Fundamenta Informaticae 47 (2001) 307–323
IOS Press

Sensor, Filter, and Fusion Models with Rough Petri Nets

James F. Peters, S. Ramanna, M. Borkowski*
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB R3T 5V6 Canada
email: jfpeters,ramanna,maciey@ee.umanitoba.ca

Andrzej Skowron‡
Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
email: skowron@mimuw.edu.pl

Zbigniew Suraj†
Department of Foundations of Computer Science
University of Information Technology and Management
35-225 Rzeszów, ul. H. Sucharskiego 2, Poland
e-mail: zsuraj@merkury.wsiz.rzeszow.pl

Abstract. This paper considers models of sensors, filters, and sensor fusion with Petri nets defined in the context of rough sets. Sensors and filters are fundamental computational units in the design of systems. The intent of this work is to construct Petri nets to simulate conditional computation in approximate reasoning systems, which are dependent on filtered input from selected sensors considered relevant in problem solving. In this paper, coloured Petri nets provide a computational framework for the definition of a family of Petri nets based on rough set theory. Sensors are modeled with what are known as receptor processes in rough Petri nets. Filters are modeled as Łukasiewicz guards on

*Address for correspondence: Department of Electrical and Computer Engineering, University of Manitoba, 15 Gillson St., ENGR 504, Winnipeg, Manitoba R3T 5V6 Canada.
‡Address for correspondence: Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
†Address for correspondence: Department of Foundations of Computer Science, University of Information Technology and Management, 35-225 Rzeszów, ul. H. Sucharskiego 2, Poland

some transitions in rough Petri nets. A Łukasiewicz guard is defined in the context of multivalued logic. Łukasiewicz guards are useful in culling from a collection of active sensors those sensors with the greatest relevance in a problem-solving effort such as classification of a ”perceived” phenomenon in the environment of an agent. The relevance of a sensor is computed using a discrete rough integral. The form of sensor fusion considered in this paper consists in selecting only those sensors considered relevant in solving a problem. The contribution of this paper is the modeling of sensors, filters, and fusion in the context of receptor processes, Łukasiewicz guards, and rough integration, respectively.

Keywords: approximation, enabling, filter, fusion, guard, multivalued logic, Petri net, rough measure, rough integral, rough sets, sensor.

1. Introduction

Considerable work has already been carried out in modeling various forms of systems with Petri nets in the context of rough sets (see, for example, [21]-[26],[27],[29],[30]-[34],). The aim of the earlier as well as the current research has been to provide a complete framework for approximate reasoning, especially in the context of rough set theory from Pawlak [13]-[18]. Rough set theory also provides an inductive approach to reasoning about data. This paper returns to the idea of rough Petri nets in [23]-[26]. Guarded transitions are conceptualized in the context of multivalued logic from Łukasiewicz [11] and rough sets [13]-[14]. Such transitions provide the basis for the design Petri net models of sensor filters, which are fundamental in the design of rough neural computing systems. Dill receptor processes are used to define input places in sensor-driven systems. Łukasiewicz guards are introduced to provide conditional firing to a degree of one or more transitions in Petri net models of dynamical systems. Rough integrals are used to design particular forms of Łukasiewicz guards. The contribution of this paper is the modeling of sensors as receptor processes, filters as Łukasiewicz guards, and fusion of sensors considered relevant in a problem-solving effort, respectively.

This paper is structured as follows. A brief presentation of a rough set approach to set approximation and a form of rough membership function is given in Section 2. Rough measures and rough integrals are introduced briefly in Section 3. Rough Petri nets, receptor processes, and Łukasiewicz guards are presented in Section 4. Petri net models of sensors and filters are given in Section 5. The culmination of these ideas appears in a Petri net model of sensor fusion in Section 6.

2. Basic Concepts of Rough Sets and Rough Measures

Rough set theory offers a systematic approach to set approximation [13]-[14]. In this paper, $\wp(X)$ denotes the set of all subsets of X and $card(X)$ denotes the number of elements of set X.

2.1. Set Approximation

To begin, let $S = (U, A)$ be an information system where U is a non-empty, finite set of objects and A is a non-empty, finite set of attributes, where $a : U \to V_a$ for every $a \in A$. For each $B \subseteq A$, there is associated an equivalence relation $Ind_A(B)$ such that

$$Ind_A(B) = \{(x, x') \in U^2 \mid \forall a \in B\ a(x) = a(x')\}$$

If $(x, x') \in Ind_A(B)$, we say that objects x and x' are indiscernible from each other relative to attributes from B. The notation $[x]_B$ denotes equivalence classes of $Ind_A(B)$. Further, partition $U/Ind_A(B)$ denotes the family of all equivalence classes of relation $Ind_A(B)$ on U. For $X \subseteq U$, the set X can be approximated only from information contained in B by constructing a B-lower and B-upper approximation denoted by $\underline{B}X$ and $\overline{B}X$ respectively, where $\underline{B}X = \{x \mid [x]_B \subseteq X\}$ and $\overline{B}X = \{x \mid [x]_B \cap X \neq \emptyset\}$.

2.2. Rough Membership Functions

In this section, a set function form of the traditional rough membership function is presented.

Definition 2.1. Let $S = (U, A)$ be an information system, and let $\wp(U)$ denote the powerset of $U, B \subseteq A, u \in U$ and let $[u]_B$ be an equivalence class of an object $u \in U$ of $Ind_A(B)$. The set function

$$\mu_u^B : \wp(U) \rightarrow [0, 1], \text{ where } \mu_u^B(X) = \frac{card(X \cap [u]_B)}{card([u]_B)} \tag{1}$$

for any $X \in \wp(U)$ is called a *rough membership function*. A rough membership function provides a classification measure inasmuch as it tests the degree of overlap between the set X in $\wp(U)$ and equivalence class $[u]_B$. The form of rough membership function in Def. 2.1 is slightly different from the classical definition where the argument of the rough membership function is an object x and the set X is fixed [18].

Example 2.2.

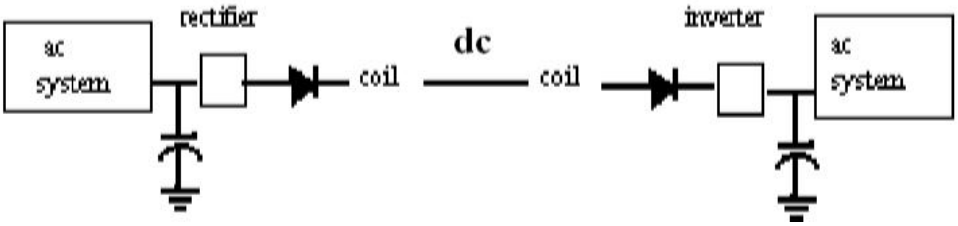

Figure 1. Sample dc Link Between Two ac Systems

A high voltage direct current (dc) transmission system connected between alternating current (ac) source and ac power distribution system has two converters. The notation ac (or dc) is commonly used instead of AC (or DC) in electrical engineering (see, for example, [1],[8],[10],[12]). In the case where the flow of power is from the ac side to the dc side as in Fig. 1, then a converter acts as a rectifier in changing ac to dc. A device that converts dc power into ac power at desired output voltage and frequency is called an inverter [1]. The Dorsey Station in the Manitoba Hydro system, for example, acts as an inverter in converting dc power received from hydroelectric plants in northern Manitoba to ac power, which is distributed throughout North America.

