2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

On the Reachability Space and Deadlock-Freeness in Flexible Nets

Jorge Júlvez

Abstract— **Deadlock-freeness is a basic property of dynamical systems that ensures that at least one process of the system can operate indefinitely. Given that a system is deadlock-free if at any reachable state there is at least one process that can operate, the reachability space, i.e. the set of states that can be reached, and deadlock-freeness are closely related. This paper focuses on some fundamental properties of the reachability space of Flexible Nets, a modeling formalism that can easily account for uncertain parameters. After showing that the reachability space is convex, a sufficient condition for deadlock-freeness is derived.**

I. INTRODUCTION

A number of modeling formalisms have been proposed to model and analyze dynamical systems. These formalisms establish the primitives and the construction semantics to build mathematical models of real systems. Moreover, the analysis capabilities of a model is largely determined by the formalism used to produce the model.

Petri nets [1], [2] is an appealing modeling formalism that can be represented graphically and that offers a number of analysis possibilities. Petri nets have been successfully applied to very different application domains such as hardware design [3], systems biology [4], manufacturing [5], game theory [6], conformance checking [7], etc. Despite the modeling power of Petri nets, there are some common features of real systems that are difficult to model by Petri nets such as nonlinear dynamics and uncertainties in the state change produced by an event, the speed of transitions or the initial state.

Flexible nets (FNs) [8] is a relatively novel modeling formalism that aims to enhance the modeling and analysis [9] capabilities of Petri nets. In addition to places and transitions, FNs introduce *handlers* which capture the relationships between the state and the processes of the system. The association of sets of linear inequalities with handlers allows the modeler to capture the potentially uncertain relation between state and processes. In order to avoid the state explosion problem of large discrete event systems, the state of an FN is defined in the reals, nevertheless, hybrid systems can also be modeled by considering guards [10].

In the scope of FNs, this paper focuses on two basic properties: reachability and deadlock-freeness. While reachability accounts for the set of states that can be reached by the system, deadlock-freeness ensures that no reachable marking blocks the system completely, i.e. at least one process of the system can operate indefinitely. Similarly to continuous Petri nets, it is shown that the reachability space of a FN is convex. With respect to deadlock-freeness, a sufficient condition is obtained.

The rest of the paper is organized as follows: Once Section II introduces event nets, Section III develops the main results related to the reachability space. Intensity and FNs are introduced in Sections IV and V respectively. Section VI focuses on deadlock-freeness. The main conclusions are drawn in Section VII.

II. EVENT NETS

In the following, the reader is assumed to be familiar with Petri nets (see [1], [2] for a gentle introduction).

A Flexible Net (FN) is composed of an event net and an intensity net: the event net determines how the processes of the system change the state of the system, and the intensity net establishes the speed of such processes (see [8], [10] for a detailed definition an modeling examples of FNs).

Definition 1 (Event net): An event net is a tuple $\mathcal{N}_V = (P, T, V, E_V, A, B)$ where (P, T, V, E_V) is a tripartite graph determining the net structure and (A, B) are matrices determining the potential evolutions of the marking.

The vertices of the event net are P, T and V, where P is a set of $|P|$ places, T is a set of $|T|$ transitions and V is a set of $|V|$ event handlers. Places are depicted as circles and model the types of components in the system. Transitions are depicted as rectangles and model the processes of the system. Event handlers are depicted as dots and model the different ways in which the transitions can change the marking.

The vertices of the event net are connected by the edges in E_V. Each pair of vertices can be connected by at most one edge. The set E_V is partitioned into two sets E_V^P and E_V^T: The edges in E_V^P are directed and are referred as *event arcs*, every $e \in E_V^P$ is either an arc $e = (p_i, v_k)$ from a place p_i to a handler v_k, or an arc $e = (v_k, p_i)$ from a handler v_k to a place p_i. Every $e \in E_V^T$ is an edge $e = \{t_j, v_k\}$ connecting a transition t_j and a handler v_k. In a similar way to Petri nets, the following notation is used: ${}^p v_k$ denotes the input places of v_k; v_k^p denotes the output places of v_k; ${}^v p_i$ denotes the input handlers of p_i; p_i^v denotes the output handlers of p_i; ${}^t v_k$ denotes the transitions connected to v_k; and t_j^v denotes the handlers connected to t_j.

For instance, the event net in Fig. 1 has two places, $P = \{p_1, p_2\}$, one transition, $T = \{t_1\}$, and one event handler, $V = \{v_1\}$. The set of arcs of the net is $E_V^P = \{(p_1, v_1), (v_1, p_2)\}$ and the set of edges is $E_V^T = \{\{t_1, v_1\}\}$.

In an event net, each place contains a number of tokens (or marking), and each transition contains a number of actions

This work was supported by the Spanish Ministry of Science and Innovation through the projects DAMOCLES-PID2020-113969RB-I00 and TED2021-130449B-I00.

