Modelling adaptive systems using plausible Petri nets

J. Chiachío,[1*] M. Chiachío,[1,2] D. Prescott,[1] and J. Andrews[1]

[1] *Resilience Engineering Research Group, University of Nottingham, UK*
[2] *Department of Structural Mechanics and Hydraulic Engineering, University of Granada, Spain*
[*]Corresponding author: juan.chiachioruano[at]nottingham.ac.uk

Abstract

One of the main challenges when analyzing and modelling complex systems using Petri nets is to deal with uncertain information, and moreover, to be able to use such uncertainty to dynamically adapt the modelled system to uncertain (changing) contextual conditions. Such self-adaptation relies on some form of learning capability of the Petri net, which can be hardly implemented using the existing Petri net formalisms. This paper shows how uncertainty management and self-adaptation can be achieved naturally using Plausible Petri Nets, a new Petri net paradigm recently developed by the authors [*Information Sciences*, 453 (2018) pp. 323-345]. The methodology is exemplified using a case study about railway track asset management, where several track maintenance and inspection activities are modelled jointly with a stochastic track geometry degradation process using a Plausible Petri net. The resulting expert system is shown to be able to autonomously adapt to contextual changes coming from noisy condition monitoring data. This adaptation is carried out taking advantage of a Bayesian updating mechanism which is inherently implemented in the execution semantics of the Plausible Petri net.

Keywords: Uncertain information, Bayesian learning, Plausible Petri nets, Infrastructure Asset Management

1 Introduction

A Petri net (PN) is a mathematical and graphical modelling tool first introduced by Carl Petri in 1962 [1] for the analysis of the dynamic behaviour of sequential asynchronous automatons. Since then, the PN methodology has been greatly expanded and combined with other modelling techniques like fuzzy sets, neural networks, etc., for the modelling of complex processes and intelligent systems [2]. The main concepts relative to the theory of PNs are summarised in [3], while a tutorial for practical engineering applications can be found in [4]. A known limitation of the original PN formalism is the difficulty to deal with uncertain information during runtime, and moreover, its inability to use such information to dynamically accommodate the modelled system to new environmental and contextual conditions [5, 6]. In the literature, a number of PN variants have been introduced in order to enhance the original PN formalism with improved rules for uncertainty quantification and self-adaptation [7, 8]. However, none of the existing PN formalisms provides direct means to efficiently deal with uncertainty while considering the hybrid systems, consisting of a combination of discrete and continuous processes whose evolution may be uncertain [9].

This paper overviews a recent PN formalism proposed by the authors in [9, 10], known as Plausible Petri nets (PPNs), whereby discrete events (e.g., go/no-go decisions, intervention activities, etc.) can be modelled together with continuous processes whose evolution may be uncertain (e.g. a component deterioration process). In particular, this paper highlights through a case study how uncertain information and self-adaptation can be achieved naturally using PPNs, since they have an inherent Bayesian updating mechanism implemented within their execution rules. As an illustrative example, a railway track asset management problem is idealised using a PPN model, whereby a number of operational rules (e.g., maintenance operations and inspection activities) are considered together with a continuous-variable stochastic process for track geometry degradation [11]. Through simulation, the operational rules are shown to be autonomously and adaptively triggered based on the *degree of belief* [12] about the state of degradation of the track, which is sequentially *updated* as long as contextual changes in the form of (noisy) condition monitoring data become available.

The remainder of the paper is organised as follows. Section 2 presents the fundamentals of the classical Petri nets. In Section 3, an overview of the PPN paradigm is provided. Section 4 is devoted to present the case study about railway track asset management modelling using PPNs. Finally, Section 5 provides concluding remarks.

2 Basics of Petri nets modelling

Petri nets (PN) [1] are typically regarded as a powerful modelling tool for complex systems, especially when system-level operational nonlinearities (e.g., resource availability, concurrency and synchronisation of components, etc.) need to be considered in the analysis. PNs are based on a graphical and mathematical language with well-established execution semantics, which greatly increases their suitability for modelling complex distributed systems and systems of systems [2, 13].

