

Article

# The Refinement of Petri Net with Inhibitor Arcs Based Representation for Embedded Systems

Chuanliang Xia \*, Zhuangzhuang Wang and Zhong Wang

School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China; w18596360835@outlook.com (Z.W.); wangzhong@sdjzu.edu.cn (Z.W.)

\* Correspondence: chuanliang\_xia@sdjzu.edu.cn

**Abstract:** Embedded systems are widely used in various devices. PRES+ (Petri net-based Representation for Embedded Systems) has been used to model and analyze embedded systems. However, it cannot characterize the priority of events, and cannot fully express the complex data flow and control flow. To solve this problem, inhibitor arcs are added to PRES+ and PIRES+ (PRES+ with Inhibitor arcs) is obtained. However, PIRES+'s state space explosion problem is a handicap when modeling, verifying, and controlling complex, large embedded systems. To mitigate the state space explosion problem of PIRES+ and analyze complex embedded systems, we propose the place refinement approach and the place set refinement approach for PIRES+. Under specific conditions, several important properties of PIRES+, such as timing, functionality, reachability, liveness, and boundedness, are preserved by using these refinement approaches. In order to illustrate the effectiveness of these refinement methods, as an example, the modeling and analysis of a network communication system is proposed. The refinement methods proposed have certain feasibility and practicability and provide a more practical theoretical basis for the modeling of some embedded systems.

**Keywords:** Petri nets; refinement; modeling; reachability; embedded system design



**Citation:** Xia, C.; Wang, Z.; Wang, Z. The Refinement of Petri Net with Inhibitor Arcs Based Representation for Embedded Systems. *Electronics* **2022**, *11*, 1389. <https://doi.org/10.3390/electronics11091389>

Academic Editors: Nikos Petrellis, Lambros T. Doulos and George K. Adam

Received: 29 March 2022

Accepted: 22 April 2022

Published: 26 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Embedded systems are used in diverse contexts from automobiles to communication systems, home appliances, and the Internet of Things, and are becoming more and more complex with increasing requirements. Therefore, higher requirements are put forward for the reliability, correctness, and timing of the embedded system, so it is important to focus on both how to model embedded systems and how to guarantee their correctness.

In the literature, there exist many models to represent embedded systems [1,2], such as finite state machines [3], data flow graphs [4], Petri nets [5], and communicating processes [6], etc. These formal modeling methods describe the characteristics of embedded systems from different aspects, but do not form a unified standard. Petri net has a formal mathematical definition and standardized derivation rules, which is a more standardized formal modeling method.

Ordinary Petri nets can describe systems with features such as concurrency, conflict, and uncertainty. However, there are several drawbacks when it comes to modeling embedded systems: ordinary Petri nets have no hierarchical structure, lack the notion of time, and have a limited ability to describe data flows. In order to overcome these drawbacks, several extended Petri nets for an embedded system's modeling have been proposed, such as colored Petri net [7], logical delay Petri net, time Petri net [8], hybrid Petri net, fuzzy Petri net, PRES+ (Petri net-based Representation for Embedded Systems) [9], etc. These extended forms of Petri nets start from different application requirements and improve the ability to model and analyze embedded systems.

Among them, PRES+ is an extended Petri net, and has previously been presented to represent embedded systems. Based on PRES+, Cortés et al. [9] proposed a computation

model for embedded systems and gave a method to solve the formal verification problem of the PRES+ model. Karlsson et al. [10] proposed a verification approach for the PRES+ by using a divide-and-conquer approach. In order to enhance the verification effectiveness of PRES+, Xia [11] presented transition refinement method. A sharing synthesis approach for PRES+ is presented in [12]. Under certain constraints, the synthesis PRES+ will preserve boundedness and liveness. A more formal successor marking definition of PRES+ is proposed in [13]. In order to analyze complex embedded systems, Xia et al. [14] proposed two kinds of transition synthesis methods for PRES+.

Although PRES+ can represent the embedded system at different levels, capture the real-time information of the embedded system, and describe the uncertainty of the system, the PRES+ model cannot describe the priority of events, and cannot fully express the complex data flow and control flow.