Power system faults are recorded in fault files. Consider, next, the information system $S = (A, F)$, where A is a set of attributes such as phase current, current setting, and maximum phase current, and

F is a set of fault files. Assume that $\overline{A}F = \{\text{file3}, \text{file4}, \text{file7}, \text{file8}\}$. Further, assume that $[f_3]_A = \{$ equivalence class consisting of files representing a known power system fault$\} = \{\text{file3}, \text{file9}, \text{file10}\}$. Then consider the degree of overlap between $\overline{A}F$ and $[f_3]_A$ (see Fig. 2).

$$\mu_{f_3}^{A}\left(\overline{A}F\right) = \frac{\left|\overline{A}F \cap [f_3]_A\right|}{\left|[f_3]_A\right|} = \frac{1}{3}$$

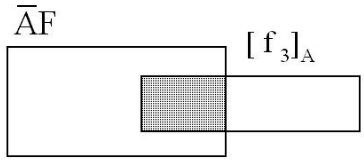

Fig. 2 Overlapping regions

3. Rough Measures and Integrals

This Section gives a brief introduction to rough measures and rough integrals [19]-[20]. We assume X and U are finite, non-empty sets.

3.1. Additive Set Functions

Let us recall some basic definitions [7].

Definition 3.1. A function $\lambda : \wp(X) \to \mathbb{R}$ where $\mathbb{R}$ is the set of all real numbers is called a *set function* on X.

Definition 3.2. Let λ be a set function on X. The function λ is said to be *additive* on X iff $\lambda(A \cup B) = \lambda(A) + \lambda(B)$ for every $A, B \in \wp(X)$ such that $A \cap B = \emptyset$ (i.e., A and B are disjoint subsets of X).

Definition 3.3. A set function λ on X is called to be *non-negative* on X iff $\lambda(Y) \geq 0$ for any $Y \in \wp(X)$.

Fact 3.4. The rough membership function $\mu_x^B : \wp(X) \to [0, 1]$ is a non-negative and additive set function on X.

3.2. Rough Measures

Let $S = (U, A)$ be an information system, $X \subseteq U, B \subseteq A$, and let $Ind_A(B)$ be the indiscernibility relation on U.

Definition 3.5. The tuple $(X, \wp(X), U/Ind_A(B))$, where $U/Ind_A(B)$ denotes a set of all equivalence classes determined by $Ind_A(B)$ on U, is called an *indiscernibility space* over X and B.

Definition 3.6. Let $u \in U$. A non-negative and additive set function $\rho_u : \wp(X) \to [0, \infty)$ defined by $\rho_u(Y) = \rho'(Y \cap [u]_B)$ for $Y \in \wp(X)$, where $\rho' : \wp(X) \to [0, \infty)$ is a set function, is called a *rough measure* relative to $U/Ind_A(B)$ and u on the indiscernibility space $(X, \wp(X), U/Ind_A(B))$.

Definition 3.7. Let ρ_u for $u \in U$ be a *rough measure* on the indiscernibility space $(X, \wp(X), U/Ind_A(B))$ for $u \in U$. The tuple $(X, \wp(X), U/Ind_A(B), \{\rho_u\}_{u \in U})$ is a *rough measure space* over X and B.

Proposition 3.8. [20] $(X, \wp(X), U/Ind_A(B), \{\mu_u^B\}_{u\in U})$ is a rough measure space over X and B.

3.3. Discrete Rough Integrals

Rough integrals were introduced in [19], and elaborated in [20]. In this Section, we consider a variation of the Lebesgue integral, the discrete Choquet integral defined relative to a rough measure. In what follows, let $X = \{x_1, \ldots, x_n\}$ be a finite, non-empty set with n elements. The elements of X are indexed from 1 to n. By $(\bullet)$ we denote a permutation of the set $\{1, \ldots, n\}$ and (i) denotes its value for i. The notation $X_{(i)}$ denotes the set $\{x_{(i)}, x_{(i+1)}, \ldots, x_{(n)}\}$ where $i \geq 1$ and $n = card(X)$. The subscript (i) is called a permutation index because the indices on elements of $X_{(i)}$ are chosen after a reordering of the elements of X. This reordering is "induced" by an external mechanism.

Example 3.9. Let $X = \{x_1, x_2\}$ the function $a : X \to \mathbb{R}^+$ where $\mathbb{R}^+$ is the set of non-negative real numbers, be defined such that $a(x_1) = 2001, a(x_2) = 44$. That is, $a(x_1) \geq a(x_2)$. Then, after reordering the elements of X and assigning permutation indices to the reordered elements, we obtain $a(x_{(1)}) \leq a(x_{(2)})$ where $x_{(1)} = x_2$ and $x_{(2)} = x_1$; $X_{(1)} = \{x_1, x_2\}$, $X_{(2)} = \{x_1\}$. Next, we use a functional defined by Choquet in 1953 in capacity theory [3].

Definition 3.10. Let ρ be a rough measure on X where the elements of X are denoted by $x_1, \ldots, x_n$. The discrete Choquet integral of $f : X \to \mathbb{R}^+$ with respect to the rough measure ρ is defined by

$$(C) \int f d\rho = \sum_{i=1}^{n} (f(x_{(i)}) - f(x_{(i-1)}))\rho(X_{(i)})$$

where $\bullet_{(i)}$ specifies that indices have been permuted so that $0 \leq f(x_{(i)}) \leq \cdots \leq f(x_{(n)})$, $X_{(i)} := \{x_{(i)}, \ldots, x_{(n)}\}$, and $f(x_{(0)}) = 0$.

This definition of the Choquet integral is based on a formulation in Grabisch [6]. The rough measure $\rho(X_{(i)})$ value serves as a "weight" of a coalition (or combination) of objects in set $X_{(i)}$ relative to $f(x_{(i)})$. It should be observed that in general the Choquet integral has the effect of "averaging" the values of a measurable function. This averaging closely resembles the well-known Ordered Weighted Average (OWA) operator [35].

Definition 3.11. Ordered Weighted Average (OWA). Let $a_1, \ldots, a_n$ be real-numbers that are ordered so that $a_{(1)} \leq \cdots \leq a_{(n)}$, and let $w_1, \ldots, w_n$ a set of weights such that $\sum_{i=1}^{n} w_i = 1$. The OWA operator is defined as follows:

$$\mathrm{OWA}_{w_1,\ldots,w_n}(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i a_{(i)}$$

We are interested in integrating an attribute in the search for attributes that "score" well relative to weighted sets $X_{(i)}$ of ordered objects.

