A METHODOLOGY FOR RAILWAY TRACK MAINTENANCE MODELLING USING PLAUSIBLE PETRI NETS

Manuel Chiachio[*a,b], **Juan Chiachio**[a], **Darren Prescott**[a], **John Andrews**[a]

[a]*Resilience Engineering Research Group, Faculty of Engineering, University of Nottingham, University Park, Nottingham, (UK)*
[b]*Dept. of Structural Mechanics & Hydraulics Engineering, University of Granada, Granada, (Spain)*

Abstract: This paper proposes a new mathematical methodology to model expert systems with the ability to sequentially learn from data. To this end, the Plausible Petri nets (PPNs) methodology, first developed in M. Chiachío et al. [Proceedings of the Future Technologies Conference, San Francisco, (2016), pp. 165-172] is used due to their ability to integrate continuous and discrete dynamics in a single net model, which allows us to analyse hybrid systems with interaction of diverse sources of information, like in expert systems. The efficiency of the proposed approach is demonstrated in an expert system model for railway track inspection management taken as case study using published data from a laboratory simulation of train loading on ballast, carried out at the Nottingham Railway Test Facility, University of Nottingham.

Keywords: Plausible Petri nets, Expert Systems, Bayesian Learning, Maintenance Modelling

1. INTRODUCTION

Developed countries are facing the onset of a new industrial revolution due to the rapid development of technologies including artificial intelligence (AI), sensing and robotics. Infrastructures and the built environment will certainly be part of this revolution due to their massive impact on the economy and society, and the low cost of these new technologies in relation to the cost of the infrastructure. As a result, the amount of real-time data and information coming from monitored infrastructures is expected to increase exponentially over the coming decades. This information has the potential to reduce by billions the national expenditure on infrastructure asset management [1]. Henceforth, there is a clear need to exploit the full potential of such infrastructure monitoring data by combining state-of the art AI approaches (like *expert systems*) with physics-based models for infrastructure operation and ageing, as a paradigm shift on what is typically known as Smart Infrastructure. An expert system is an engineered system that emulates human capacity to make decisions within a specific application domain using execution rules [2]. By learning from data, these rules can be enabled to dynamically accommodate environmental and contextual changes, therefore making the expert system more resilient to the new conditions.

This paper proposes an engineering application of an expert system capable to learn from data using *Plausible Petri nets* (PPNs). PPNs, firstly developed in [3, 4] by the authors, are a variant of the Petri nets (PNs) [5, 6] whereby discrete events (e.g., go/no-go decision, maintenance activities, resource availability, etc.) can be jointly modelled together with continuous processes whose evolution may be uncertain (e.g. deterioration process) under the same execution semantics. In PPNs, the uncertainty is modelled

*Corresponding author: manuel.chiachio-ruano1@nottingham.ac.uk, mchiachio@ugr.es

using *states of information* [7], which provide a mapping between the possible numerical values of a state variable with their relative plausibility. These states of information are used to integrate uncertain information about the system, like information from sensors, expert knowledge, etc., using probability density functions (PDFs). The methodology is illustrated using a case study about railway track geometry inspection. To this end, a specially suited PPN is proposed representing an expert system for railway track inspection, where several non-linearities like resource availability and operational restrictions are included as constraints to the system dynamics. The results reveal data learning and uncertainty management at a system level as key enabling aspects to allow critical infrastructures being operated more efficiently and autonomously.

The remainder of the paper is organised as follows. Section 2 briefly overviews basic concepts about PNs before introducing the PPNs and their learning from data in Section 3. Section 4 illustrates and discusses our approach in application to a case base self-adaptive expert system for railway track inspection. Finally, Section 5 gives concluding remarks.

2. BASIC CONCEPTS

2.1. Basis of Petri nets

PNs are bipartite directed graphs (digraph) which were introduced in the celebrated thesis dissertation *Kommunikation mit Automaten* by Carl Petri in 1962 [5] for modelling the dynamic behaviour of sequential asynchronous automatons. Two types of nodes are represented in a PN: *places* and *transitions*, where *arcs* are either from a place to transition or vice versa. A place represents a particular discrete state of the system or activity being modelled (e.g. considering health management modelling, places can be used to indicate the current state of a component or sub-system, or if any maintenance activity is currently in progress). Places are temporarily visited by *tokens*, the abstract moving units of PNs. The distribution of tokens over the PN at a specific time of execution is referred to as *marking*, which is expressed as a vector indicative of the state of the PN. The transitions are responsible of the dynamic behaviour of the PN, and enable the system to move from one state to another. For example, a component wear process is one of such processes which can be reflected using a transitions [10]. In practical applications of PNs, transitions are typically assigned with time delays which are useful for performance evaluation and scheduling problems of dynamical systems [6]. The resulting PNs are called *Timed Petri nets* if the delays are deterministic, and *Stochastic Petri nets* if the delays are randomly chosen by sampling distributions. In such cases, a transition is fired once its time delay has passed. In graphical representation, places are typically expressed using circles while transitions are drawn as bars or boxes. Arcs are labeled with their corresponding *weights*, non-negative integer values indicating the amount of parallel arcs (1 by default). Figure 1 illustrates a sample PN of three places (p_1, p_2, p_3), and one transition (t_1).