Adding inhibitor arcs to Petri nets can improve the analysis ability of Petri nets [15]. The Petri net with inhibitor arcs not only has the ability of zero detection, but also can simulate any random access machine. Inhibitor arcs can be used to control the sequence of transitions and describe the priority relationship between events. In embedded systems, some transitions occur successively under certain conditions, which affect the expression of data flow and control flow. Therefore, it is necessary to introduce inhibitor arcs. In order to improve the modeling and analysis ability of PRES+, we add inhibitor arcs into PRES+, and obtain PIREs+ (Petri net with Inhibitor arcs-based Representation for Embedded Systems).

However, when large complex embedded systems are modeled by PIREs+, the state space explosion problem is a major disadvantage for PIREs+. In the literature, many authors have investigated Petri net transformation methods to mitigate the problem of state space explosion. The three popular transformations of Petri net are refinement [11,16–29], reduction [30–33], and synthesis [12,14,34–39].

Petri net-based refinement is an important method to system design and verification. As for refinement, Huang et al. [16] presented several refinement conditions to preserve 19 properties for Place/Transition Petri nets. Padberg et al. [17] provided a thorough survey of refinements based on several nets, namely place/transition nets, predicate/transition nets, and elementary nets. For colored Petri nets (CPNs), Lakos et al. [18] gave three types of refinement: node refinement, type refinement, and subnet refinement. Mejía et al. [19] presented a refinement process for extended attribute time place Petri nets. For workflow nets, Hee et al. [20] proposed several refinement rules which preserve soundness under certain conditions. Li et al. [21] defined algebraic steps for refinement and reduction, and investigated the Petri net property preservation. Xia [22] proposed a place type refinement method for ordinary Petri nets. Under certain conditions, liveness, boundedness, and reversibility of these refined nets are preserved.

A top-down methodology achieves a formal specification of the logic control structure, refines the model so that more system operation details can be included, and consequently implements the control at a physical level [23]. Choppy et al. [24] investigated a refinement method of colored Petri net subclass and proved the correctness of this method. For a kind of well-behaved Petri net, Ding et al. [25] proposed a refinement approach. Wang et al. [26] proposed a refinement approach which may not require reachability analysis. For systems specified in Elementary Net Systems, Bernardinello et al. [27] presented a refinement method for distributed systems. Lacheheb et al. [28] focused on analyzing the transformation of the business process (BP) to the Petri net and gave the transition refinement operation to verify soundness, liveness, and boundedness. Based on the UML state machine and the Petri net model, Łabiak et al. [29] presented a design of digital controllers. Petri nets were applied for macroplaces' hierarchy and some elements of modularity.

Cortés et al. [9] extended PRES+ by introducing the definition of hierarchy and proposed some concepts of equivalence for the PRES+ model. A strong refinement and a weak refinement of a certain transition were presented. These refinements preserve equivalence. In [11], we extended the concepts of refinement in [9] and investigated transition refinement approaches for PRES+.

In order to model, analyze, and verify large complex embedded systems, and to resolve PIRES+'s state space explosion problem, the main contributions of this work include: (1) refinement methods for PIRES+ are proposed; (2) the property preservation of the refinement methods is investigated.

Compared with several refinement methods for ordinary Petri nets [16,22], Colored Petri nets [18], and PRES+ [9,11], the refinement methods for PIRES+ can nicely describe the characteristics of embedded systems.

Specifically, we first give the refinement approaches for PIRES+, and then investigate their property preservations. Under some constraints, reachability, timing, functionality, liveness, and boundedness of PIRES+ are preserved. Since these refinement methods can preserve these important properties, there is no need for using reachability analysis when the PIRES+ model is expanded. Thus, these refinement approaches can mitigate PRES+'s state space explosion problem.

The rest of the paper is structured as follows. Basic notions of PIRES+ are presented in Section 2. Section 3 proposes the refinement methods of PIRES+. Section 4 investigates reachability, functionality, and timing preservation. Section 5 studies the preservation of boundedness and liveness of PIRES+. Section 6 illustrates the efficiency of the PIRES+ refinement methods with an applicable example. We conclude by summarizing the contributions of the refinement technique in Section 7.

## 2. Basic Notions of PIRES+

In the following, we add inhibitor arcs into PRES+, and obtain PIRES+. The PIRES+ model can capture an embedded system's timing aspects and obtain both control and data information which fully express complex data flow and control flow.

**Definition 1.** A Petri net with inhibitor arcs is a five-tuple  $N = (P, T, F, I, M)$ , where  $(P, T, F)$  represents a net,  $M_0$  is the initial marking,  $I \subseteq P \times T$  is a set of inhibitor arcs,  $I \cap F = \emptyset$ .