J. Júlvez is with the Department of Computer Science and Systems Engineering, University of Zaragoza, Zaragoza, Spain julvez@unizar.es

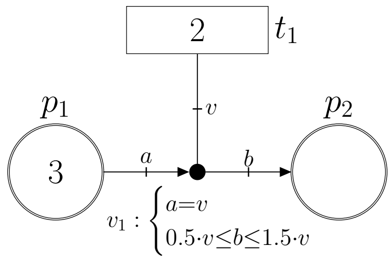

Fig. 1: Event net with two places, one transition and one event handler.

that represent the potential of the system to carry out the process that the transition models. The state of an event net accounts not only for the marking and the number of actions, but also for the marking changes and the execution of actions:

Definition 2 (State): The state of an event net $\mathcal{N}_V$ is given by the tuple $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$, where:

- $\sigma \in \mathbb{R}_{\geq 0}^{|T|}$ is a vector indexed by T where $\sigma[t_j]$ is the number of actions produced in t_j.
- $a_T \in \mathbb{R}_{\geq 0}^{|T|}$ is a vector indexed by T where $a_T[t_j]$ is the number of actions available in t_j.
- $a_E \in \mathbb{R}_{\geq 0}^{|E_V^T|}$ is a vector indexed by E_V^T where $a_E[\{t_j, v_k\}]$ is the number of actions of t_j executed by v_k.
- $\Delta m \in \mathbb{R}_{\geq 0}^{|E_V^P|}$ is a vector indexed by E_V^P where $\Delta m[(p_i, v_k)]$ is the number of tokens in p_i consumed by v_k, and $\Delta m[(v_k, p_i)]$ is the number of tokens in p_i produced by v_k.
- $m \in \mathbb{R}_{\geq 0}^{|P|}$ is the marking, i.e. a vector indexed by P where $m[p_i]$ is the number of tokens in p_i.

The state depicted in the event net in Fig. 1 is given by $\sigma[t_1] = 2$, $a_T[t_1] = 2$, $a_E[\{t_1, v_1\}] = 0$, $\Delta m[(p_1, v_1)] = 0$, $\Delta m[(v_1, p_2)] = 0$, $m[p_1] = 3$, and $m[p_2] = 0$. The interpretation of this state is the following: $\sigma[t_1] = 2$ and $a_T[t_1] = 2$ mean that two actions were produced in t_1 and both actions are available; $a_E[\{t_1, v_1\}] = 0$ means that v_1 has executed 0 actions of t_1; $\Delta m[(p_1, v_1)] = 0$ means that 0 tokens in p_1 have been consumed by v_1; $\Delta m[(v_1, p_2)] = 0$ means that 0 tokens have been produced in p_2 by v_1; and $m[p_1] = 3$ and $m[p_2] = 0$ mean that the number of tokens in p_1 and p_2 is 3 and 0 respectively.

Each event handler $v_k \in V$ is associated with a set of linear inequalities that relate the number of actions executed in the connected transitions to the marking changes in the connected places. For instance, the labels a, b and v associated with the arcs and the edge of the net in Fig. 1 and the inequalities associated with v_1 mean that each action of t_1 executed by v_1 consumes one token from p_1 and produces in p_2 a nondeterministic amount of tokens in the interval $[0.5, 1.5]$.

The coefficients of the inequalities associated with an event handler v_k can be expressed by two matrices (A_k, B_k) of real numbers that have the same number of rows. The number of actions, $a_f \in \mathbb{R}_{\geq 0}^{|{}^{t}v_k|}$, executed by v_k and the produced marking changes, $\Delta m_f \in \mathbb{R}_{\geq 0}^{|{}^{p}v_k|+|v_k^p|}$, is given by $A_k \Delta m_f {\leq} B_k a_f$. The columns of A_k are indexed by the arcs connecting v_k to places. The columns of B_k are indexed by the edges connecting transitions to v_k. For example, the matrices A_1 and B_1 associated with the inequalities of v_1 in Fig. 1 are:

$$A_1 {=} \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix}; B_1 {=} \begin{pmatrix} 1 \\ -1 \\ -0.5 \\ 1.5 \end{pmatrix}$$

where the indices of the columns of A_1 are ordered as (p_1, v_1), (v_1, p_2), and the index of the column of B_1 is $\{t_1, v_1\}$. Thus, if the number of actions executed by v_1 is 1, i.e. $a_f[\{t_1, v_1\}] = 1$, then $A_1 \Delta m_f {\leq} B_1 a_f$ implies that $\Delta m_f[(p_1, v_1)] = 1$ and $0.5 \leq \Delta m_f[(v_1, p_2)] \leq 1.5$.

Matrices A and B are obtained by arranging all the matrices A_k and B_k *diagonally*. For the sake of simplicity, the inequalities of handlers will be omitted if all the labels are equal, e.g. the omission of the inequalities of v_1 in Fig. 1 would imply $a = b = v$.