From a mathematical perspective, a PN is defined as a tuple $\mathfrak{N} = \langle \mathbf{P}, \mathbf{T}, \mathbf{E}, \mathbf{W}, \mathbf{M}_0 \rangle$, where $\mathbf{P} \in \mathbb{N}^{n_p}$ is an n_p-dimensional set of places, $\mathbf{T} \in \mathbb{N}^{n_t}$ is an n_t-dimensional set of transitions, $\mathbf{E} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ represents a set of directed arcs connecting places to transitions and vice versa, $\mathbf{W}$ is a set of non-negative numerical values (1 by default) acting as weights applied to each arc within $\mathbf{E}$, and $\mathbf{M}_0$ is a vector containing the initial distribution of tokens over the set of places, and is referred to as the initial marking. The following notation will also be considered:

${}^\bullet t$ is the set of input places of transition t, also referred to as the *pre-set* of t;

$t^\bullet$ is the set of output places of transition t, also referred to as the *post-set* of t;

At a certain $k \in \mathbb{N}$, the dynamics of a PN can be described through a state equation defined as follows

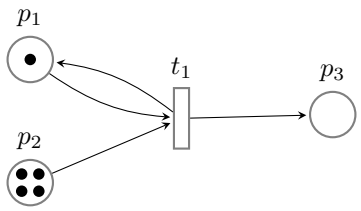


Figure 1: Example of Petri net of three places and one transition.

[6]:

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{A}^T \mathbf{u}_k \tag{1}$$

where $\mathbf{u}_k = (u_{1,k}, u_{2,k}, \ldots, u_{n_t,k})^T$ is the *firing vector*, whose elements are binary values taking 1 if transition t_i is fired, and 0 otherwise. $\mathbf{A}$ is an $n_t \times n_p$ matrix typically referred to as the *incidence matrix*, which can be obtained as the result of subtracting the *forward* ($\mathbf{A}^+$) and *backward* ($\mathbf{A}^-$) *incidence matrices* respectively, i.e.:

$$\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^- \tag{2}$$

where $\mathbf{A}^+ = \left[a_{ij}^+\right]$, $\mathbf{A}^- = \left[a_{ij}^-\right]$, $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$. The element a_{ij}^+ is the weight of the arc from transition $t_i \in \mathbf{T}$ to output place $p_j \in \mathbf{P}$, whereas a_{ij}^- is the weight of the arc to transition t_i from input place p_j.

In PNs, any transition t_i needs to be enabled as a condition to be fired, which occurs when each input place of t_i is marked with at least a_{ij}^- tokens. Mathematically:

$$M(j) \geqslant a_{ij}^- \;\; \forall p_j \in \;\; {}^\bullet t_i \tag{3}$$

where $M(j) \in \mathbb{N}$ is the marking for place p_j.