From a graphical point of view, PNs are composed by two types of nodes, namely, *places* (represented by circles) and *transitions* (represented by boxes). The nodes are connected by *arcs* either from places to transitions or vice versa. In most practical applications, the places represent particular discrete states of the system being modelled (e.g. the health state of a component), while the transitions represent processes that enable the system to move from one place to another. The places contain *tokens* that travel through the net to other places depending on the *firing* of the transitions [3]. The presence of a token in a particular place can be interpreted as holding the truth of the condition or information represented by that place (e.g., "component failed"), and the distribution of tokens over the PN at a specific time is referred to as the *marking* of the net, which is expressed as a vector indicative of the *state* of the PN. A particular transition t is fired only if all places leading to that transition have at least one token. Those places define the *pre-set* of transition t, denoted by ${}^\bullet t$. After firing the transition, one token is added to each of its output places, thus defining the *post-set* of the transition, referred to as $t^\bullet$. Arcs are labeled with their corresponding *weights*, which are non-negative integer values indicating the amount of parallel arcs (1 by default). In Figure 1, an illustration of a sample PN comprised of three places (p_1, p_2, p_3) and one transition (t_1) is depicted.

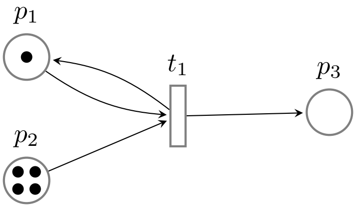


Figure 1: Example of Petri net comprised of three places (p_1, p_2, p_3) and one transition (t_1).

From a mathematical perspective, a PN can be 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 the set of places, $\mathbf{T} \in \mathbb{N}^{n_t}$ is the set of transitions, $\mathbf{E} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ is the set of 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 (initial marking).

At a particular time $k \in \mathbb{N}$, the dynamics of the overall PN can be described using the *state transition equation*, which is mathematically expressed as [3]:

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

where $\mathbf{M}_k$ is the marking of the PN at time k; $\mathbf{u}_k = (u_{1,k}, u_{2,k}, \ldots, u_{n_t,k})^T$ is the *firing vector* at time k, with $u_{i,k} = 1$ if transition t_i is fired, and $u_{i,k} = 0$ otherwise; and $\mathbf{A}$ is the $n_t \times n_p$ *incidence matrix*, whose (i, j)-th element is obtained as $a_{ij} = a_{ij}^+ - a_{ij}^-$, $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$, with a_{ij}^+ being the weight of the arc from transition t_i to output place p_j, whereas a_{ij}^- is the weight of the arc from input place p_j to transition t_i. Therefore, using Equation (1), the evolution of the marking and thus the dynamics of the overall system can be simulated time to time.

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:

$$M(j) \geqslant a_{ij}^- \;\; \forall p_j \in \;\; {}^\bullet t_i \tag{2}$$

where $M(j) \in \mathbb{N}$ is the marking for place p_j. Note that in practical engineering applications, transitions are typically assigned with time delays which are useful for task scheduling modelling and performance evaluation in dynamical systems [3]. In such cases, a transition is fired once its time delay has passed. The resulting PNs are known as *Timed Petri nets* if the delays are deterministic, and *Stochastic Petri nets* if the delays are uncertain, represented by probability distributions.

3 Dealing with uncertainty using Plausible Petri nets

PNs, as originally conceived by Carl A. Petri [1], are well-suited to represent the dynamics of complex systems and processes since they provide a graphical support for system idealization, but also because they rely on rigorous mathematical principles which enable their computational implementation and simulation. However, a typical criticism of PNs is that they are not adequate to deal with uncertain information (e.g., uncertain system states, uncertain processes, etc.) [8, 14], nor do they consider the hybrid nature of real-world dynamical systems, consisting of a combination of discrete and continuous processes whose evolution may be uncertain [9].

Plausible Petri nets (PPNs) are a variant of PNs recently developed by the authors in [9, 10], where the uncertainty is rigorously accounted for through *states of information*, which provide a mapping that assigns to each possible numerical value of the state variable its relative plausibility. Two types of processes can be jointly simulated using PPNs; namely, discrete processes, where the tokens are objects in the sense of integer moving units, as in classical PNs; and continuous processes using numerical variables (e.g., a degradation process), where the tokens are probability density functions (referred to as states of information) which are transferred through the net based on particular execution rules. Thus, in PPNs, the overall net is partitioned into two disjoint subnets, namely the *numerical* subnet and the *symbolic* subnet, which evolve interactively under the same execution rules. These are denoted using the superscripts $(\mathcal{N})$ and $(\mathcal{S})$, respectively. Specific details about the PPN paradigm can be found in [9], but here, the key differences between PNs and PPNs are provided for the sake of clarity and better readability.