Notice that the number of actions available in the transitions, a_T, is equal to the number of actions that were produced, σ, minus the number of actions, a_E, that have been executed by the connected event handlers. Hence, for every $t_j \in T$ it holds that:

$$a_T[t_j] = \sigma[t_j] - \sum_{v_k \in t_j^v} a_E[\{t_j, v_k\}] \tag{1}$$

Similarly, the number of tokens in a place p_i is equal to the initial number of tokens, which is denoted $m_0[p_i]$, minus the number of tokens consumed plus the number of tokens produced by the connected event handlers. Hence, for every $p_i \in P$ it holds that:

$$m[p_i] {=} m_0[p_i] {-} \sum_{v_k \in p_i^v} \Delta m[(p_i, v_k)] {+} \sum_{v_k \in {}^{v}p_i} \Delta m[(v_k, p_i)] \tag{2}$$

As an example, if one action is executed by v_1 in Fig. 1, then one of the potential states that can be reached (remember that a nondeterministic amount of tokens is produced) is $\sigma[t_1] = 2$, $a_T[t_1] = 1$, $a_E[\{t_1, v_1\}] = 1$, $\Delta m[(p_1, v_1)] = 1$, $\Delta m[(v_1, p_2)] = 1.2$, $m[p_1] = 2$, and $m[p_2] = 1.2$.

III. REACHABILITY

This section first introduces some concepts related to the enabling and firing of event handlers, and then shows that the reachability space of an event net is convex.

Definition 3 (Enabling): Event handler v_k is enabled at $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ if a vector $a_f \in \mathbb{R}_{\geq 0}^{|{}^{t}v_k|}$ indexed by the edges of v_k, and a vector $\Delta m_f \in \mathbb{R}_{\geq 0}^{|{}^{p}v_k|+|v_k^p|}$ indexed by the arcs of v_k exist such that:

$$a_f[\{t_j, v_k\}] \leq a_T[t_j] \quad \forall\ t_j \in {}^{t}v_k \tag{3}$$

$$A_k \Delta m_f \leq B_k a_f \tag{4}$$

$$\Delta m_f[(p_i, v_k)] \leq m[p_i] \quad \forall\ p_i \in {}^{p}v_k \tag{5}$$

$$\mathbf{1} a_f + \mathbf{1} \Delta m_f > 0 \tag{6}$$

Inequality (3) guarantees that enough actions are available, (4) makes use of the matrices A_k and B_k (as discussed in Secion II) to relate the number of executed actions to the marking changes, (5) guarantees that enough tokens are available to be consumed in the input places, and (6) guarantees that the overall state change is not null. Notice that the inequalities (4) allow the modeling of uncertainty in the marking changes produced by the execution of actions.

Definition 4 (Firing): An event handler $v_k \in V$ enabled at $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ can fire. The firing of v_k leads instantaneously to a new state $\mathbf{x}' = (\sigma, a'_T, a'_E, \Delta m', m')$ where only the variables associated with edges, arcs, places and transitions connected to v_k are updated as follows:

$$\begin{aligned}
a'_T[t_j] &= a_T[t_j] - a_f[\{t_j, v_k\}] && \forall\ t_j \in {}^t v_k \\
a'_E[\{t_j, v_k\}] &= a_E[\{t_j, v_k\}] + a_f[\{t_j, v_k\}] && \forall\ t_j \in {}^t v_k \\
\Delta m'[(p_i, v_k)] &= \Delta m[(p_i, v_k)] + \Delta m_f[(p_i, v_k)] && \forall\ p_i \in {}^p v_k \\
\Delta m'[(v_k, p_i)] &= \Delta m[(v_k, p_i)] + \Delta m_f[(v_k, p_i)] && \forall\ p_i \in v_k^p \\
m'[p_i] &= m[p_i] - \Delta m_f[(p_i, v_k)] && \forall\ p_i \in {}^p v_k \\
m'[p_i] &= m[p_i] + \Delta m_f[(v_k, p_i)] && \forall\ p_i \in v_k^p
\end{aligned}$$

where a_f and Δm_f satisfy (3), (4), (5) and (6).

The fact that $\mathbf{x}'$ is reached from $\mathbf{x}$ after the firing of v_k with a_f and Δm_f is denoted as:

$$\mathbf{x} \xrightarrow{(a_f, \Delta m_f, v_k)} \mathbf{x}'$$

An event handler is said to be well-defined when it allows the flow of tokens and the execution of actions in all the connected places and transitions,

Definition 5 (Well-defined): An event handler $v_k \in V$ is well-defined if there exist $\Delta m_f \geq \mathbf{1}$ and $a_f \geq \mathbf{1}$ such that $A_k \Delta m_f \leq B_k a_f$

If an event handler is not well-defined, it will avoid the flow of tokens or the execution of actions. Given that such a behavior is not useful from a modeling point of view, in the following, it will be assumed that all the event handlers are well-defined. Notice, however, that the firing of an event handler does not necessarily imply consumption and production of tokens and execution of actions in all its connected places and transitions.