3. PLAUSIBLE PETRI NETS

Plausible Petri nets (PPNs) are a variant of PNs recently developed by the authors, which are based on a combination of discrete and continuous numerical processes whose values may be uncertain (*plausible*). Two interacting subnets form the PPN graph: 1) a *symbolic subnet*, where the tokens are objects in the sense of integer moving units, as in classical PNs [5], 2) a *numerical subnet*, where tokens are *states of information* [2], which are denoted using superscripts ($\mathcal{N}$) and ($\mathcal{S}$), respectively. In particular, the set of places $\mathbf{P}$ are partitioned into subset $\mathbf{P}^{(\mathcal{N})} \in \mathbb{N}^{n_p}$ and $\mathbf{P}^{(\mathcal{S})} \in \mathbb{N}^{n'_p}$, such that $\mathbf{P}^{(\mathcal{N})} \cup \mathbf{P}^{(\mathcal{S})} = \mathbf{P}$, and $\mathbf{P}^{(\mathcal{N})} \cap \mathbf{P}^{(\mathcal{S})} = \emptyset$. Superscripts n_p, n'_p represent the number of numerical and symbolic places, respectively. Analogously, transitions $\mathbf{T}$ are partitioned into numerical transitions $\mathbf{T}^{(\mathcal{N})} \in \mathbb{N}^{n_t}$ and symbolic transitions $\mathbf{T}^{(\mathcal{S})} \in \mathbb{N}^{n'_t}$, where $\mathbf{T}^{(\mathcal{N})} \cup \mathbf{T}^{(\mathcal{S})} = \mathbf{T}$, and $\mathbf{T}^{(\mathcal{N})} \cap \mathbf{T}^{(\mathcal{S})} \neq \emptyset$. In this case, n_t, n'_t denote the number of numerical and symbolic transitions, respectively. Observe that those transitions that belong to $\mathbf{T}^{(\mathcal{N})} \cap \mathbf{T}^{(\mathcal{S})}$ are referred to as *mixed transitions* [3, 4].

In PPNs, the referred states of information about a system state variable $\mathbf{x}_k \in \mathcal{X}$ are denoted by the PDFs $f^p(\mathbf{x}_k)$ and $f^t(\mathbf{x}_k)$ for numerical places and transitions, respectively. Thus, the marking $\mathbf{M}_k$ of a PPN at a certain time k consists in a combined vector $\mathbf{M}_k = \left(\mathbf{M}_k^{(\mathcal{N})}, \mathbf{M}_k^{(\mathcal{S})}\right)$, where $\mathbf{M}_k^{(\mathcal{N})}$ and $\mathbf{M}_k^{(\mathcal{S})}$ are column vectors of normalised PDFs and integer values, respectively. Moreover, in PPNs there exist arc weights for the symbolic places as in classical, denoted by $a_{ij}^{'+}, a_{ij}^{'-} \in \mathbf{W}^{(\mathcal{S})} \subset \mathbb{N}$, whereby the incidence

[2] A state of information can be described by a set of numerical values about a state variable, along with a mapping over them that assigns each numerical value with its *relative plausibility* [8, 7].

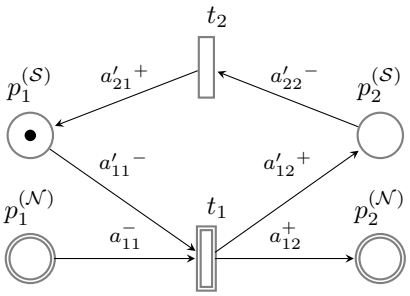


Figure 2: Illustration of a sample PPN with two numerical places ($p_1^{(\mathcal{N})}$, $p_2^{(\mathcal{N})}$), two symbolic places ($p_1^{(\mathcal{S})}$,$p_2^{(\mathcal{S})}$), and two transitions (t_1,t_2), of which one is a mixed transition.

matrix $\mathbf{A}^{(\mathcal{S})}$ can be obtained by Eq. (2). The arc weights for the numerical places are denoted by $a_{ij}^{+}, a_{ij}^{-} \in \mathbf{W}^{(\mathcal{N})} \subset \mathbb{R}^{+}$, such that $\mathbf{A}^{(\mathcal{N})} = \left[a_{ij}^{+}\right] - \left[a_{ij}^{-}\right]$, and $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$, where n_t, n_p represent the amount of numerical transitions and numerical places of the PPN, respectively. Note that the arc weights from the symbolic subnet, e.g. $\left(a'_{11}{}^{-}\right)$ are differentiated from the numerical ones using an accent ($'$). A PPN model is shown in Figure 2 for illustration purposes. Note that in the graphical representation, numerical nodes are drawn using double lines, with single lines used for the others. The dashed rectangles shown in Figure 2 highlight the pre-set and post-set of t_1.

3.1. Execution rules

In PPNs, the marking evolution of the symbolic subnet corresponds to the state equation of a PN [6] (recall Eq. [1]). However, the evolution of $\mathbf{M}_k^{(\mathcal{N})}$ relies on an *ad hoc* information flow dynamics based on two basic operations referred to as [3]: the *conjunction* and *disjunction* of states of information [8, 7]. In these operations, the logic operators AND ($\wedge$) and OR ($\vee$) are invoked to allow the continuos information from the numerical subnet to be exchanged into the PPN.

