i > 0$ . However, at a certain value of  $\sigma$ , the optimal solution corresponds to assets only learning about themselves. This is explained because the number of collaborating assets not actually belonging to the same family induces an error  $Err_i$  that overcomes the reduction on uncertainty given by sharing the estimates. The system then never fulfils Eq. (15). By applying the fix proposed in Section 3.2, one can see that collaboration is more resilient to large values of  $\sigma$  as long as the definition of convergence considers the increase in noise.
- **On the effect of  $\gamma$ :**  $\gamma$  has a crucial effect on the long term values of  $K$  for the simulation. For small values of  $\gamma$  (close to 0.1), the collaborative approach is sensitive to higher noise levels due to incorporation of unsuitable data. Increasing the value of  $\gamma$  either globally or in real time upon convergence of each individual asset eliminates this effect.
- **On the effect of the number of collaborating assets,  $N_i$ :** in general, increasing the number of collaborating assets helps convergence and reduces cost, as long as  $\sigma$  is kept sufficiently low. When  $\sigma$  reaches a given level, a value of  $N_i$  which differs from the maximum value of  $N_i$  can be found to correspond to the minimum in the maintenance cost function  $K$  with no additional local minima. If  $\sigma$  is further increased, the configuration of minimum cost corresponds to self-learning ( $N_i = 0$ ). Care must be taken when  $N_i$  starts being on the order of the number of assets belonging to each asset family. If  $N_i$  overcomes that value, the cost starts increasing due to the increment of connections with assets belonging to different families. Fleet-wide learning (corresponding to  $N_i = 8$ ), is significantly

costlier than collaborative learning. This observation has practical relevance as in many industrial systems data abundance is often prioritised over specificity, leading to prediction errors.

- **On the effect of  $\eta$ :**  $\eta$  affects the cost as expected in a time based maintenance system: there is a single value of  $\eta$  corresponding to the global minimum in the cost function. In practice,  $\eta_i$  can be optimised in real time for each asset in the network. This optimisation will result in a less costly system than choosing an optimal general value of  $\eta$ .

The typical optimisation equations for single asset cases cannot be used to obtain the optimal  $\eta$  for an heterogeneous asset network. Instead, under convergence, each asset maintenance strategy must be optimised separately, obtaining individual values of  $\eta_i$ .

Apart from discussing the parametric dependencies, it is important to discuss the choice of the distance metric  $d_{ij}$ . Note how in our distance, all parameters receive the same weight when, in fact, the maintenance model is only determined by the time of failure  $\bar{t}_{fi}^e$  (see Section 2.3). In principle, one could define the distance differently, weighting solely or mostly the time of failure and not the parameters  $\bar{a}_i^e$  and  $\bar{b}_i^e$ . However, defining the metric solely based on  $\bar{t}_{fi}^e$  would not work in a system where the whole data series is exchanged between assets (as in most realistic machine learning algorithms). In such a system, if each asset were to perform a weighted fit of the shared data, the differences in  $\bar{a}_i^e, \bar{b}_i^e$  would propagate through the fit creating a big uncertainty in the estimated  $t_{fi}$ , resulting in a costlier maintenance system. We choose Eq. (2) because it can be generalised to a system where the assets exchange larger amounts of data with each other, apart from the triplet  $(a_i^e, b_i^e, t_{fi}^e)$ .

Previously published  $k$  near neighbour treatments focus on calculating the distance between measured Health Indicator trajectories [16], some assigning more weight to more recent measurements [36]. Focusing solely in recorded trajectories is reasonable, in the sense that similar failure modes are likely to be shared along the multi-unit system. Thus, the trajectory recorded for a certain asset may serve as a good predictor of the future behaviour of another particular asset. However, this does not always properly account for the differences between assets. Similar to our paper, measuring asset similarity just by the similarity between observations disregards the effect of the operating environment. This is acknowledged by [36], who define an ideal reference system to be “a system physically identical with the operating system, which worked in the same operating conditions”.

The balance between asset similarity and observed trajectory similarity is an important one, that could be compared to the difference between changing the experimental conditions and changing the experiment itself. Controlling for asset similarity will correspond to observing different trajectories of the same process influenced by the individual conditions of each asset. When observing sufficiently similar assets, trajectory differences will correspond to several different processes or failure modes that may be shared between the assets. In

practice, failure mode similarities shall be used to classify failure modes and asset similarities to weight different failure data in a predictive trend. This issue has been studied before in the field of cooperative agent information, in which Puuronen and Terziyan proposed similarity evaluation in three layers: problems, agents and solutions [37]. In a typical multi unit maintenance case, the problems would be the failure modes, the agents would correspond to the assets and the solutions would be the prognostic models generated by each asset for each failure mode.