3.1 Execution rules of PPNs

In PPNs, the referred states of information about a system state variable $\mathbf{x}_k \in \mathcal{X}$ are denoted by the probability density functions (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 the overall PPN at a particular time k consists in a combined vector $\mathbf{M}_k = (\mathbf{M}_k^{(\mathcal{N})}, \mathbf{M}_k^{(\mathcal{S})})$, where $\mathbf{M}_k^{(\mathcal{N})}$ is the marking for the numerical subnet (consisting of a column vector of normalised PDFs) and $\mathbf{M}_k^{(\mathcal{S})}$ is the marking of the symbolic subnet (consisting of a column vector of discrete values). While the marking evolution of the symbolic subnet is given by the state transition equation in (1), the marking evolution of the numerical subnet relies on two basics operations for information flow dynamics: the *conjunction* and *disjunction* of states of information [12, 15]. Using these operations, the logic operators AND ($\wedge$) and OR ($\vee$) are invoked to enable the information exchange across the numerical subnet by respectively combining and aggregating states of information [9]. An illustrative explanation of the conjunction and disjunction of states of information operations is given in Figure 2. From this standpoint, the dynamics of PPNs can be formulated by adopting of the following rules [9]:

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

2. Transition t_i produces to an output arc a state of information given by $a_{ij}^{+}\big(f^{\bullet t_i} \wedge f^{t_i}\big)(\mathbf{x}_k)$, where $f^{\bullet t_i}(\mathbf{x}_k)$ is the PDF resulting from the disjunction of the states of information of the pre-set of t_i, given by $f^{\bullet t_i}(\mathbf{x}_k) = \frac{1}{\beta}\big(f^{p_1} + f^{p_2} + \cdots + f^{p_m}\big)(\mathbf{x}_k)$, where β is a normalising constant, and $p_1, \ldots, p_m \in {}^{\bullet}t_i \subset \mathbf{P}^{(\mathcal{N})}$;

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

$$f^{p_j}(\mathbf{x}_{k+1}) = \Big(f^{p_j} \vee a_{ij}^{+}\big(f^{t_i} \wedge f^{\bullet t_i}\big)\Big)(\mathbf{x}_k) \tag{3}$$

Note that the execution rules given above for PPNs are analytically intractable except for very simple cases, since the conjunction of states of information requires the evaluation of normalising constants involving an intractable integral. For these cases, a numerical method using *particles* [17] is proposed in [9] to compute the conjunction of states of information while circumventing the evaluation of the normalising constants. Note also in the execution rules that PPNs have an inherent Bayesian learning mechanism implemented in their execution semantics. In particular, if one observes rule 1 and just assumes that $f^{t_i}(\mathbf{x}_k)$ acts as likelihood function for a set of data $\mathbf{y}_k \in \mathcal{D}$ (i.e. $f^{t_i}(\mathbf{x}_k) = p(\mathbf{y}_k|\mathbf{x}_k)$), and that $p_j^{(\mathcal{N})}$ represents a *prior* PDF of $\mathbf{x}_k$, the conjunction of both states of information leads to a *posterior* PDF of the state variable, assuming that the $\mathcal{X}$-space is a linear space [15]. This interesting property of PPNs is further exploited within the context of an engineering case study.

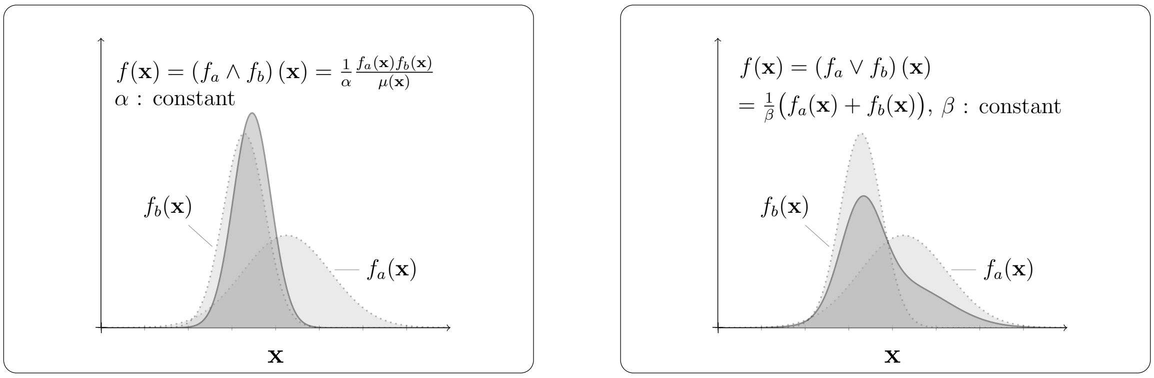


Figure 2: Conjunction (left panel) and disjunction (right panel) of two arbitrary states of information, $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$. The term $\mu(\mathbf{x})$ is the *homogeneous density function* [15, 16], representing a state of complete ignorance about the stochastic variable $\mathbf{x} \in \mathcal{X}$.