From this standpoint, the dynamics of PPNs is formulated under the adoption of the following rules [4]:

1. An input arc from place $p_j^{(\mathcal{N})}$ to transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ conveys a state of information given by $a_{ij}^{-}(f^{p_j} \wedge f^{t_i})(\mathbf{x}_k)$, which remains in $p_j^{(\mathcal{N})}$ after transition t_i has fired;

2. Transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ produces to an output arc a state of information given by $a_{ij}^{+}(f^{\bullet t_i} \wedge f^{t_i})(\mathbf{x}_k)$, where $f^{\bullet t_i}(\mathbf{x}_k)$ denotes the resulting density from the disjunction of the states of information of the pre-set of t_i. The normalised version of $f^{\bullet t_i}(\mathbf{x}_k)$ can be obtained as:

$$f^{\bullet \mathbf{P}_{t_i}}(\mathbf{x}_k) = \frac{1}{\beta}(f^{p_1} + f^{p_2} + \cdots + f^{p_m})(\mathbf{x}_k) \tag{4}$$

   where β is a constant, and $p_1, \ldots, p_m \in {}^{\bullet}t_i \subset \mathbf{P}^{(\mathcal{N})}$;

3. After firing numerical transition t_i, the state of information resulting in place $p_j^{(\mathcal{N})}$ from the post-set of t_i, is the disjunction of the state of information $f^{p_j}(\mathbf{x}_k)$ (the previous state of information), and $a_{ij}^{+}(f^{t_i} \wedge f^{\bullet t_i})(\mathbf{x}_k)$ (the information produced after firing transition t_i). Mathematically:

$$f^{p_j}(\mathbf{x}_{k+1}) = \left(f^{p_j} \vee a_{ij}^{+}(f^{t_i} \wedge f^{\bullet t_i})\right)(\mathbf{x}_k) \tag{5}$$

Note that execution rules given above for PPNs are, in general, difficult to evaluate analytically since the conjunction of states of information requires the evaluation of normalising constants involving an intractable integral. Particle methods [11] can be used in these cases to circumvent the evaluation of the normalising constant with a feasible computational cost. In particle methods, a set of N samples $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ with associated weights $\{\omega^{(n)}\}_{n=1}^{N}$ are used to obtain an approximation for the required density function [e.g. $(f_a \wedge f_b)(\mathbf{x})$], as follows:

$$(f_a \wedge f_b)(\mathbf{x}) \approx \sum_{n=1}^{N} \omega^{(n)} \delta(\mathbf{x} - \mathbf{x}^{(n)}) \tag{6}$$

where δ is the Dirac delta and $\mathbf{x}^{(n)} \sim (f_a \wedge f_b)(\mathbf{x})$. The particle weight $\omega^{(n)}$ represents the likelihood value of $\mathbf{x}^{(n)}$, and is representative of the plausibility of $\mathbf{x}^{(n)}$ when it is distributed according to $(f_a \wedge f_b)(\mathbf{x})$. It can be evaluated for the case of $\mathcal{X}$ being a linear space as follows:

$$\omega^{(n)} = \frac{f_a(\mathbf{x}^{(n)}) f_b(\mathbf{x}^{(n)})}{\sum_{n=1}^{N} f_a(\mathbf{x}^{(n)}) f_b(\mathbf{x}^{(n)})} \tag{7}$$

3.1.1. Transition firing in PPNs

In PPNs, any transition $t_i \in \mathbf{T}$ is fired at time k if the delay time has passed and:

1. Every symbolic place from the pre-set of t_i has enough tokens according to their input arc weight, as in classical PNs (recall Eq. (3));
2. Each of the conjunction of states of information between f^{t_i} and $f_k^{p_j}$ is possible, where $p_j^{(\mathcal{N})}$ belongs to the pre-set of t_i;
3. Conditions (a) and (b) are both satisfied when t_i is a mixed transition, i.e. $t_i \in (\mathbf{T}^{(\mathcal{S})} \cap \mathbf{T}^{(\mathcal{N})})$.

Note from Condition (b) that a conjunction, e.g. $(f_k^{p_j} \wedge f_k^{t_i})(\mathbf{x}_k)$, is possible if $(f_k^{p_j} \wedge f_k^{t_i})(\mathbf{x}_k) \neq \emptyset$ [8]. Note also that when any of the states of information involved in a conjunction is the *homogenous density* (also referred to as "non-informative density") $\mu(\mathbf{x}_k)$ of the state space of consideration $\mathcal{X}$, then the conjunction is always possible [8, 4], thus Condition (b) is automatically fulfilled. This argument is important in terms of using PPNs in practical examples, as will be demonstrated in next section.