Definition 6 (Reachable state): Let the state of an event net $\mathcal{N}_V$ be $\mathbf{x_0} = (\sigma, \sigma, 0, 0, m_0)$, i.e. σ actions are available and no event handler has fired. A state $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ is reachable from $\mathbf{x_0}$ if a finite sequence $(\alpha_1, \beta_1, v_{a_1}), (\alpha_2, \beta_2, v_{a_2}), \ldots, (\alpha_k, \beta_k, v_{a_k})$ exists such that:

$$\mathbf{x_0} \xrightarrow{(\alpha_1, \beta_1, v_{a_1})} \mathbf{x_1} \xrightarrow{(\alpha_2, \beta_2, v_{a_2})} \mathbf{x_2} \ldots \xrightarrow{(\alpha_k, \beta_k, v_{a_k})} \mathbf{x}$$

The reachability space, denoted as $RS_{\mathcal{N}_V}(\mathbf{x_0})$, is the set of all reachable states from $\mathbf{x_0}$. For the sake of clarity, a firing sequence $(\alpha_1, \beta_1, v_{a_1}), (\alpha_2, \beta_2, v_{a_2}) \ldots, (\alpha_k, \beta_k, v_{a_k})$ can be abbreviated as q, and $\gamma \cdot q$ with $\gamma \in \mathbb{R}_{>0}$ denotes the same firing sequence where each term (α_i, β_i) is multiplied by γ.

Given that the firing of a handler implies a linear transformation of the state, the following property holds:

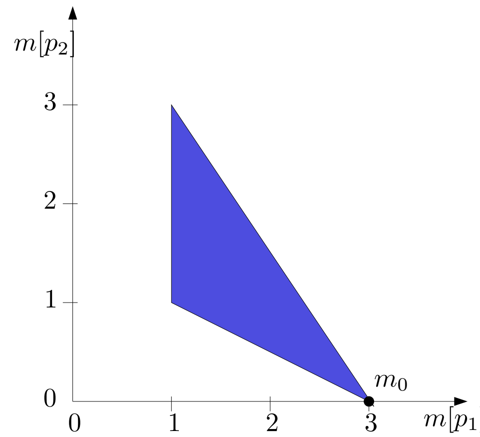

Fig. 2: Reachable markings of the event net in Fig. 1.

Lemma 1: Let $\mathcal{N}_V$ be an event net and $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ be a state reachable from $\mathbf{x_0} = (\sigma, \sigma, 0, 0, m_0)$ through the sequence $q = \{(\alpha_1, \beta_1, v_{a_1}), (\alpha_2, \beta_2, v_{a_2}), \ldots, (\alpha_k, \beta_k, v_{a_k})\}$, i.e.

$$\mathbf{x_0} \xrightarrow{(\alpha_1, \beta_1, v_{a_1})} \mathbf{x_1} \xrightarrow{(\alpha_2, \beta_2, v_{a_2})} \mathbf{x_2} \ldots \xrightarrow{(\alpha_k, \beta_k, v_{a_k})} \mathbf{x}$$

Then, for every $\gamma \in \mathbb{R}_{>0}$, the state $\gamma{\cdot}\mathbf{x} = (\gamma{\cdot}\sigma, \gamma{\cdot}a_T, \gamma{\cdot}a_E, \gamma{\cdot}\Delta m, \gamma{\cdot}m)$ is reachable from $\gamma{\cdot}\mathbf{x_0} = (\gamma{\cdot}\sigma, \gamma{\cdot}\sigma, 0, 0, \gamma{\cdot}m_0)$ through the sequence $\gamma{\cdot}q = \{(\gamma{\cdot}\alpha_1, \gamma{\cdot}\beta_1, v_{a_1}), (\gamma{\cdot}\alpha_2, \gamma{\cdot}\beta_2, v_{a_2}), \ldots, (\gamma{\cdot}\alpha_k, \gamma{\cdot}\beta_k, v_{a_k})\}$.

Similarly to continuous Petri nets [11], this property implies that the reachability space of an event net is a convex set.

Proposition 2: The reachability space, $RS_{\mathcal{N}_V}(\mathbf{x_0})$, is a convex set.

Proof: Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two states that can be reached by the sequences q_1 and q_2 respectively, i.e. $\mathbf{x_0} \xrightarrow{q_1} \mathbf{x}_1$ and $\mathbf{x_0} \xrightarrow{q_2} \mathbf{x}_2$. Let $\gamma \in (0, 1)$, then, by Lemma 1, the state $\gamma \cdot \mathbf{x}_1$ can be reached from $\gamma \cdot \mathbf{x_0}$ by the sequence $\gamma \cdot q_1$, i.e. $\gamma \cdot \mathbf{x_0} \xrightarrow{\gamma \cdot q_1} \gamma \cdot \mathbf{x}_1$, and the state $(1-\gamma) \cdot \mathbf{x}_2$ can be reached from $(1-\gamma) \cdot \mathbf{x_0}$ by the sequence $(1-\gamma) \cdot q_2$, i.e. $(1-\gamma) \cdot \mathbf{x_0} \xrightarrow{(1-\gamma) \cdot q_2} (1-\gamma) \cdot \mathbf{x}_2$. Hence, $\gamma{\cdot}\mathbf{x}_1 + (1-\gamma){\cdot}\mathbf{x}_2$ can be reached from $\mathbf{x_0}$ by firing the sequences $\gamma{\cdot}q_1$ and $(1-\gamma){\cdot}q_2$. ■

The triangle in Fig. 2 represents the convex reachability space of the markings of the net in Fig. 1 with initial state $\mathbf{x_0} = (\sigma, \sigma, 0, 0, m_0)$ where $\sigma[t_1] = 2$, $m_0[p_1] = 3$, $m_0[p_2] = 0$.