4 Case study

In this section, an engineering case study about railway track asset management is provided to illustrate how uncertain information and condition monitoring data can be integrated using PPNs to support autonomous and adaptive decisions about infrastructure degradation and maintenance. To this end, a PPN model is developed to idealise an expert system for railway track asset management, incorporating a physics-based model for track geometry degradation, condition monitoring data, along with inspection activities. The PPN is depicted in Figure 3. For this case study, the deterioration of the track is assumed to occur due to traffic loadings (expressed in load cycles), and also, it is assumed that the track geometry degradation can be periodically measured using train-borne sensors. In this sense, every time a new measurement is available, the PPN incorporates that data to *update* the underlying track degradation model so that further inspections and maintenance activities are autonomously triggered based on up-to-date model predictions, instead of using simply the raw data as a base for decision making.

Observe in Figure 3 that the PPN is comprised of one numerical place, $p_1^{(\mathcal{N})}$ (containing the track degradation model), five symbolic places, $p_2^{(\mathcal{S})}$ to $p_6^{(\mathcal{S})}$ (representing the activation of inspection activities and other management variables), three mixed transitions, t_1 to t_3, and two symbolic transitions, t_4, t_5. The formulation and implementation details of the adopted track degradation model can be found in [11], so they are not repeated here. The condition monitoring data used for this case study consist in a set of measurements about track settlement $Y = \{y_i, y_j, \dots, y_k\}$ taken from the literature [18], which are sequentially introduced to the system at loading cycles $\{i, j, \dots, k\} \subset \mathbb{N}$. In Figure 3, a token in place $p_1^{(\mathcal{S})}$ represents a data point arrived to the system. The dataset is reproduced in Table 1. The measurements are assumed to come with a 5% white-noise type error, therefore $y_k \sim \mathcal{N}(x_k, \sigma_{w_k})$, with $\sigma_{w_k} = 0.05\|y_k\|$. This PDF represents the *likelihood* of the measurements, and comprises a state of information within transition t_1, given by $f^{t_1}(x_k) = \mathcal{N}(x_k, \sigma_{w_k})$.

By evaluating the proposed PPN-based expert system, changes in the numerical and discrete states of the system are obtained based on a number of automated actions which are activated through firing transitions t_1 to t_5. A summary of the functions defining the transitions is provided in Table 2. The execution rules given in Section 3.1 are applied to obtain the overall system evolution described through the marking $\mathbf{M}_k$, $k > 0$. In particular, observe that each time a new measurement is available, transition t_1 is enabled, which leads to the conjunction of the states of information of $p_1^{(\mathcal{N})}$ and t_1. As explained in Section 3.1, this conjunction conveys a Bayesian updating of the degradation variable x_k using monitoring data, as per Table 1. The results for the updated degradation variable along with its $5\% - 95\%$ confidence interval are depicted in Figure 4a (see the leftmost panel). In panel 4b, the temporal evolution of the

Table 1: Railway track settlement (strain) data used for calculations taken from [18].

Loading cycles ($\times\mathbf{10^3}$)	0	0.625	1.25	2.5	5	10	20	30	50	75
Plastic strain	0	0.0017	0.0045	0.0058	0.0075	0.0087	0.0104	0.011	0.012	0.01275