3.4. Relevance of an Attribute

In this section, we consider the measurement of the relevance of an attribute using a rough integral. The measure (1) is fundamental in computing an "average" sensor value. Intuitively, we want to identify those sensors with outputs closest to some threshold.

Example 3.12. Let $\{a\} = B \subseteq A$ where $a : U \to [0, 0.5]$ where $a(x)$ is rounded to two decimal places. Let $(Y, U-\mathrm{Y})$ will be a partition defined by an expert e and let $[u]_e$ denote a set in this partition containing u for a fixed $u \in U$. We assume a decision system (X_a, a, e) is given for any considered attribute (sensor) a such that $X_a \subseteq U$, $a : X_a \to \Re_+$ and e is an expert decision restricted to X_a defining a partition $(Y \cap X_a, (U-\mathrm{Y}) \cap \mathrm{X_a})$ of X_a. Moreover, we assume $X_a \cap [u]_e \neq \emptyset$. Consider the following decision tables.

Table 1(a)

$X_a \setminus \{a, e\}$	a	e
x_1=0.203	0.2	0
x_2=0.454	0.45	1
x_3=0.453	0.45	1
x_4=0.106	0.11	0
x_5=0.104	0.10	0

Table 1(b)

$X_a \setminus \{a, e\}$	a	e
x_2= 0.454	0.45	1
x_9= 0.455	0.46	1
x_{10}= 0.401	0.4	1
x_{11}= 0.407	0.41	1
x_{12}= 0.429	0.43	1

From Table 1(a), $(C) \int a \, d\mu_u^e = 0.1$ From Table 1(b), $(C) \int a \, d\mu_u^e = 0.239$

In Table 1(a) and Table 1(b) the set X_a represents the set of objects observed by sensor a in time while $[u]_e$ denotes a fixed a priori set of objects. The decision d is 1 for objects from $[u]_e$. The goal is to estimate how *close* the sensor measurements are to $[u]_e$.

From these two cases, it can be seen the relevance of attribute improves as the value of the rough integral increases in value. For a particular $[u]_e$, the rough integral measures the relevance of an attribute for a particular table in a classification effort. That is, the integral (over all values for a given attribute) reflects in a sense the degree of definability of the partition of objects created by singletons by a partition defined by the values of a given attribute.

One can observe that the following property holds for rough integrals.

Proposition 3.13. Let $0 < s \leq r$. If $a(x) \in [s, r]$ for all $x \in X_a$, then $\int a \, d\mu_u^e \in (0, r]$ where $u \in U$.

Proof: Since $0 < s \leq r$, it is enough to show that (1) $\int a \, d\mu_u^e \leq r$ and (2) $\int a \, d\mu_u^e > 0$.

(1) We have $\mu_u^e(X_{(i)}) \leq 1$, $B = \{e\}$. Hence

$$\int a \, d\mu_u^e = \sum_{i=1}^{n} (a(x_{(i)}) - a(x_{(i-1)}))\mu_u^e(X_{(i)}) \leq \sum_{i=1}^{n} (a(x_{(i)}) - a(x_{(i-1)})) =$$

$$a(x_{(n)}) \leq r.$$

(2) From the assumptions, there exists at least one $x \in [u]_e$. Hence, $\exists_{k\geq 1} : \mu_u^e(X_{(k)}) > 0$. After reordering of the subsets of $\wp(X)$, we know that μ_u^e is a non-increasing, non-negative function. Hence, $\mu_u^e(X_{(1)}) > 0$. Consider the case where $a(x_{(1)}) = s$. Then compute

$$\int a\, d\mu_u^e = \sum_{i=1}^{n} (a(x_{(i)}) - a(x_{(i-1)}))\mu_u^e(X_{(i)}) \geq (a(x_{(1)}) - a(x_{(0)})) \cdot \mu_u^e(X_{(1)}) =$$

$$a(x_{(1)}) \cdot \mu_u^e(X_{(1)}) > 0.$$

□

Moreover, when the set of sensor a values is close to $[u]_e$, then $\int a\, d\mu_u^e$ is close to the maximal value integral can take for a sensor. Measures of closeness depend on applications and their parameters can be tuned for specific data cases and targets.

4. Rough Petri Nets

A rough Petri net models a process that implements one or more features of rough set theory. Rough Petri nets are derived from coloured and hierarchical Petri nets as well as from rough set theory. Coloured Petri nets provide a well-understood framework for introducing computational mechanisms (data types, variables, constants, and functions) that are useful in describing processes that carry out set approximation, information granulation and engage in approximate reasoning. The new form of Petri net utilizes rough set theory, multivalued logic and receptor process theory to extend coloured Petri nets. Three extensions of coloured Petri nets are described in this article: (1) multivalued guards, (2) receptor processes, and (3) rough computing. Briefly, this is explained as follows. Boolean valued guards in traditional coloured Petri nets are replaced by multivalued guards in rough Petri nets. Let T be a set of transitions. In a coloured Petri net, a guard is a mapping $G : T \rightarrow \{1, 0\}$. A transition is enabled if G returns 1. In keeping with an interest in modeling approximate reasoning, we augment coloured Petri nets with guards of the form $G : T \rightarrow [0, 1]$. With this form of guard, it is possible to model a level-of-enabling of a transition that is more general than the usual "on/off" enabling model. Places in a Petri net represent the states of a system. An input place is a source of input for a method associated with a transition. An output place is repository for results computed by a transition method. In a rough Petri net, an input place can be a receptor process. This form of input place responds to each stimulus from the environment by measuring a stimulus and by making each measurement available to a transition method. This extension of the input place convention in coloured Petri nets is needed to model dynamically changing systems that perform actions in response to sensor signals. There is an iteration "inside" a receptor process that idealizes a typical sensor-action system found in agents. That is, the intent behind a provision of input places that are receptor processes is to model a sensor that enables a response mechanism represented by a transition method each time the sensor is stimulated. Rough computation is the third extension of coloured Petri nets considered in this article. This feature is the hallmark of a rough Petri net. It is characterized by the design of transition methods that compute rough set structures as well as values of measurable functions and rough integrals. This feature is important to us because we are interested in modeling intelligent systems and information granulation. In this article, the modeling of rough computation is limited to a partial model of an information granulation system where granulation results from a fusion of sensor signal values. A rough Petri net also includes a strength-of-connection mapping from arcs to weights. This feature of rough Petri nets is useful in modeling neural computation but is not considered in this article. A rough Petri net provides a basis for modeling, simulating and analyzing approximate reasoning, decision and control systems. In what follows, it is assumed that the reader is familiar with classical Petri nets in Petri [28] and coloured Petri nets in Jensen [9].