The issue of uncertainty in Social Asset Networks is complex. The approach presented in this paper stands mostly on international standards for measurement uncertainty. Concretely we focus on the Guide to the Expression of Uncertainty in Measurement (GUM) rules of measurement uncertainty and propagation [33]. Other approaches, such as belief functions and empirical Bayes methods can also be used to investigate the effects of data sharing between dissimilar assets. For the intend of our work -to pinpoint the opportunities and problems offered by this approach- the GUM rules suffice.

## 5. Conclusion

We present a novel approach to optimise preventive maintenance policies in multi unit systems connected to an IoT platform: Social Asset Networks. The approach consists on weighting the similarities between different assets in order to connect assets to collaborating assets, and consecutively update asset predictions according to the neighbours data. Social Asset Networks are real time, distributed smart maintenance systems that do not depend on large datasets of historic data. Using simulations within NetLogo, a multi-agent software, we prove the validity of such approach in a simple but generalisable case based on known Health Indicator models with a noise term. We conclude:

- Collaborative learning outperforms self-learning, as long as collaborative learning is reduced once assets have reached convergence and as long as  $\sigma$  is kept low. In other words, collaborative learning helps reaching convergence to the real values faster, but once the system has converged self-learning can outperform collaborative learning for arbitrarily large time scales.
- Maintenance cost decreases with increased number of neighbours until a tipping point when it increases again.
- Upon convergence, network dynamics do not greatly influence the optimisation of the maintenance policy. A maintenance policy can be defined for each individual asset in a multi asset system.

Additionally, we show non-trivial dependencies between the cost function and the different variables of the system. Social Asset Networks enable an asset manager to optimise the maintenance of large fleets of assets equipped with sensing, communication and processing technology without the need of centralised servers.

## 6. Future Work

We have only simulated an ad hoc case with assets belonging to three different families. However, in reality, depending on the system the asset types may be several and their relative distances may be much closer or further away than the proposed case. Future work should focus on the change of the validity of the approach when the relative distances between families decrease and the system is much more sensitive to measurement noise.

The effect on maintenance cost of the presented approach has been computed assuming negligible communication costs. That is, the results only show the effect of Social Asset Networks in improving prognostics accuracy. In order to completely assess the industrial applicability of the proposed approach, communication costs must be weighted in the calculations too. For this, the connection policy must be rigorously and specifically outlined. This remains as one of the fundamental pieces of future work to be undertaken.

In this paper, we have considered that differences between assets fully determine differences between their Health Indicator models. Future work also should consider differences between failure modes and the different trajectories followed by these modes determined by the environment and the asset state.

In practice, distance metrics are unlikely to stem solely from parametric components. Indicators such as the production model of the asset, customer information, etcetera, are likely to be nominal attributes. In order to take into account such attributes, one could use metrics that combine parametric and nominal components, such as the Heterogeneous Value Difference Metric or the Interpolated Value Difference Metric [31].

The applicability of the presented approach should be demonstrated in an industrial test case. Ideal cases would be industries where the assets are expected to have a well-defined range of similarity. An example is the car industry, where car models vary across years and across variants of the same model, and where the deterioration of the cars depends greatly of environmental variables such as weather and driving culture. Another example is the gas turbine industry, where subsequent turbine models show very different maintenance histories depending on the workload, history of repairs and placement.

## 7. Acknowledgements

We acknowledge Bang Ming Yong and the VR team at the Institute For Manufacturing for providing the computer used to perform the simulations. This research was supported by SustainOwner (Sustainable Design and Management of Industrial Assets through Total Value and Cost of Ownership), a project sponsored by the EU Framework Programme Horizon 2020, MSCA-RISE-2014: Marie Skłodowska-Curie Research and Innovation Staff Exchange (Rise) (grant agreement number 645733 Sustain-owner H2020-MSCA-RISE-2014).