Figure 1 shows an example of the Petri net with inhibitor arcs. For example, there are inhibition arcs  $I = \{(p_3, t_1), (p_6, t_2)\}$  in Figure 1. The initial marking is  $[1,0,1,0,0,0]^T$ . At the initial marking  $M_0$ , for transitions  $t_1$  and  $t_2$ ,  $\forall pet_i : M(p) \geq 1, i \in \{1,2\}$ . Since  $M_0(p_3) = 1$  and  $(p_3, t_1) \in I$ ,  $t_1$  is not an enabled transition. At the same time,  $(p_6, t_2) \in I$ , and  $M_0(p_6) = 0$ ,  $t_2$  is an enabled transition.

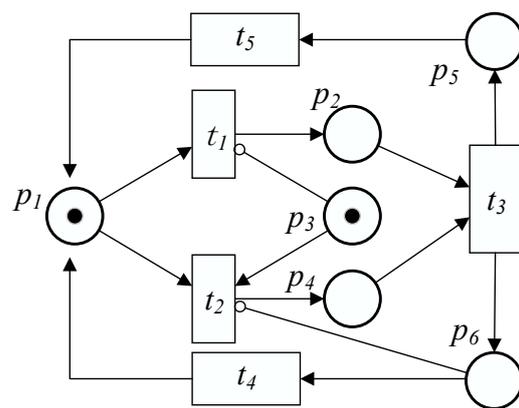


Figure 1. A Petri net with inhibitor arcs.

**Definition 2.** Ref. [9] A PRES+ model is  $N = (P, T, I, O, M_0)$ , where  $P = \{p_1, p_2, \dots, p_m\}$  is a finite non-empty set of places,  $T = \{t_1, t_2, \dots, t_n\}$  is a finite non-empty set of transitions,  $I \subseteq P \times T$  is a finite non-empty set of input arcs,  $O \subseteq T \times P$  is a finite non-empty set of output arcs, and  $M_0$  is the initial marking,  $k = \langle v, r \rangle$  is a token, where  $v$  is the token value, and  $r$  is the token time.

The inhibitor arcs are added to the PRES+, the PIREs+ (PRES+ with inhibitor arcs) is obtained.

**Definition 3.**  $N = (P, T, F_I, F_O, I, M_0)$  is a PIREs+ model, where  $P = \{p_1, p_2, \dots, p_m\}$  is a finite non-empty set of places,  $T = \{t_1, t_2, \dots, t_n\}$  is a finite non-empty set of transitions,  $F_I \subseteq P \times T$  is a finite non-empty set of input arcs,  $F_O \subseteq T \times P$  is a finite non-empty set of output arcs,  $I \subseteq P \times T$  is a set of inhibitor arcs, and  $M_0$  is the initial marking.

Figure 2 gives an example of a PIREs+ model, where  $P = \{p_0, p_1, p_2, p_3, p_4, p_5\}$ ,  $T = \{t_0, t_1, t_2, t_3, t_4\}$ ,  $F_I = \{(p_0, t_0), (p_1, t_1), (p_1, t_4), (p_2, t_2), (p_3, t_2), (p_4, t_3), (p_5, t_2)\}$ ,  $F_O = \{(t_0, p_1), (t_0, p_2), (t_1, p_3), (t_2, p_4), (t_3, p_0), (t_4, p_3)\}$ , and  $I = \{(p_5, t_4)\}$ .

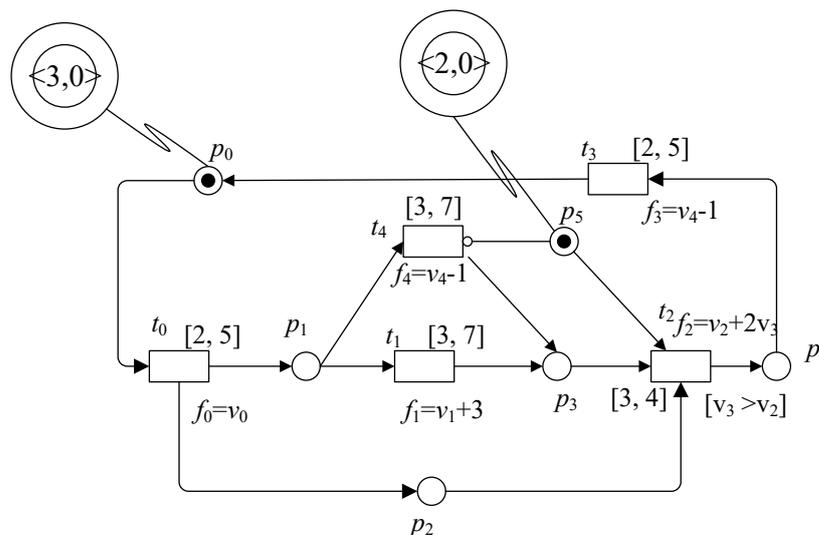


Figure 2. A PIREs+ model.