Definition 4.1. *Rough Petri Net.* A rough Petri net is a structure ($\sum$, P, T, A, N, C, G, E, I, W, $\Re$, ξ) where

- $\sum$ is a finite set of non-empty data types called color sets;
- P is a finite set of places;
- T is a finite set of transitions;
- A is a finite set of arcs such that P ∩ T = P ∩ A = T ∩ A = ∅;
- N is a 1-1 node function where N: A → (P × T) ∪ (T × P);
- C is a color function where C: P → $\sum$;
- G is a guard function where G: T → [0, 1];
- E is an arc expression function where E: A → Set_of_Expressions where E(a) is an expression of type C(p(a)) and p(a) is the place component of N(a);
- I is an initialization function where I: P → Set_of_Closed_Expressions where I(p) is an expression of type C(p);
- W is a set of strengths-of-connections where ξ : A → W;
- $\Re = \{\rho_\sigma | \rho_\sigma$ is a method $\}$;
- ρ_σ is a method that constructs a rough set structure or computes that a value.

A sample ρ_σ is a method that constructs a rough set structure (e.g., an upper approximation of a set X relative to a set of attributes B, or the set OPT(S) of all rules derived from reducts of a decision system table for an information system S). Another example of ρ_σ is a rough membership function. The availability of guards on transitions makes it possible to model sensor filters and various forms of fuzzy Petri nets. Higher order places representing receptor processes are part of rough Petri nets.

4.1. Receptor Processes

The notion of a receptor process comes from Dill [4]. In a rough Petri net, a receptor process is a higher order place that models a sensor. The input place labeled ?p1 in Fig. 3(a), for example, represents a form of receptive process that accumulates a signal.

A *receptor process* is a process that provides an interface between a system and its environment by recording its response to each stimulus in a finite set of sample sensor values (a signal) whenever stimuli are detected.

In the case where an environment is a source of continuous stimulation, a receptor process continuously enqueues its responses and periodically enables any transition connected to it. The advantage in constructing such a Petri net model of sensor-dependent systems is that it facilitates reasoning about a system design and simulation of the responses of a system to stimuli. For example, let ?p1 be a receptor process; X, a set of inputs (signal) produced by ?p1 and let μ_u^e be a rough membership function. Let $(Y, U-\mathrm{Y})$ will be a partition defined by an expert e and let $[u]_e$ denote a set in this partition containing u for a fixed $u \in U$ (see Fig. 3(a)).

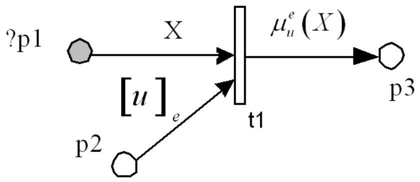

Fig. 3(a) rPN without guard

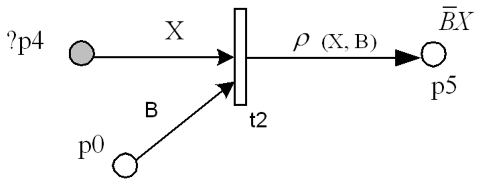

Fig. 3(b) guarded rPN

When transition t1 in Fig. 3(a) fires, $\mu_u^e(X)$ computes a rmf value. The notation X (denoting a set of values "produced" by the receptor process named ?p4) used to label the input to transition t2 in Fig. 3(b) is commonly used in the Petri net models given in this paper. The set X represents a signal or set of sample receptor process values that are accumulated over time. The notation B on the arc from place p0 to transition t2 represents a set of attributes that will be used by method ρ to construct a rough set structure. Whenever t2 fires, ρ constructs the upper approximation $\overline{B}X$. A Petri net model of a receptor process is given in Fig. 4.

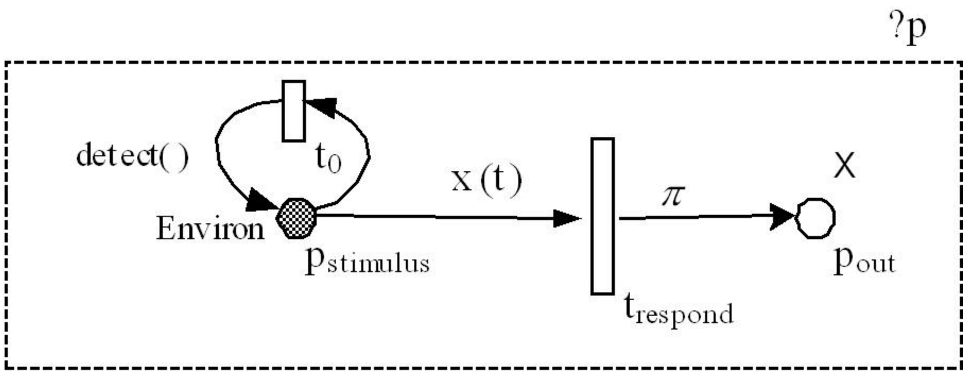

Figure 4. Petri net model of a Receptor Process

Transition $t_{respond}$ in Fig. 4 is enabled whenever it receives a stimulus $x_{stimulus}$. Transition t_0 is continuously enabled. Whenever t_0 fires, it performs a detect operation ("looking" for a stimulus from the environment). In Fig. 4, the function π constructs a set of excitation signal values. The function π describes a sensor as a mapping of an input signal value $x(t)$ (also called a stimulus [5]) in the time domain to some convenient form of output signal. In classical signal processing, a signal is a real-valued function x of time t. Let $x : \mathbb{R}^+ \to \mathbb{R}$ where $\mathbb{R}^+$ is the set of all non-negative reals, denote a signal. The behavior of a sensor over a time interval $[t_0, t_k]$ is represented by a set X of sample sensor values $\{x(t_0), \ldots, x(t_k)\}$. The transition labeled t_1 in Fig. 3(b), for example, is enabled by a sensor signal X and set of attributes B. That is, define $\pi(x(t)) = \{x(t)\}$, if $X = \emptyset$, otherwise $\pi(x(t)) = X \cup \{x(t)\}$.

4.2. Guarded transitions

A guard $G(\text{t})$ is an enabling condition associated with a transition t. In a rough Petri net, various families of guards can be defined which induce a level-of-enabling of transitions. Łukasiewicz guards were introduced in [23]. There are many forms of Łukasiewicz guards. In this section, we consider a guard which is a real-valued, non-negative set function. Let U denote a universe of objects.

Let $(L, \leq)$ be a lattice with the smallest element $\perp$ and all other elements being incomparable. In the paper we assume $L - \{\perp\}$ to be equal an interval of reals.