The overall change in the state produced by several firings is the result of adding the changes produced by each firing. This fact allows us to write a set of equations that must be satisfied by all the states reachable from the initial state.

Proposition 3 (State equations): Let the state of an event net $\mathcal{N}_V$ be $\mathbf{x_0} = (\sigma, \sigma, 0, 0, m_0)$. Every state $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ reachable from $\mathbf{x_0}$ belongs to $SE_{\mathcal{N}_V}(\mathbf{x_0})$ where:

$$\begin{aligned}
SE_{\mathcal{N}_V}(\mathbf{x_0}) = \{\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m) |& \\
\sigma &= a_T + Y_\sigma a_E \\
A \Delta m &\leq B a_E \\
m &= m_0 + Z_m \Delta m\}
\end{aligned} \tag{7}$$

where Y_σ and Z_m are determined by the net structure:

- Y_σ is a matrix with rows indexed by T, columns indexed by E_V^T, and such that $Y_\sigma[t_j, \{t_j, v_k\}]{=}1 \ \forall \ \{t_j, v_k\} \in E_V^T$ and the rest of the elements in Y_σ are 0,
- Z_m is a matrix with rows indexed by P, columns indexed by E_V^P, and such that $Z_m[p_i, (p_i, v_k)]{=}{-1} \quad \forall \quad (p_i, v_k) \quad \in \quad E_V^P$, $Z_m[p_i, (v_k, p_i)]{=}1 \ \forall \ (v_k, p_i) \ \in \ E_V^P$ and the rest of the elements in Z_m are 0,

and a_T, a_E, Δm and m are nonnegative variables.

Roughly speaking, the role of matrix Y_σ is to distribute the actions in transitions among the handlers connected to them, see (1). The role of Z_m is to collect and add the marking changes produced by the firings, see (2).

Proposition 3 states that $RS_{\mathcal{N}_V}(\mathbf{x_0}) \subseteq ES_{\mathcal{N}_V}(\mathbf{x_0})$. In other words, the state equations (7) represent necessary reachability conditions. This means that, as in Petri nets, equations (7) can contain spurious solutions, i.e. states that satisfy the equations but are not reachable (see Fig. 7 for an example of a spurious solution).

IV. INTENSITY NETS

The vector $\sigma \in \mathbb{R}_{\geq 0}^{|T|}$ in the state $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ of an event net denotes the number of actions produced in the transitions. Such actions are produced by the intensity net as time elapses. This section introduces the basic concepts of intensity nets.

Definition 7 (Intensity net): An intensity net is a tuple $\mathcal{N}_S = (P, T, S, E_S, C, D)$ where (P, T, S, E_S) is a tripartite graph determining the net structure and (C, D) are matrices determining the potential intensities produced by the marking.

The vertices of an intensity net are P, T and S, where P is a set of $|P|$ places, T is a set of $|T|$ transitions and S is a set of $|S|$ intensity handlers. Places and transitions model the same system features as in the event net. The intensity handlers are depicted as dots and model the different ways in which the tokens can generate intensities, or speeds, in the transitions. The vertices of the intensity net are connected by the edges in E_S. Matrices C and D have similar roles to A and B in the event net (see [8] for a detailed description).

Every transition t_j has a nonnegative real intensity, or speed, $\lambda[t_j] \in \mathbb{R}_{\geq 0}$. The integral of $\lambda[t_j]$ over time is equal to $\sigma[t_j]$, i.e. the number of actions produced in t_j.

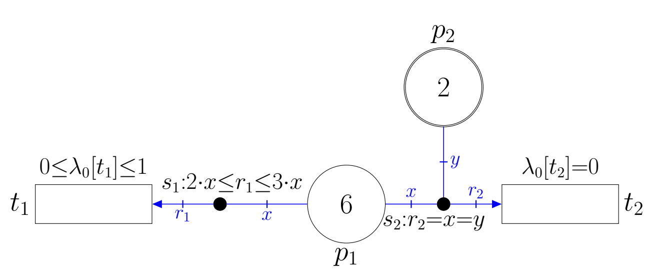

Fig. 3: Intensity net with two places, two transitions and two intensity handlers.