- [1] T. Dabney, L. Hernandez, P. A. Scandura, R. Vodicka, Enterprise health management framework - A holistic approach for technology planning,

- R&D collaboration and transition, 2008 International Conference on Prognostics and Health Management, PHM 2008.
- [2] G. Medina-Oliva, P. Weber, B. Iung, Industrial system knowledge formalization to aid decision making in maintenance strategies assessment, *Eng. Appl. Artif. Intel.* 37 (2015) 343–360.
  - [3] L. D. Xu, W. He, S. Li, Internet of things in industries: A survey, *IEEE Transactions on Industrial Informatics* 10 (4) (2014) 2233–2243.
  - [4] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, *Future. Gener. Comp. Sy.* 29 (7) (2013) 1645–1660.
  - [5] P. Tuwanut, S. Kraijak, A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends, 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015) (2015) 6.–6 .
  - [6] V. Gazis, M. Goertz, M. Huber, A. Leonardi, K. Mathioudakis, A. Wiesmaier, F. Zeiger, Short Paper : IoT : Challenges , Projects , Architectures (2015) 145–147.
  - [7] H. Li, A. K. Parlikad, Social Internet of Industrial Things for Industrial and Manufacturing Assets, *IFAC-PapersOnLine* 49 (28) (2016) 208–213.
  - [8] I. Mezei, V. Malbasa, I. Stojmenovic, Robot to robot, *IEEE Robotics and Automation Magazine* 17 (4) (2010) 63–69.
  - [9] S. M. Wood, D. L. Goodman, Prognostics in High Reliability Telecom Applications (2006) 1–3.
  - [10] D. Goyal, B. S. Pabla, Condition based maintenance of machine tools-A review, *CIRP Journal of Manufacturing Science and Technology* 10 (2015) 24–35.
  - [11] S. Alaswad, Y. Xiang, A review on condition-based maintenance optimization models for stochastically deteriorating system, *Reliability Engineering and System Safety* 157 (2017) 54–63.
  - [12] G. Zhao, Wireless sensor networks for industrial process monitoring and control: a survey, *Network Protocols and Algorithms* 3 (1) (2011) 46–63.
  - [13] A. Grall, L. Dieulle, C. Bérenguer, M. Roussignol, Continuous-time predictive-maintenance scheduling for a deteriorating system, *IEEE Transactions on Reliability* 51 (2) (2002) 141–150.
  - [14] J. Sun, H. Zuo, W. Wang, M. G. Pecht, Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance, *Mech. Syst. Signal. Pr.* 28 (2012) 585–596.

- [15] Fang Qian, Gang Niu, Remaining useful life prediction using ranking mutual information based monotonic health indicator, 2015 Prognostics and System Health Management Conference (PHM) (2015) 1–5.
- [16] T. Wang, J. Yu, D. Siegel, J. Lee, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, 2008 International Conference on Prognostics and Health Management, PHM 2008 (November).
- [17] Y. Zhou, M. Chioua, W. Ni, Data-driven multi-unit monitoring scheme with hierarchical fault detection and diagnosis, 2016 24th Mediterranean Conference on Control and Automation (MED).
- [18] P. K. Shtessel, Y. B., Blood glucose regulation using higher-order sliding modes, *Int. J. Robust. Nonlin.* 18 (October 2014) (2008) 557–569.
- [19] H. Ning, H. Liu, J. Ma, L. T. Yang, R. Huang, Cybermatics: Cyberphysicalsocialthinking hyperspace based science and technology, *Future. Gener. Comp. Sy.* 56 (2016) 504–522.
- [20] E. R. Lapira, Fault Detection In a Network of Similar Machines Using Clustering Approach, Ph.D. thesis, University of Cincinnati (2012).
- [21] F. Cannarile, M. Compare, F. Di Maio, E. Zio, Handling reliability big data: A similarity-based approach for clustering a large fleet of assets, *Safety and Reliability of Complex Engineered Systems* (2015) 891–896.
- [22] A. Bleakie, D. Djurdjanovic, Analytical approach to similarity-based prediction of manufacturing system performance, *Comput. Ind.* 64 (6) (2013) 625–633.
- [23] E. Ramasso, Joint prediction of observations and states in time-series : a partially supervised prognostics approach based on belief functions and KNN (2012) 1–13.
- [24] C. Yang, S. Letourneau, J. Liu, Q. Cheng, Y. Yang, Machine learning-based methods for TTF estimation with application to APU prognostics, *Appl. Intell.* 46 (1) (2017) 227–239.
- [25] C. Schneeweiss, Distributed decision making - A unified approach, *Eur. J. Oper. Res.* 150 (2) (2003) 237–252.
- [26] R. Yu, B. Iung, H. Panetto, A multi-agents based E-maintenance system with case-based reasoning decision support, *Eng. Appl. Artif. Intel.* 16 (4) (2003) 321–333.
- [27] N. Bayar, S. Darmoul, S. Hajri-Gabouj, H. Pierreval, Fault detection, diagnosis and recovery using Artificial Immune Systems: A review, *Eng. Appl. Artif. Intel.* 46 (2015) 43–57.