As a matter of fact, PPNs have the property that an input arc from place $p_j^{(\mathcal{N})}$ to transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ carries a state of information given by the posterior density function of state variable $\mathbf{x}_k$ by just assuming that $f^{t_i}(\mathbf{x}_k)$ acts as likelihood function for a set of data $\mathbf{y}_k \in \mathcal{D}$ (which can be denoted as $p(\mathbf{y}_k|\mathbf{x}_k)$) and that the state-space $\mathcal{X}$ is a linear space [8]. This interesting property will be further exploited within the context of an engineering case study.

4. CASE STUDY

The PPN methodology explained above is exemplified here using data about permanent axial strain in a ballasted railway track taken from the the literature [12]. The interest of this engineering application resides in the need for AI methodologies which allow automated and adaptive decisions about maintenance activities and inspection actions in railway networks based on monitoring data [13]. In this case study, a PPN is developed to act as an expert system for railway track management, incorporating a physics-based model of track geometry degradation, data-learning, along with a number of operational rules, which provide the basis for triggering a number of control operations and inspection activities.

The data in this case study consists of a set of non-regularly scheduled (noisy) measurements $Y = (y_1, y_2, \ldots, y_k)$ of track settlement taken from Aursudkij *et al.* [12], which are sequentially introduced to the system at a set of discrete loading cycles. The test, as reported in [12], was conducted on the Railway Test Facility of the University of Nottingham [14], and simulates an axle load of approximately 20 tonnes over a ballasted track section comprised of 0.9 [m] (depth) subgrade material and 0.3 [m] (depth) ballast material. The dataset is reproduced in Table 1.

4.1. PPN model

The PPN-based expert system for railway track inspection considered in this case study is depicted in Figure 3. The system represents a number of rules which autonomously raise an alarm (e.g. "Line closure") or trigger inspection activities of a particular railway track section subjected to traffic loading degradation. Observe that the PPN is comprised of one numerical place ($p_1^{(\mathcal{N})}$), seven symbolic places ($p_1^{(\mathcal{S})}$ to $p_7^{(\mathcal{S})}$), three mixed transitions (t_1 to t_3), and four symbolic transitions (t_4 to t_7). The stochastic model for track degradation is embedded within the numerical place $p_1^{(\mathcal{N})}$ (details of the railway track degradation model is omitted here for clarity and lack of space. The reader is referred to [15] for further description, and to [16] by the authors to obtain the implementation details). The measurements $y_1, y_2, \ldots, y_k$ are assumed to come with a 5% white-noise type error, hence $y_k \sim \mathcal{N}(x_k, \sigma_{w_k})$, where $\sigma_{w_k} = 0.05\|y_k\|$. This PDF represents the state of information within transition t_1, which is given by:

$$f^{t_1}(x_k) = p(y_k|x_k) = (2\pi\sigma_{w_k}^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\frac{y_k - x_k}{\sigma_{w_k}}\right)^2\right) \tag{8}$$

Note that each time a new measurement arrives, transition t_1 is enabled, which by the PPN execution rules explained in § 3.1, leads to the conjunction of the states of information of $p_1^{(\mathcal{N})}$ and t_1. Note that, this conjunction leads to the posterior PDF and therefore to the update of the degradation variable x_k, except for a normalising constant.

An overview of the complete set of transitions is provided in Table 2. Observe from this table that the mixed transitions t_2 and t_3 are defined based on condition [4], henceforth, their activation is prescribed for the state variable x_k on fulfilling the condition $x_k \in C_i$, where subspaces C_i, $i = 2, 3$ are specified in the third column of Table 2. These transitions are driven by states of information that are expressed by Dirac Delta density functions [8], i.e., $f^{t_i} = \mathbb{I}_{C_i}(x_k), i = 2, 3$. In Table 2, function $H : \mathcal{X} \to \mathbb{R}$ denotes the differential entropy (DE)[3] of the degradation variable x_k , and $\mathbb{E}_{f^{p_1}}[x_k]$ denotes the expectation of x_k with respect to f^{p_1}. From a computational point of view, note that the conditions in the mixed transitions t_2 and t_3 specify numerical rules used by the expert system to raise an alarm or trigger an inspection action.

4.2. Results

By evaluating the proposed PPN-based system, changes in the numerical and discrete track states are obtained with reference to a number of automated actions which are activated through firing transitions

[3]The differential entropy of a stochastic variable x_k, is a measure of the uncertainty about the values taken by x_k, which is given by $^1/_2 \ln[(2\pi e) var(x_k)]$

Table 1: Experimental sequence of permanent unitary settlement (strain) data used for calculations, taken from [12] and obtained using the University of Nottingham Railway Test Facility (RTF).