**Definition 4.** A token in PIREs+ is  $k = \langle v, r \rangle$ , where  $v$  is the token value, and  $r$  is the token time, which is a nonnegative real number describing the time stamp of the token.

In Figure 2, for instance, the token of place  $p_0$  is  $k_0 = \langle 3, 0 \rangle$ , where the token value is 3 and the time stamp is 0.

**Definition 5.** For every transition  $t \in T$ , there is transition function  $.f : \tau(p_1) \times \tau(p_2) \times \dots \times \tau(p_a) \rightarrow \tau(q)$ , where  $\tau$  is a type function,  $\bullet t = \{p_1, p_2, \dots, p_a\}$ ,  $q \in t \bullet$ .

For instance, in Figure 2,  $t_1$  has the function  $f_1 = v_1 + 3$ , where  $v_1$  is the value of the token that will appear in  $p_1$ , and after  $t_1$  fires, the value of the token in  $p_3$  will be the value of the token in  $p_1$  plus 3.

**Definition 6.** For every transition  $t \in T$ , there are  $d^-, d^+ \in R^+$ , where  $d^-$  is a minimum delay,  $d^+$  is a maximum delay, and  $R^+$  is the non-negative real number set.

For instance, in Figure 2, the delay of  $t_3$  is  $[2, 5]$ .

**Definition 7.** A transition  $t \in T$  may have a guard  $G : \tau(p_1) \times \tau(p_2) \times \dots \times \tau(p_a) \rightarrow \{0, 1\}$ , where  $\bullet t = \{p_1, p_2, \dots, p_a\}$ .

The guard is an important factor to determine whether  $t$  can be enabled. In Figure 2, transition  $t_2$  has guard  $G = \text{TURE}$  if  $v_3 > v_2$ , where  $v_2$  and  $v_3$  are the values of the tokens in places  $p_2$  and  $p_3$ , respectively.

**Definition 8.**  $b = (k_1, k_2, \dots, k_a)$  is a binding of transition  $t$ , where  $\forall p_i \in \bullet t, k_i \in M(p_i)$ .

**Definition 9.** Ref. [9] For an enabled transition  $t$ , there exists the enabling time  $et$ , where  $et$  is the time instant at which  $t$  becomes enabled and is given by the maximum token time of the tokens in the binding.

**Definition 10.** For an enabled transition  $t$ , there exists the earliest trigger time  $t^- = et + d^-$  and the latest trigger time  $t^+ = et + d^+$ .

As in P/T Petri nets, when a transition is enabled in the PIREs+ model, it can be fired.

**Definition 11.** The firing of transition  $t$  for a binding  $b$  changes  $M$  into a new marking  $M'$ :

- (i) Tokens from  $\bullet t$  are removed, i.e.,  $\forall p_i \in \bullet t, M'(p_i) = M(p_i) - \{k_i\}$ .
- (ii) A new token  $k = \langle v, r \rangle$  is put into each place of  $t \bullet$ , i.e.,  $M'(p) = M(p) + \{k\}$  for every  $p \in t \bullet$ .
- (iii) The marking of places different from  $\bullet t$  and  $t \bullet$  remain unchanged, i.e.,  $\forall p \in P - \{\bullet t \cup t \bullet\}, M'(p) = M(p)$ .

### 3. Refinement Operations for PIREs+

In this section, we will investigate the refinement operation for the PIREs+ model. First, several relevant concepts will be proposed. Then, we propose the refinement operations.