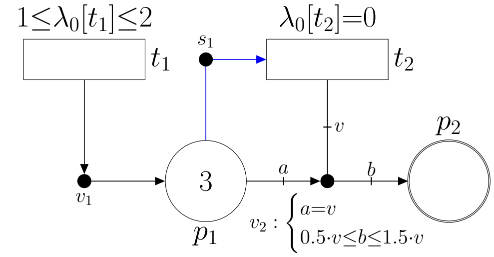

Fig. 4: FN in which the default intensity of t_1 is uncertain and the firing of v_2 produces an uncertain number of tokens in p_2.

The intensity of a transition t_j is equal to its default intensity, $\lambda_0[t_j]$, plus the intensity of its incoming arcs minus the intensity of its outgoing arcs. The intensity in the arcs is produced by the tokens in places connected to the intensity handlers. A token is *active* if it is being used by an intensity handler, otherwise it is *idle*. While idle tokens are associated with places, active tokens are associated with edges. Places whose tokens are forced to be active are drawn as single circles, places whose tokens can be idle are drawn as double circles. Each intensity handler determines how much intensity is produced in its arcs as a function of the number of active tokens in its edges.

For instance, in the intensity net in Fig.3, all the tokens in p_1 (depicted as a single circle) are forced to be active, and the tokens in p_2 (double circle) are not forced to be active. Hence, each token in p_1 must be used either by s_1 or s_2. Each token used by s_1 produces between 2 and 3 units of intensity (see inequality associated with s_1) which is added to the default intensity $\lambda_0[t_1]$ of t_1. Each token of p_1 used by s_2 synchronizes with a token in p_2 to produce one unit of intensity in t_2. As in event nets, if the inequalities of an intensity handler are omitted, it is assumed that all the labels of its arcs and edges are equal.

V. FLEXIBLE NETS

An FN is composed of an event net and an intensity net that have the same set of places and the same set of transition [8]. While the intensity net produces actions in the transitions, the event net makes use of the produced actions to carry out changes in the marking.

Given that uncertain system parameters can be modeled by the inequalities associated with the initial marking, the default intensities, and the handlers, any time trajectory that satisfies the constraints imposed by such inequalities is possible. In the FN in Fig. 4, the default intensity of t_1 is uncertain and constrained to the interval $[1, 2]$, the number of tokens produced in p_2 by each action executed by v_2 is also uncertain and constrained to $[0.5, 1.5]$. Any trajectory that satisfies these constraints is possible. Figure 5 shows the upper and lower bounds of all the potential trajectories of the marking.

Notice that the intensity net of an FN constrains the states that are reachable by the associated event net by producing particular amounts of actions. Thus, the state equations (7)

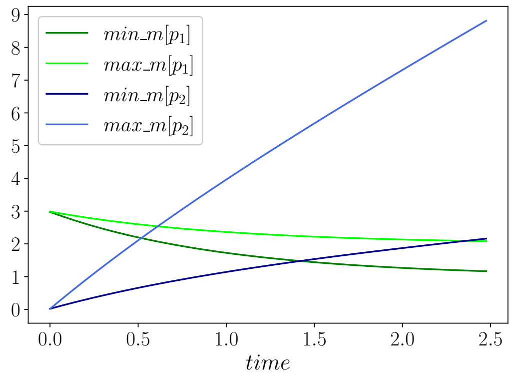

Fig. 5: Upper and lower bounds of the potential time evolutions of the markings of the FN in Fig. 4.

can be used to establish necessary conditions for reachability of the event net by ignoring the actual amounts of actions produced by the intensity net.

Proposition 4: Let $\mathcal{N}$ be an FN that contains an event net $\mathcal{N}_V$ with initial state $\mathbf{x_0} = (0, 0, 0, 0, m_0)$. Every state $\mathbf{x} = (\sigma, a_T, a_E, \Delta m, m)$ of $\mathcal{N}_V$ reachable from $\mathbf{x_0}$ over time satisfies:

$$\begin{aligned} &\sigma = a_T + Y_\sigma a_E \\ &A\Delta m \leq B a_E \\ &m = m_0 + Z_m \Delta m \end{aligned} \tag{8}$$

VI. DECIDING ON DEADLOCK-FREENESS

Deadlock-freeness is a basic behavioral property that guarantees that at least part of the system can operate indefinitely. In other words, an FN is said to be deadllock-free if for every state reachable at a any time, there exists an event handler that can fire.

In order to derive a sufficient condition for deadlock-freeness, let us first define some preliminary concepts.

A set of input places of an event handler, v_k, is an enabling set if the event handler can be enabled by the tokens in those places. More formally, the enabling sets of v_k are defined as:

Definition 8 (Enabling sets of v_k): $ES(v_k) = \{u | u \subseteq {}^p v_k$ and there exist $\Delta m_f \in \mathbb{R}_{\geq 0}^{|{}^p v_k| + |v_k^p|}$ and $a_f \in \mathbb{R}_{\geq 0}^{|{}^t v_k|}$