Loading cycles k, ($\times\mathbf{10^3}$)	0	0.625	1.25	2.5	5	10	20	30	50	75
Unitary settlement [dimensionless]	0	0.0017	0.0045	0.0058	0.0075	0.0087	0.0104	0.011	0.012	0.01275

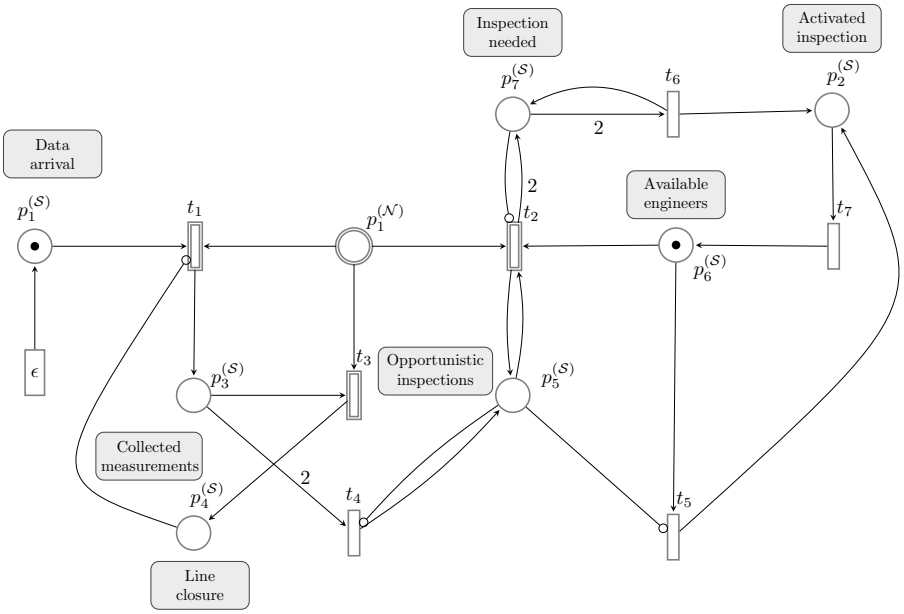
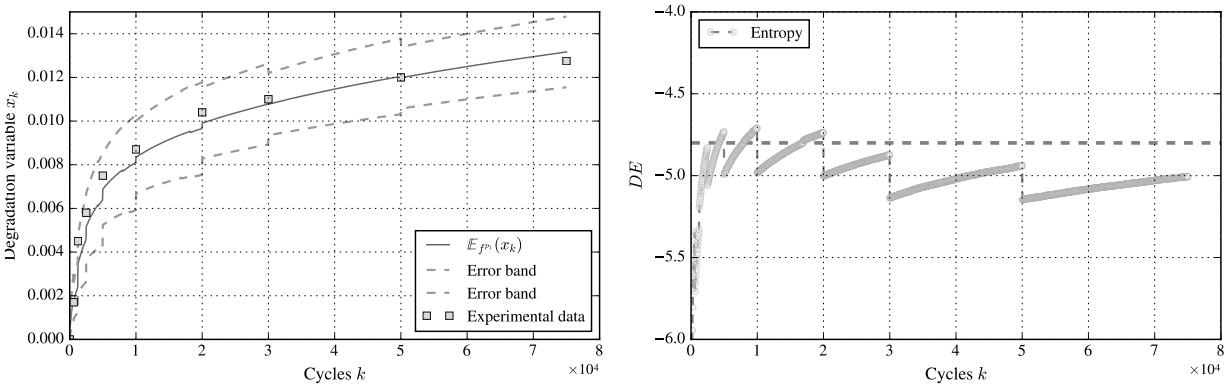


Figure 3: PPN of the case study presented in § 4. $p_1^{(\mathcal{N})}$ is where on-line predictions of track state take place. The predictions are updated as new data are collected. Note that a number of *inhibitor arcs* (those ending with a small circle) are used to prevent a transition from firing once its pre-set places are marked. A *cold transition* (ϵ) is used to represent the data arrival, which are assumed to be available at a set of non-regularly scheduled time instants, as shown in Table 1.

Table 2: Description of the transitions shown in Figure 3. In the third column (rules), the delays are expressed in cycles. The last column provides a description of the action taken by the PPN-expert system when the rules are met. PI: Periodic inspections, OI: opportunistic inspections, LC: Line Closure.