Definition 4.2. For a given function $\lambda(X)$ from the domain of the variable X into $L - \{\perp\}$ and a condition α, i.e. function from $L - \{\perp\}$ into $\{0, 1\}$ we define the Łukasiewicz guard $P_{\lambda,\alpha}(X)$ as a function from the domain of the variable X into L defined by

$$P_{\lambda,\alpha}(X) = \lambda(X) \text{ if } \alpha(\lambda(X)) = 1, \text{ and } \perp \text{ otherwise.}$$

We assume t labeled by $P_{\lambda,\alpha}(X)$ is enabled iff $\alpha(\lambda(X))$ holds, i.e. $\alpha(\lambda(X)) = 1$. The value $\lambda(X)$ of $P_{\lambda,\alpha}(X)$ is a part of output labeling the edges outgoing from t if t is fired.

Examples of conditions α are $0 < \lambda(X) \leq 1$ or $\lambda(X) \geq b > 0$ where b is a selected $b \in (0, 1]$.

For example, let $b = 0.75$, $X = \{x(t)\}$, and let a Łukasiewicz guard be defined on a transition t2 as in Fig. 3(b), where t2 is enabled for all values of $\lambda(x) \in [0.75, 1]$. If we assume $\lambda(x) = 0.79$, then $\lambda(x) = 0.79$ enables transition t2 (see Fig. 5).

Sample model for a Łukasiewicz guard:

$$\lambda(x) = e^{\left[\frac{-(x-\bar{x})^2}{s^2}\right]}$$

$P_{\alpha,\lambda}(x) = \lambda(x)$ if $\alpha(\lambda(x))$ holds, 0 otherwise. For example $\alpha(x)$ can be defined as follows: $\alpha(x)$ iff $x \in (0.75,1]$. Hence, we have in particular: $P_{\alpha,\lambda}(x) = \lambda(x)$ iff $\alpha(\lambda(x))$ holds.

Sample enabling:

$\bar{x} = 45$

$s = 20$

$\lambda(44) = 0.9975$

$P_{\alpha,\lambda}(44) = \lambda(44) = 0.9975$ that enables transition t

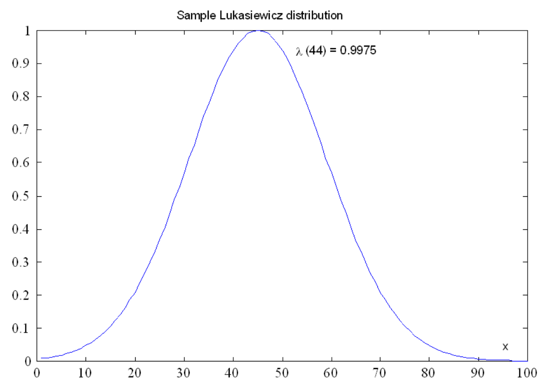

Figure 5. Distribution for a Łukasiewicz Guard

5. Sensors and Filters

Petri net models for sensors and filters were introduced in [26].

5.1. Sensors

A sensor converts some form of stimulus into a measurable output. By contrast with transducers, the output of some sensors might not be some form of energy. For example, the output of a digital thermometer is numerical. Hence, a digital thermometer would be classified as a sensor but not a transducer. Notice, also, that some sensors are also transducers. Most sensors employ one or more transduction mechanisms to produce a usable output signal.

A *sensor* is a device that responds to each stimulus by converting its measured input to some form of usable output.

A sensor responds to each stimulus that it is designed to detect. In effect, a sensor is always input-ready. By definition, a receptor process can be used to model a sensor. Hence, any sensor can be modeled as a receptor process.

Example 5.1. A light sensor L responds to light (i.e., a detected number of photons per second) by a voltage ouput (e.g., 0.87 volts when a photoresistor is struck by 2.3 x 10^{13} photons per second) [5]. Light sensors can be used to enable mobile robots to stop its activity in the dark or to move toward a beacon. Let $R, R_L, v(t), v_{PE}(t)$ be the ohms of a resistor in the photoresistor circuit for L, resistance of L, voltage input $v(t)$ at time t in a circuit containing L, and output voltage $v_{PE}(t)$ at time t in L, respectively, where

$$v_{PE}(t) = \frac{R_L}{R + R_L} v(t)$$

To obtain a receptor process model of sensor L, replace $x(t)$ in Fig. 4 with $R, R_L, v(t)$ and π with $v_{PE}(t)$, respectively (see Fig. 6).

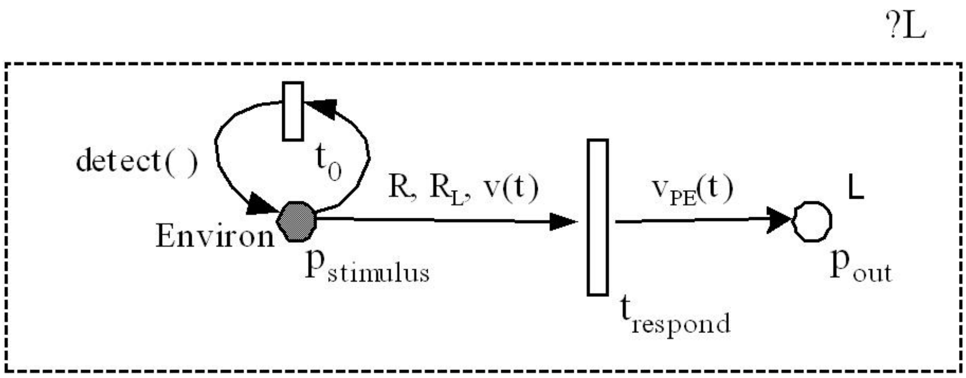

Figure 6. Petri Net Model of a Light Sensor

5.2. Filters

Filters provide a means of weeding out sensor signals which are not wanted and in modifying or manipulating sensor signals to facilitate their usage in a system. This is case in noisy communication system, for example, where electric filters are used to eliminate signal contamination.

A *filter* is a mechanism that selects readings of a sensor relative to one or more selection criteria.

In the case of an electric filter, its selection can include the modification, reshaping or manipulation of the frequency spectrum of an electric signal according to some prescribed requirements. Filters are useful, for example, in eliminating signal contamination and in separating relevant from irrelevant sensor inputs. It is in this latter sense of filter-usefulness that filters are considered in the context of rough Petri nets. Notice that a Łukasiewicz guard can be used as the basis for a model of a filter on sensor input of an approximation neuron, since there is interest in preventing input signals with approximately zero strength from enabling an input transition. As mentioned earlier, a Łukasiewicz guard can be defined over $[a, b] = [a, 1]$ where $a > 0$ relative to a set X of sample signal values. In effect, a Łukasiewicz guard can be used to model a sensor filter.

To complete the modeling of a sensor filter, a restricted Łukasiewicz guard can be introduced. Let ω_L, ω_H denote real-valued low signal cutoff and high signal cutoff, respectively. This form of guard is motivated by a need to identify what is known as a bandpass filter, where no signal is "accepted" outside an interval $[\omega_L, \omega_H]$.