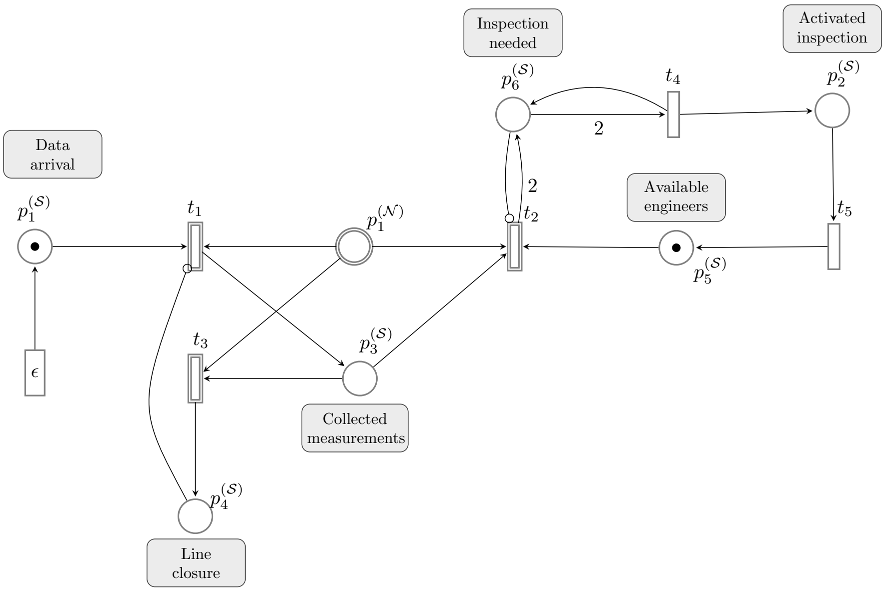


Figure 3: PPN of the case study presented in Section 4. Note that a *cold transition* (ϵ) is used to represent the data arrival, which are assumed to be available at a set of non-regularly scheduled time instants, as shown in Table 1.

uncertainty (differential entropy, DE) in the estimation of x_k within place $p_1^{(\mathcal{N})}$ is shown, with indication of the reference level when inspections are needed. The observed drops in the sequence of DE values in Figure 4b correspond to the uncertainty reduction due to Bayesian learning of the track degradation model when new measurements become available.

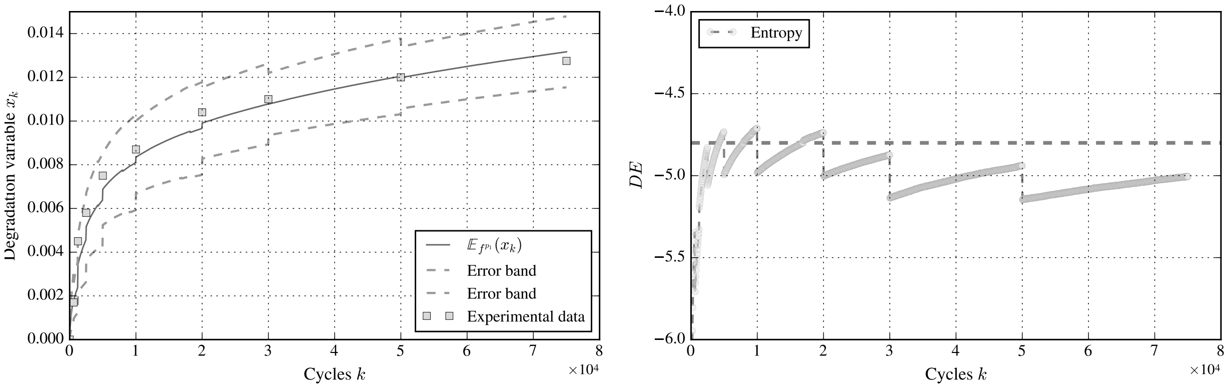
Table 2: Description of the transitions adopted in PPN depicted in Figure 3. PI: Periodic inspections, OI: opportunistic inspections, LC: Line Closure.

ID	Type	Rule	State of information	Action
t_1	Mixed	–	$f^{t_1} \sim p(y_k \vert x_k)$ (Likelihood)	Update predictions
t_2	Mixed	$H(x_k) \geqslant -4.8$	$f^{t_2} \sim \mathbb{I}_{\mathcal{C}_2}(x_k)$	Activates OI
t_3	Mixed	$\mathbb{E}_{f^{p_1}}[x_k] \geqslant 0.014$ [m]	$f^{t_3} \sim \mathbb{I}_{\mathcal{C}_3}(x_k)$	Switches to LC
t_4	Symbolic	$\tau_4 \sim \mathcal{N}(1,1)$ (delay)	–	Activates OI
t_5	Symbolic	$\tau_5 \sim \mathcal{N}(24,1)$ (delay)	–	Concludes inspections

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 a numerical evidence of the Bayesian learning taking place in $p_1^{(\mathcal{N})}$, and therefore, an evidence of the self-adaptiveness of the PPN from monitoring data. Correspondingly, the activated inspections (represented in $p_2^{(\mathcal{S})}$) are adaptively triggered based on the updated degradation variable, following the rules given in Table 2.