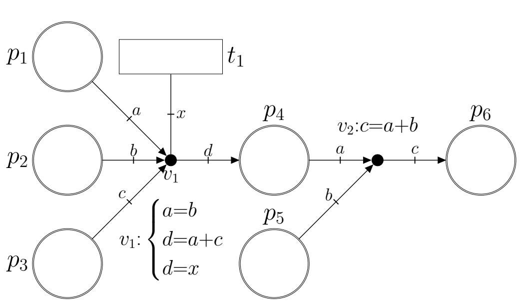

Fig. 6: Event net in which both event handlers, v_1 and v_2, have three enabling sets.

such that:

$$A_k \Delta m_f \leq B_k a_f \tag{9}$$

$$\Delta m_f[(p_i, v_k)] > 0 \quad \forall\ p_i \in u \tag{10}$$

$$\Delta m_f[(p_i, v_k)] = 0 \quad \forall\ p_i \in {}^p v_k \setminus u \tag{11}$$

$\}$

As an example, the enabling sets of the event handler v_1 in Fig. 6 are: $ES(v_1) = \{\{p_1, p_2\}, \{p_3\}, \{p_1, p_2, p_3\}\}$, and the enabling sets of v_2 are: $ES(v_2) = \{\{p_4\}, \{p_5\}, \{p_4, p_5\}\}$.

Given that in an FN, actions can be produced and executed simultaneously, an event handler can have a positive consumption and production of tokens while there are no available actions at the connected transitions. Thus, to check whether a transition can provide actions to an event handler in an FN, it is more convenient to focus on its intensity rather than on its available actions. For simplicity, in the following it will be assumed that if the marking of all the places of an enabling set $u \in ES(v_k)$ is positive, then the intensity of the transitions connected to v_k is also positive. This is a mild assumption, since the input places of an event handler are usually part of the intensity net that produces intensity in the connected transitions, see Figs. 4 and 7. This way, enabling sets can be used to determine whether an event handler in an FN can operate at a given state.

An enabling set is said to be minimum if it does not contain any other enabling set:

Definition 9 (Minimum enabling sets of v_k): $MES(v_k) = \{u | u \in ES(v_k)$ and $\nexists\ w \in ES(v_k)$ such that $w \subset u\}$

The minimum enabling sets of v_1 in Fig. 6 are: $MES(v_1) = \{\{p_1, p_2\}, \{p_3\}\}$, and the minimum enabling sets of v_2 are: $MES(v_2) = \{\{p_4\}, \{p_5\}\}$.

For each $p_i \in P$, let us define a binary variable $\delta_{pi} \in \{0, 1\}$ such that:

$$\delta_{pi} = 0 \rightarrow m[p_i] = 0 \tag{12}$$

Notice that (12) can be expressed algebraically as:

$$m[p_i] \leq \delta_{pi} \cdot M \tag{13}$$

where M is any upper bound of $m[p_i]$. If the the markings of all the places are bounded by (8), then a value M that upper bounds the marking of any place can be computed efficiently by the following linear programming problem:

$$\max \sum_{p_i \in P} m[p_i] \quad \text{subject to}$$
$$\begin{aligned} &\sigma = a_T + Y_\sigma a_E \\ &A\Delta m \leq B a_E \\ &m = m_0 + Z_m \Delta m \end{aligned}$$

Let us now define a binary variable, $\delta_{ku} \in \{0, 1\}$, for each $v_k \in V$ and each $u \in MES(v_k)$, such that $\delta_{ku} = 0$ implies that at least one place in u is empty:

$$\delta_{ku} = 0 \rightarrow \sum_{p_i \in u} \delta_{pi} < |u| \tag{14}$$

where $|u|$ denotes the number of places in u. Clearly, if $\delta_{ku}=0$, then the marking of the places in u do not enable v_k.

The implication (14) can be expressed algebraically as:

$$\sum_{p_i \in u} \delta_{pi} < |u| \cdot (\delta_{ku}+1) \tag{15}$$

The defined binary variables together with the necessary reachability conditions in (8) allow us to derive a sufficient condition for deadlock-freeness:

Theorem 5: Let the state of an event net $\mathcal{N}_V$ be $\mathbf{x_0} = (0,0,0,0,m_0)$ and let $M \in \mathbb{R}$ be un upper bound for all the markings. If the solution of the following programming problem (where the variables δ_{pi} and δ_{ku} are binary, and the rest of variables are nonnegative reals):

$$\begin{aligned} \min \sum_{v_k \in V} \sum_{u \in MES(v_k)} \delta_{ku} \quad & \text{subject to} \\ \sigma = a_T + Y_\sigma a_E & \\ A\Delta m \leq B a_E & \\ m = m_0 + Z_m \Delta m & \\ m[p_i] \leq \delta_{pi} \cdot M \quad \forall\, p_i \in P & \\ \sum_{p_i \in u} \delta_{p_i} < |u| \cdot (\delta_{ku}+1) \quad \forall v_k \in V, \forall u \in MES(v_k) & \end{aligned} \tag{16}$$

is greater than 0, then any FN with $\mathcal{N}_V$ as event net is deadlock-free.