Definition 5.2. *Restricted Łukasiewicz Guard.* A Łukasiewicz guard $P_{\alpha,\lambda}(X)$ on transition t with input X is called *restricted* if the condition $\alpha(z)$ is defined by $\alpha(z)$ holds iff $z \in [\omega_L, \omega_H] \subset [0, 1]$ for some $\omega_L, \omega_H \in [0, 1]$.

If we assume that the input to a Łukasiewicz guard comes from a sensor, then such a guard acts a filter. Hence, any restricted Łukasiewicz guard with sensor input is a bandpass filter.

Notice that a Łukasiewicz guard can be defined over $[\omega_L, \omega_H] = [\omega_L, 1]$ where $\omega_L > 0$ and $\omega_H = 1$. In effect, we define condition a by $\alpha(x)$ iff $\lambda(x) \in [\omega_L, 1]$, which is called a highpass band. Hence, the Łukasiewicz guard in Def. 5.2 can be used to model a highpass filter. Hence, any Łukasiewicz guard with sensor output restricted to $[\omega_L, 1]$ is a highpass filter model.

Example 5.3. Consider a large room where a uniform temperature must be achieved with a combination of sensors, sensor filters and heating elements (see Fig. 7). Ambient temperatures are a problem in controlling room temperature. It may be warm in one part of the room, and cold in another part of the room. For simplicity, a single sensor plus filter and single heating element are represented in Fig. 7. Let τ denote a temperature sensor and let $\tau(x)$ denote a sample temperature reading. Also assume that τ is connected to a filter λ, which maps the recorded temperature to $[0, 1]$ as shown in Fig. 8. A sample decision table reflecting reactions of the system to sensed temperatures is shown in Fig. 8. The decision d has a value chosen by an expert for any local temperature state. For example, $\tau(x_1)$ returns a chilly sample temperature of 1.49 C and $\lambda(1.49)$ returns a 0.9 filter value. The corresponding approximation regions with respect to λ, τ are shown in Fig. 9.

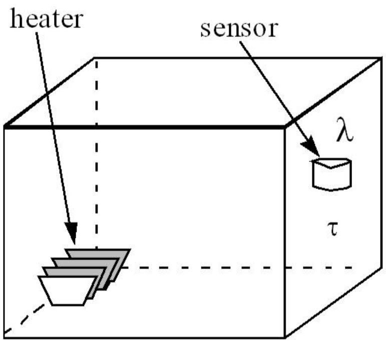

Figure 7. Heat controlled room

	τ	λ	d
x_1	1.49	0.9	heatOn
x_2	22.0	1.0	heatOff
x_3	19.0	0.98	heatOn
x_4	19.0	0.98	heatOff
x_5	18.0	0.89	heatOff
x_6	1.25	0.95	heatOn

Figure 8. Decision Table

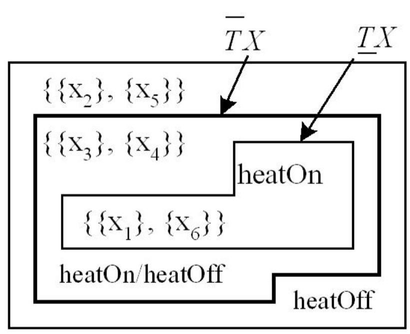

Figure 9. Approximation Regions

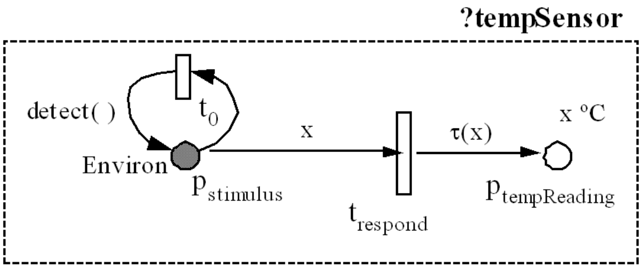

Figure 10. Temperature Sensor Model

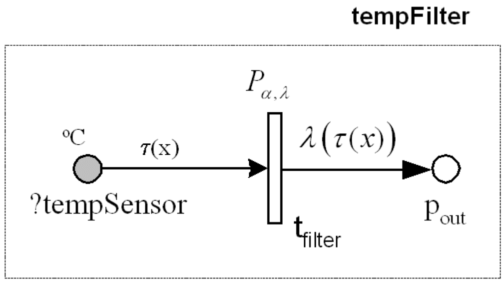

Figure 11. Filter Model

A Petri net model of the sensor-filter combination used to construct the decision table in Fig. 8 can now be given. First, a receptor process model for a receptor place denoted ?tempSensor representing a temperature sensor is given in Fig. 10. The temperature sensor is then connected to a model of the filter in form of a Łukasiewicz guard (see Fig. 11). To complete the model of the system used to produce approximation regions specified by $\overline{T}X$, where the enabling of a transition rough leads to a computation ρ that constructs an upper approximation (see Fig. 12). The model in Fig. 12 can be taken a step further to create a model of a rough neuron [25]. Let $\overline{T}X, [u]_e$ denote an input place which supplies an upper approximation, and input place which supplies a indiscernibility class, respectively. Further, assume that $\overline{T}X, [u]_e$ supply input to a transition that computes a rough membership function value as in Fig. 3(a). Such a Petri net provides a simple model of what is known as a rough neuron, which can be used in classifying room temperature readings.

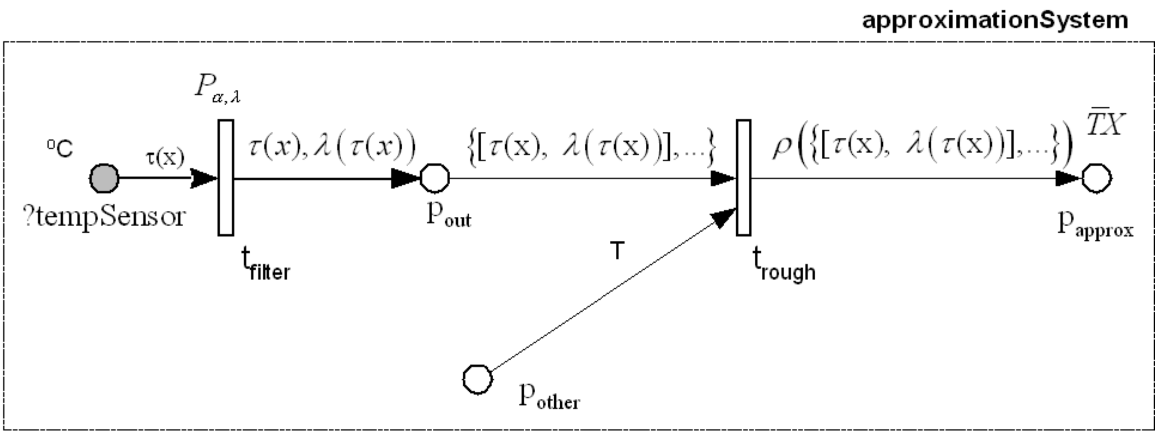

Figure 12. Approximation System Model

6. Sensor Fusion

Consider, next, the case where there is interest in discovering which sensor is more relevant among a set of sensors. The computation required to identify relevant sensors provides a form of sensor fusion. The term sensor fusion generally refers to some process of combining sensor readings [28]. The term relevance in this context denotes the ”closeness” of a set of experimental sensor values relative to a set of a pre-calibrated, target sensor values that are considered important in a classification effort. The identification of relevant sensors provides a form of sensor fusion. Further, assume that each of the sensors have the same model with essentially the same accuracy. At this stage, we will ignore the issue of the accuracy of a sensor, and trust that each sensor in the set of sensors produces output with low error.