5 Conclusions

This paper provided an overview of the modelling capabilities of the recently proposed Plausible Petri nets (PPNs), with special emphasis on their ability to deal with uncertain information and self-adaptation in complex systems modelling. In particular, an engineering case study about railway track asset management was provided to illustrate the real-world problems that can be modelled using PPNs. The results revealed how uncertain information and condition monitoring data can be integrated using PPNs to support autonomous and adaptive decisions about infrastructure inspection and maintenance. This example reveals that smart infrastructure asset management can be achieved using novel computational tools like PPNs, which has the potential to not only reduce the expenditure of a country in infrastructure asset management, but also to dramatically change the way the infrastructures are designed and managed.



(a)

(b)

Figure 4: Left: Plot of mean values and probability bands of $f_k^{p_1}$ for $k = 0 \to 75,000$. 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.

Acknowledgments

The authors would like to thank the EPSRC and RSSB who jointly and equally support grant EP/M023028/1 "Whole-life Cost Assessment of Novel Material Railway Drainage Systems", and also the Lloyd's Register Foundation which partially provides support to this work. RSSB is a rail industry body. Through research, analysis, and insight, RSSB supports its members and stakeholders to deliver a safer, more efficient and sustainable rail system. The Lloyd's Register Foundation is a charitable foundation in the U.K. helping to protect life and property by supporting engineering-related education, public engagement, and the application of research.

References

[1] C. A. Petri, Ph.D. thesis, Institut f́ur Instrumentelle Mathematik an der Universit́at Bonn (1962).

[2] W. Pedrycz and F. Gomide, A generalized fuzzy Petri net model, *IEEE Transactions on Fuzzy Systems* **2**, 295 (1994).

[3] T. Murata, Petri nets: Properties, analysis and applications, *Proceedings of the IEEE* **77**, 541 (1989).

[4] R. Zurawski and M. Zhou, Petri nets and industrial applications: A tutorial, *IEEE Transactions on Industrial Electronics* **41**, 567 (1994).

[5] J. Kastner and S. Hong, A review of expert systems, *European Journal of Operational Research* **18**, 285 (1984).

[6] 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* **76**, 85 (2017).

[7] C. G. Looney, Fuzzy Petri nets for rule-based decision making, *IEEE Transactions on Systems, Man, and Cybernetics* **18**, 178 (1988).

[8] K.-Q. Zhou and A. M. Zain, Fuzzy Petri nets and industrial applications: a review, *Artificial Intelligence Review* **45**, 405 (2016).

[9] M. Chiachío, J. Chiachío, D. Prescott, and J. Andrews, A new paradigm for uncertain knowledge representation by Plausible Petri nets, *Information Sciences* **453**, 323 (2018).

[10] 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.* (IEEE, 2016), 165–172.

[11] J. Chiachío, M. Chiachío, D. Prescott, and J. Andrews, A reliability-based prognostics framework for railway track management, in *Proceedings of the Annual Conference of the Prognostics and Health Management* (PHM, 2017), 396–406.

[12] G. Rus, J. Chiachío, and M. Chiachío, Logical inference for inverse problems, *Inverse Problems in Science and Engineering* **24**, 448 (2016).

[13] X. Li and F. Lara-Rosano, Adaptive fuzzy Petri nets for dynamic knowledge representation and inference, *Expert Systems with Applications* **19**, 235 (2000).

[14] H.-C. Liu, J.-X. You, Z. Li, and G. Tian, Fuzzy Petri nets for knowledge representation and reasoning:

A literature review, *Engineering Applications of Artificial Intelligence* **60**, 45 (2017).

[15] A. Tarantola and B. Valette, Inverse problems = quest for information, *Journal of Geophysics* **50**, 159 (1982).

[16] K. Mosegaard and A. Tarantola, in *International Handbook of Earthquake and Engineering Seismology, Part A* (Academics Press Ltd, 2002), vol. 81 of *International Geophysics*, 237–265.

[17] A. Doucet, N. De Freitas, and N. Gordon, in *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, N. De Freitas, and N. Gordon (Springer, 2001), 3–14.

[18] B. Aursudkij, G. R. McDowell, and A. C. Collop, Cyclic loading of railway ballast under triaxial conditions and in a railway test facility, *Granular Matter* **11**, 391 (2009).