ID	Type	Rule	State of information	Action
t_1	Mixed	–	$f^{t_1} \sim p(y_k\|x_k)$ (Likelihood)	Update predictions
t_2	Mixed	$H(x_k) \geqslant -4.8$	$f^{t_2} \sim \mathbb{I}_{C_2}(x_k)$	Activates OI
t_3	Mixed	$\mathbb{E}_{f^{p_1}}[x_k] \geqslant 0.014$ [m]	$f^{t_3} \sim \mathbb{I}_{C_3}(x_k)$	Switches to LC
t_4	Symbolic	$\tau_4 = 0$ (delay)	–	Switches to OI
t_5	Symbolic	$\tau_5 = 0$ (delay)	–	Switches to PI
t_6	Symbolic	$\tau_6 \sim \mathcal{N}(1,1)$ (delay)	–	Activates OI
t_7	Symbolic	$\tau_7 \sim \mathcal{N}(24,1)$ (delay)	–	Concludes inspections

t_1 to t_7. The execution rules given in § 3.1 are applied to obtain the overall system evolution described through the marking $\mathbf{M}_k$, $k > 0$. In particular, the results for the estimated degradation variable in place $p_1^{(\mathcal{N})}$ along with its $5\% - 95\%$ probability bands, are depicted in Figure 4 for $k = 0 \rightarrow 75 \times 10^3$ cycles (see the leftmost panel). Panel 4b illustrates the temporal evolution of the uncertainty in the estimation of x_k within place $p_1^{(\mathcal{N})}$, with indication of the reference level when inspections are needed. This uncertainty is expressed and quantified through the DE. The observed drops in the sequence of DE values in Figure 4b correspond to the uncertainty reduction due to Bayesian learning when new measurements become available. Observe from these results that there is a period required by the PPN model to learn from the data, which corresponds to the loading cycles in the interval $(0, 5 \cdot 10^3]$. After this learning period, not only does the precision of the prediction of x_k clearly improve with time (predicted values of x_k closer to data y_k), but also the uncertainty of the prediction gradually tends to diminish, which is numerical evidence of the Bayesian learning taking place in $p_1^{(\mathcal{N})}$. Figure 5 provides a plot of the history of the tokens visiting place $p_2^{(\mathcal{S})}$ during the overall period of evaluation $k = 0 \rightarrow 75 \times 10^3$, and indicates the sequence of activated inspections within that period. Note that at the beginning of the process (specifically the first 2,500 cycles), inspections are activated even when the uncertainty (DE) about x_k



(a) (b)

Figure 4: Left: Plot of mean values and probability bands of $f_k^{p_1}$ for $k = 0 \rightarrow 75,000$. Right: History plot of the differential entropy of $f_k^{p_1}$. The dashed-horizontal line represents the threshold value (-4.8) given to activate transition t_2.

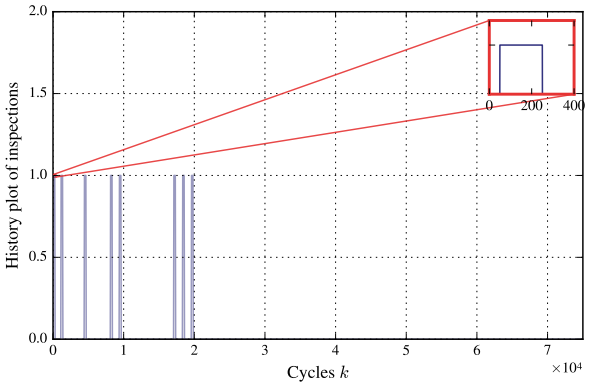


Figure 5: Plot of visiting tokens in place $p_2^{(\mathcal{S})}$ as a response of the PPN from Figure 3 using on-line data from the dataset (Y) shown in Table 1.

in $p_1^{(\mathcal{N})}$ is below the threshold value. According to the PPN graph in Figure 3, these correspond to PIs which must be carried out until at least two measurements are available, whereupon $p_6^{(\mathcal{S})}$ is marked and the system switches from PI to OI mode, as explained above. Note also from Figure 5 that a number of tokens visit place $p_2^{(\mathcal{S})}$ from cycle $k = 2.5 \times 10^3$ to about $k = 2 \times 10^4$, corresponding to inspection activities triggered because the uncertainty of the degradation variable x_k in this initial period passes the threshold several times, i.e. (DE $\geqslant -4.8$), activating t_2. After this initial period, the system identifies that no more inspections are needed. Observe that these results reveal that the PPN autonomously responds to the arrival of data through adaptation so that the sequence of discrete states (like inspection activities) are altered in response to the most up-to-date information from data Y. The results also show that the Bayesian learning of the PPN serves to control the uncertainty of the track settlement predictions so as to avoid triggering unnecessary inspections.