6.1. Relevant Sensors

Consider the information system (U, A). Let $u \in U, B \subseteq A$. Further, let $B = \{a_1, \ldots, a_n\}$ be a set of homogenous sensors. Next, determine $[u]_e$, which is crucial in assessing the relevance of the sensors in B. The set $[u]_e$ denotes a partition defined by an expert decision e. This partition is needed to classify sensors using a rough integral [20]. Let $[r, s]$ be a real-valued sensor signal value range used to gauge the relevance of a sensor. Then, for example, the selection R of the most relevant sensors in a set of sensors is found using

$$R = \left\{ a_i \in B \mid \int a_i \, d\mu_{\mathrm{u}}^{\mathrm{e}} \in [s, r] \right\}$$

In effect, the integral $\int a_i \, \mu_{\mathrm{u}}^{\mathrm{e}}$ serves as a filter inasmuch as it "filters" out all sensors with integral values outside the prescribed interval.

6.2. Petri Net Model of Sensor Fusion

Consider the following scheme that utilizes a rough integral to select relevant sensor signals. We start with a receptor process ?p that models a sensor a and assume that this sensor provides input X to a guarded transition t. Let ρ be a rough measure; $F(X) = \int_X a \, d\rho$, a rough integral. The condition α in the guard $(\alpha, \lambda, P_\alpha)$ is defined by $\alpha(x)$ holds iff $x \in [s, r]$, where $\lambda(X) = F(X)/max(F(X))$ and $[s, r]$ is a pre-selected, real-valued target sensor signal value range. This scheme can be used to design a simple sensor fusion model with two sensors represented by receptors ?p1 and ?p4 (see Fig. 13).

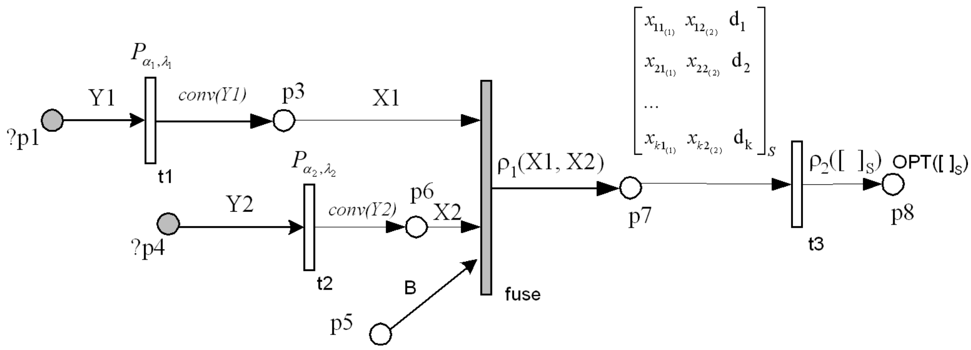

Figure 13. Simple Sensor Fusion Model

Transition t1 is enabled and results in the conversion ("conditioning" with method conv(Y1)) of signal Y1 to an appropriate form (e.g., analog to digital), if condition α_1 on $\lambda_1(\mathrm{Y1})$ is satisfied. Similarly, transition t2 is enabled, if condition α_2 on $\lambda_1(\mathrm{Y1})$ is satisfied. The function conv maps Y1 $\{\mathrm{Y2}\}$ (a sensor signal) to X1 $\{\mathrm{X2}\}$ (a more usable form of the signal). For example, the result of a "conversion" operation on signal Y1 might be the amplification X1. This is common practice in transducers where weak signals are replaced by amplified signals to facilitate signal analysis. In the case where transitions t1 and t2 are both enabled, sensor fusion occurs. In the case where the transition labeled ”fuse” fires, the method $\rho_1(X1, X2)$ derives a decision table represented by $[\,]_S$ in Fig. 13. Whenever transition t3 is enabled and fires, the method $\rho_1(X1, X2)$ implements classical rough set methods to produce a set of rules OPT($[\,]_S$).

In this simplified model of rule-production in guiding decision-making in a problem-solving system, the construction of the decision table by ρ_1 depends on the fusion of values from sensors represented by ?p1 and ?p4. Each row of this table provides an instance of an information granule useful in reasoning about perceived phenomena. he transition labeled ”fuse” becomes enabled only in the case where sensor signals Y1 and Y2 satisfy the guards on transitions t1 and t2, respectively. Once the two input signals cross the thresholds for the guards on t1 and t2, then a decision table can be built. Over time, the decision table constructed by $\rho_1(X1, X2)$ will change as the input signals change. Recall that receptor places such as ?p1 and ?p4 continuously respond to new stimuli. Signal changes ripple through this system as long as each new signal meets the required conditions specified by the guards.

7. Conclusion

Petri nets defined in the context of rough sets have provided a means of modeling sensors, filters, and fusion of sensors. The filter models described in this paper take their inspiration from classical electric filters, where low pass, pass band, and high pass filters are common. Łukasiewicz guards make it possible to model filters useful in isolating parts of a sensor signal deemed important in a problem-solving effort (e.g., classification of the movements of an agent relative to a selected spatial region). Rough measures and integrals make it possible to define particular classes of Łukasiewicz guards useful in sensor fusion. An illustration of this idea has been given in terms of a simple fusion network model that ”organizes” itself over time by selectively firing transitions activated by particular sensor signals.

Acknowledgement. The research of Sheela Ramanna and James Peters has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research grant 194376 and research grant 185986, respectively. The research of Maciej Borkowski has been supported by a grant from Manitoba Hydro. The research of Andrzej Skowron has been supported by the grants 8 T11C 025 19 from State Committee for Scientific Research (KBN) and from the Wallenberg Foundation. Zbigniew Suraj has been supported by the grant 8 T11C 025 19 from KBN and by grant STP201862 from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] P.S. Bimbhra, Inverters. In: P.S. Bimbhra, Power Electronics. Delhi, India: Khanna Publishers, 1998, pp. 309-391.