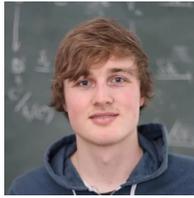
- [28] P. P. Ray, A survey on Internet of Things architectures, *Journal of King Saud University - Computer and Information Sciences*.
- [29] J. Yan, M. Koç, J. Lee, A prognostic algorithm for machine performance assessment and its application, *Production Planning & Control* 15 (8) (2004) 796–801.
- [30] A. K. S. Jardine, *Maintenance, Replacement and Reliability*, Vol. 1542, 2013.
- [31] D. R. Wilson, T. R. Martinez, Improved heterogeneous distance functions, *J. Artif. Intell. Res.* 6 (1997) 1–34.
- [32] MATLAB, version 9.0.0 (R2016a), The MathWorks Inc., Natick, Massachusetts, 2016.
- [33] Bipm, Iec, Ifcc, Ilac, Iso, Iupac, Iupap, Oiml, Evaluation of measurement data Supplement 1 to the Guide to the expression of uncertainty in measurement Propagation of distributions using a Monte Carlo method, Tech. rep. (2008).
- [34] U. Wilensky, NetLogo. <http://ccl.northwestern.edu/netlogo/>, Center for Connected Learning and ComputerBased Modeling Northwestern University Evanston IL 2009 (26.02.2009) (1999) Evanston, IL.
- [35] M. B. Biggs, J. A. Papin, Novel Multiscale Modeling Tool Applied to *Pseudomonas aeruginosa* Biofilm Formation, *PLoS ONE* 8 (10).
- [36] M.-Y. You, G. Meng, A generalized similarity measure for similarity-based residual life prediction, *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 225 (3) (2011) 151–160.
- [37] S. Puuronen, V. Terziyan, A similarity evaluation technique for cooperative problem solving with a group of agents, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1652 (1999) 163–174.



**Adrià Salvador Palau** received his B.Sc. degree in Physics from the Autonomous University of Barcelona, in 2014. He received his M.Sc. degree in 2015 from the University of Bergen (Norway), and is currently a Ph.D. Candidate in the Distributed Information and Automation Laboratory at the University of Cambridge, U.K. He has published several papers in Measurement Science, Physics and Stochastics . His research focuses on optimisation and modelling of systems, usually governed by some degree of randomness.



**Dr. Zhenglin Liang** received the Ph.D. degree in reliability engineering and asset management from the University of Cambridge, U.K., in 2016. Currently, he is a research associate at the Distributed Information and Automation Laboratory at the University of Cambridge. His research interests include stochastic process, predictive maintenance, maintenance optimisation and reliability of power equipment.



**Daniel Lütgehetmann** received his B.Sc. and M.Sc. degrees in Mathematics from Freie Universität Berlin in 2013 and 2014. Afterward, he completed his Ph.D. at Berlin Mathematical School working in the Algebraic Topology group of Prof. Holger Reich at Freie Universität Berlin. Currently, he is a postdoctoral research fellow at the University of Aberdeen. His research focuses on topology and geometry, more specifically topological aspects of configuration spaces of graphs.



**Dr. Ajith Kumar Parlikad** is a Senior Lecturer in Industrial Systems at the University of Cambridge. Ajith is the head of the Asset Management Group at the Institute for Manufacturing and is the theme leader on asset management and maintenance at the Centre for Smart Infrastructure and Construction (CSIC). His particular focus is examining how asset information can be used to improve asset performance through effective decision-making. Ajith has published over 100 papers in reputed academic journals and conferences. He is a member of The Institution of Engineering and Technology (IET) Technical Professional Network Committee on Asset Management. In addition, he is currently the chair of the IAM Academic and Research Network.