5. CONCLUSIONS

This paper presented a new methodology to model PPN-based expert systems capable to adapt their behaviour as long as new data arrive. An engineering case study has been presented, which uses experimental data of railway track degradation to demonstrate how monitoring data and model-based knowledge about track degradation can be integrated within a PPN-based expert system modelled using a PPN. The

results revealed that expert systems modelled using PPNs can respond to the arrival of data through adaptation so that the system can automatically respond to the most up-to-date information from data. Besides, in application to the railway track management problem, the results demonstrated the potential of the proposed methodology to shifts the burden of managing a track section from asset managers to an autonomous system that acts under the guidance of monitoring data (when available), and maintenance policies (implemented as rules), which are considered as inputs to our system.

Acknowledgements

Manuel Chiachío is Research Fellow of the Lloyd's Register Foundation (LRF), a charitable foundation, helping to protect life and property by supporting engineering-related education, public engagement and the application of research (www.lrfoundation.org.uk). Juan Chiachío is funded by the Engineering and Physical Sciences Research Council (UK) under grant EP/M023028/1, which has enabled the research reported in this paper. John Andrews is the LRF Director of the Resilience Engineering Research Group and also the Network Rail Professor of Infrastructure Asset Management at the University of Nottingham. The authors gratefully acknowledge the support of these organisations.

References

[1] R. Morimoto, "A socio-economic analysis of smart infrastructure sensor technology," *Transportation Research Part C: Emerging Technologies*, vol. 31, pp. 18 – 29, 2013.

[2] W. P. Wagner, "Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies," *Expert Systems with Applications*, vol. 76, pp. 85–96, 2017.

[3] M. Chiachío, J. Chiachío, D. Prescott, and J. Andrews, "An information theoretic approach for knowledge representation using Petri nets," in *Proceedings of the Future Technologies Conference 2016, San Francisco, 6-7 Dec.*, pp. 165–172, IEEE, 2016.

[4] M. Chiachío, J. Chiachío, D. Prescott, and J. Andrews, "A new paradigm for uncertain knowledge representation by Plausible Petri nets," *Information Sciences*, pp. on–line, 2018.

[5] C. A. Petri, *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik an der Universität Bonn, 1962.

[6] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[7] G. Rus, J. Chiachío, and M. Chiachío, "Logical inference for inverse problems," *Inverse Problems in Science and Engineering*, vol. 24, no. 3, pp. 448–464, 2016.

[8] A. Tarantola and B. Valette, "Inverse problems = quest for information," *Journal of Geophysics*, vol. 50, no. 3, pp. 159–170, 1982.

[9] J. Beck, "Bayesian system identification based on probability logic," *Structural Control and Health Monitoring*, vol. 17, no. 7, pp. 825–847, 2010.

[10] J. Andrews, D. Prescott, and F. De Rozières, "A stochastic model for railway track asset management," *Reliability Engineering & System Safety*, vol. 130, pp. 76–84, 2014.

[11] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo methods in practice* (A. Doucet, N. De Freitas, and N. Gordon, eds.), pp. 3–14, Springer, 2001.

[12] B. Aursudkij, G. McDowell, and A. Collop, "Cyclic loading of railway ballast under triaxial conditions and in a railway test facility," *Granular Matter*, vol. 11, no. 6, p. 391, 2009.

[13] P. Weston, C. Roberts, G. Yeo, and E. Stewart, "Perspectives on railway track geometry condition monitoring from in-service railway vehicles," *Vehicle System Dynamics*, vol. 53, no. 7, pp. 1063–1091, 2015.

[14] S. Brown, B. Brodrick, N. Thom, and G. McDowell, "The nottingham railway test facility, UK," in *Proceedings of the Institution of Civil Engineers-Transport*, vol. 160, pp. 59– 65, Thomas Telford Ltd, 2007.

[15] B. Indraratna, P. K. Thakur, J. S. Vinod, and W. Salim, "Semiempirical cyclic densification model for ballast incorporating particle breakage," *International Journal of Geomechanics*, vol. 12, no. 3, pp. 260–271, 2012.

[16] J. Chiachío, M. Chiachío, D. Prescott, and J. Andrews, "A reliability-based prognostics framework for railway track management," in *International Conference on Prognostics and Health Management (PHM, 2017)*, pp. 396–406, PHM, 2017.