[2] R.R. Brooks, S.S.Iyengar, Methods of approximate agreement for multisensor fusion. In: Kadar and Libby (Eds.), Signal Processing, Sensor Fusion and Target Recognition IV, SPIE vol. 2484, 1995, pp. 37-44.

[3] G. Choquet, Theory of capacities. Annales de l’Institut Fourier, 5, 1953, pp. 131-295.

[4] D.L.Dill, Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits. Cambridge, MA: MIT Press, 1989.

[5] J. Fraden, Handbook of Modern Sensors: Physics, Design, and Applications. Berlin: Springer-Verlag, 1996.

[6] M. Grabisch, Alternative expressions of the discrete Choquet integral. Proc. 7th IFSA World Congress, Prague, 25-29 June 1997, pp. 472-477.

[7] P.R. Halmos, Measure Theory. London: D. Van Nostrand Co., 1950.

[8] J.D.Irwin, C.-H. Wu, Basic Engineering Circuit Analysis. 6th Ed. NY: John Wiley & Sons, Inc., 1999.

[9] K. Jensen, Coloured Petri Nets–Basic Concepts. Analysis Methods and Practical Use, vol. 1. Berlin, Springer-Verlag, 1992.

[10] D.S. Kirschen, R. Bacher, G.T. Heydt, Special issue on the technology of power system competition. Proceedings of the IEEE, Feb. 2000, vol. 88, no. 2, pp. 123-127.

[11] J. Łukasiewicz, O logice trójwartościowej. Ruch Filozoficzny 5, 1920, pp. 170-171. See On three-valued logic, English translation in L. Borkowski (Ed.), Jan Łukasiewicz: Selected Works, Amsterdam, North-Holland, 1970, pp. 87-88.

[12] J.W. Nilsson, S.A. Riedel, Electric Circuits. 6th Ed. NJ: Prentice Hall, 2000.

[13] Z. Pawlak, Rough sets, Int. J. of Computer and Information Sciences. Vol. 11, 1982, pp. 341-356.

[14] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data. Dordrecht: Kluwer Academic Publishers, 1991.

[15] Z. Pawlak, Reasoning about data–A rough set perspective. Lecture Notes in Artificial Intelligence 1424, L. Polkowski and A. Skowron (Eds.). Berlin, Springer-Verlag, 1998, pp. 25-34.

[16] Z. Pawlak, Decision rules, Baye's rule and rough sets. In: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence No. 1711. Berlin: Springer-Verlag, 1999, pp. 1-9.

[17] Z. Pawlak, Drawing conclusions from data–The rough set way. Int. Journal of Intelligent Systems, Vol. 16, no. 1, 2001, pp. 3-12.

[18] Z. Pawlak, A. Skowron, Rough membership functions. In: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, NY, John Wiley & Sons, 1994, 251-271.

[19] Z. Pawlak, On rough derivatives, rough integrals, and rough differential equations. ICS Research Report 41/95, Institute of Computer Science, Nowowiejska 15/19, 00-665 Warsaw, Poland, 1995.

[20] Z. Pawlak, J.F. Peters, A. Skowron, Z.Suraj, S. Ramanna, Rough measures: Theory and application. In: Proc. Rough Set Theory and Granular Computing (RSTGR'2001), May 2001 [to appear].

[21] J.F. Peters, Time and clock information systems: Concepts and roughly fuzzy Petri net models. In: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery. Heidelberg, Physica-Verlag, Vol. 2, 1997, pp. 385-418.

[22] J.F. Peters, A. Skowron, Z. Suraj, S. Ramanna, A. Paryzek, Modeling real-time decision-making systems with rough fuzzy Petri nets. In: Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT98), Vol. 1, Aachen, Germany, pp. 985-989.

[23] J.F. Peters, A.Skowron, Z. Suraj, S. Ramanna, Guarded transitions in rough Petri nets. In: Proc. of 7th European Congress on Intelligent Systems & Soft Computing (EUFIT'99), 13-16 Sept. 1999, Aachen, Germany, pp. 203-212.

[24] J.F.Peters, S. Skowron, Z. Suraj, W. Pedrycz, S. Ramanna, Approximate real-time decision-making: Concepts and roughly fuzzy Petri net model. International Journal of Intelligent Systems, vol. 14, no. 4, 1999, pp. 4-37.

[25] J.F.Peters, A. Skowron, Z. Suraj, L. Han, S. Ramanna, Design of rough neurons: Rough set foundation and Petri net model. In: Z.W. Ras, S. Ohsuga (Eds.), Lecture Notes in Artificial Intelligence, vol. 1932, 2000, pp. 283-291.

[26] J.F.Peters, A. Skowron, Z. Suraj, S. Ramanna, Sensor and filter models with rough Petri nets. In: H.-D. Burkhard, L. Czaja, A. Skowron, P. Starke (Eds.), Proc. Workshop on Concurrency, Specification & Programming, 9-11 Oct. 2000, Berlin, vol. 2, pp. 203-212.

[27] J.F.Peters, K. Ziaei, S. Ramanna, S., Approximate Time Rough Control: Concepts and Application to Satellite Attitude Control. In: Proc. Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98), Warsaw, Poland, Lecture Notes in Artificial Intelligence 1424, Berlin, Springer-Verlag, 1998, pp. 491-498.

[28] Petri, C.A., Kommunikation mit Automaten. Schriften des IIM Nr. 3, Institut für Instrumentelle Mathematik, Bonn, West Germany, 1962.

[29] A. Skowron, Z. Suraj. A parallel algorithm for real-time decision making: a rough set approach. Journal. of Intelligent Information Systems, Vol. 7, 1996, pp. 5-28.

[30] A. Skowron, J.F.Peters, Z. Suraj, An application of rough set methods to control design. Fundamenta Informaticae Vol. 43, nos. 1-4, 2000, pp. 269-290.

[31] Z. Suraj. Discovery of concurrent data models from experimental tables: A rough set approach. Fundamenta Informaticae, vol. 28, nos. 3-4, 1996, pp. 353-376.

[32] Z. Suraj, Rough set methods for the synthesis and analysis of concurrent processes. ICS PAS Report No. 893, December 1999, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

[33] Z. Suraj, Petri nets and rough sets in controller design. In: S. Aoshima, L. Polkowski, M. Toho (Eds.), Proc. Int. Conf. on Intelligent Techniques in Robotics, Control and Decision Making, 22-23 Feb. 1999, Warsaw, pp. 86-96.

[34] Z. Suraj, ROSEPEN: Environment for the synthesis and analysis of concurrent processes based on rough set methods and Petri nets. In: S. Tsumoto, Y.Y. Yao, M. Hadijimichael (Eds.), Bulletin of International Rough Sets Society, vol. 1, no. 1, June 1998, pp. 37-39.

[35] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. on System, Man and Cybernetics, vol. 18, 1988, pp. 183-190.