Theorem 5 states that if at any marking that satisfies the necessary reachability conditions in (8), there is at least one event handler that can fire, then the net is deadlock-freee. Notice that Theorem 5 can be applied to any event net, i.e. it is not constrained to particular net subclasses. On the other hand, the problem (16) is a mixed integer linear programming problem and, hence, no method with polynomial complexity is known to solve it. This is, however, an expected cost because FNs are an extension of Petri nets, and the deadlock problem is PSPACE-complete in Petri nets [12].

The FN in Fig. 7 shows that the condition in Theorem 5 is not necessary for deadlock-freeness. Notice that a state with the marking $m[p_1]$=6, $m[p_3]$=4, and $m[p_2]$=$m[p_4]$=$m[p_5]$ $=0$ is a deadlock that satisfies the

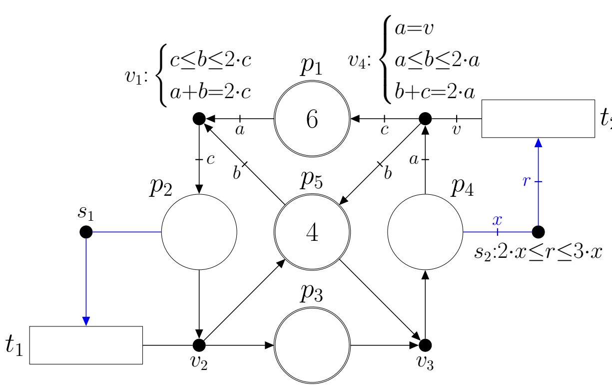

Fig. 7: FN with a spurious deadlock.

equations in (8) with $\Delta m[(p_5,v_1)]$=8, $\Delta m[(v_1,p_2)]=4$, $\Delta m[(p_2,v_2)]=4$, $\Delta m[(v_2,p_5)]=4$, $\Delta m[(v_2,p_3)]=4$, and the rest of elements in Δm equal to 0, that is not reachable, i.e., such a state is a spurious deadlock. In fact, given that the firing of v_2 consumes tokens from p_2 and produces tokens in p_5, these two places cannot be emptied simultaneously with a finite firing sequence. Moreover, the fact that the speed at which tokens are consumed from p_2 is proportional to its marking, see intensity handler s_1, makes it impossible to empty p_2 in a finite amount of time.

VII. CONCLUSIONS

Flexible Nets, a modeling formalism inspired by Petri nets, are composed of an event net and an intensity net that capture the relationships between the marking and the speed of the processes. A number of uncertain parameters can be accommodated in a Flexible Net by associating sets of inequalities with the default intensities, initial markings and handlers.

It has been shown that the reachability space of an event net, i.e. the set of states that can be reached is a convex set. Given that the intensity net just affects the speeds of the processes, the state equations of the event net represent necessary reachability conditions for any Flexible Net that contains such an event net. These state equations, together with some additional concepts, have been exploited to derive a sufficient condition for deadlock-freeness.

REFERENCES

[1] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Procs. of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
[2] M. Silva, "Introducing Petri Nets," *Practice of Petri Nets in Manufacturing*, pp. 1–62, Chapman & Hall, London, 1993.
[3] J. Cortadella, A. Yakovlev, and G. Rozenberg, eds., *Concurrency and Hardware Design, Advances in Petri Nets*, vol. 2549 of *Lecture Notes in Computer Science*, Springer, 2002.
[4] I. Koch, W. Reisig, and F. Schreiber, *Modeling in Systems Biology. The Petri Net Approach*. Springer-Verlag London, 2011.
[5] J. M. Mendes, P. Leitão, A. W. Colombo, and F. Restivo, "High-level Petri nets for the process description and control in service-oriented manufacturing systems," *International Journal of Production Research*, vol. 50, no. 6, pp. 1650–1665, 2012.
[6] J. Clempner, "Modeling shortest path games with Petri nets: A Lyapunov based theory," *Int. J. Appl. Math. Comput. Sci*, vol. 16, pp. 387–397, 01 2006.
[7] J. Carmona, B. F. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking - Relating Processes and Models*. Springer, 2018.
[8] J. Júlvez, D. Dikicioglu, and S. G. Oliver, "Handling variability and incompleteness of biological data by flexible nets: a case study for Wilson disease," *npj Systems Biology and Applications*, vol. 4, p. 7, 1 2018.
[9] J. Júlvez and S. G. Oliver, "Extending the Modeling and Analysis Capabilities of Continuous Petri nets by Flexible Nets," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1750–1756, 2021.
[10] J. Júlvez and S. G. Oliver, "Modeling, analyzing and controlling hybrid systems by Guarded Flexible Nets," *Nonlinear Analysis: Hybrid Systems*, vol. 32, pp. 131–146, 2019.
[11] L. Recalde, E. Teruel, and M. Silva, "Autonomous continuous P/T systems," in *Application and Theory of Petri Nets 1999* (J. K. S. Donatelli, ed.), vol. 1639 of *Lecture Notes in Computer Science*, pp. 107–126, Springer, 1999.
[12] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Information Sciences*, vol. 363, pp. 190–197, 2016.