**Definition 12.** Suppose  $N = (P, T, F_I, F_O, I, M_0)$  is a PIREs+ model.  $N_p = (P_p, T_p, F_{I_p}, F_{O_p}, I_p, M_{p,0})$  is called a place subnet of  $N$  if and only if:

- (i)  $P_p \subset P, T_p \subset T$  and  $P_p \neq \emptyset, T_p \neq \emptyset$ ;
- (ii)  $F_{I_p} = F_I \cap (P_p \times T_p)$  and  $F_{O_p} = F_O \cap (T_p \times P_p)$ ;
- (iii)  $\bullet T_p \cup T_p \bullet \subseteq P_p$ ;
- (iv)  $I_p \subseteq I$ ;
- (v)  $N_p$  is connected,  $\{p_{in}, p_{out}\} \subseteq P_p$ , where  $p_{in}$  is the only input place of  $N_p$ ,  $p_{out}$  is the only output place of  $N_p$ ;
- (vi)  $\forall t \in T_p, \exists f : \tau(p_1) \times \tau(p_2) \times \dots \times \tau(p_a) \rightarrow \tau(q)$  (where  $q \in t \bullet, \bullet t = \{p_1, \dots, p_a\}$ );
- (vii)  $\forall t \in T_p, \exists d^-, d^+ \in R^+$ , where  $d^-$  is a minimum delay,  $d^+$  is a maximum delay,  $R^+$  is the set of non-negative real numbers, and  $d^- \leq d^+$ .

**Definition 13** For the transition set  $T_p$  of  $N_p$ , there is a minimum delay  $d_p^-$  and a maximum delay  $d_p^+$ , i.e.,  $\exists d_p^-, d_p^+ \in R^+$  such that  $d_p^- \leq d_p^+$ .

**Definition 14.** For the transition set  $T_p$  of  $N_p$ , there exists a transition set function  $f_p$ , i.e.,  $\exists f_p : \tau(p_1) \times \tau(p_2) \times \dots \times \tau(p_a) \rightarrow \tau(q)$ , where  $\bullet T_p = \{p_1, p_2, \dots, p_n\}$  and  $q \in T_p \bullet$ .

In order to investigate property preservations in Sections 4 and 5, we propose the following supposition for the PIREs+ subnet.

**Supposition 1.** Suppose that PIREs+ subnet  $N_p = (P_p, T_p, F_{I_p}, F_{O_p}, I_p, M_{p,0})$  satisfies:

- (i) In a process, i.e., tokens flow from outside of  $N_p$  into  $p_{in}$ , through  $N_p$ , then flow out from  $p_{out}$ , the tokens' number flowing into  $p_{in}$  is equal to that flowing out from  $p_{out}$ .
- (ii)  $p_{in}$  is the unique place of  $P_p$  which may contain tokens.

In this subsection, we present the PIREs+ refinement operations.

**Definition 15.** PIREs+ refinement operation  $Ref_p(p_p, N_p)$ : let the refined PIREs+  $N' = (P', T', F_I', F_O', I', M_0')$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using  $N_p = (P_p, T_p, F_{I_p}, F_{O_p}, I_p, M_{p,0})$  to replace  $p_p$  where,

- (i)  $P' = P_p \cup P - \{p_p\}$ ;  $T' = T_p \cup T$ ;  $I' = I \cup I_p$ ;

- (ii) If  $(p, t) \in F_{Ip} \cup F_I$ , then  $(p, t) \in F_I'$ ; If  $(t, p) \in F_{Op} \cup F_O$ , then  $(t, p) \in F_O'$ ;
- (iii) If  $(t, p_p) \in F_O$ , then  $(t, p_{in}) \in F_O'$ ; If  $(p_p, t) \in F_I$ , then  $(p_{out}, t) \in F_I'$ ;
- (iv)  $M_{0'} = \begin{cases} \left( M_{(P-\{p_p\})0}, \theta_p \right) M_0(p_p) = 0 \\ \left( M_{(P-\{p_p\})}, \theta_{p0} \right) M_0(p_p) > 0 \end{cases}$  (where  $M_{(P-\{p_p\})}$  is the projection of  $M$  on  $P - \{p_p\}$ , and  $\theta_p$  is 0-vector of  $M_p$ );
- (v)  $f_p \circ f_{\bullet p_{in}} = f_{\bullet p_p}, f_{out\bullet} = f_{p_p\bullet}$ ;
- (vi)  $d_{\bullet p_{in}}^- + d_p^- = d_{\bullet p_p}^-, d_{\bullet p_{in}}^+ + d_p^+ = d_{\bullet p_p}^+; d_{p_{out}\bullet}^- = d_{p_p\bullet}^-, d_{p_{out}\bullet}^+ = d_{p_p\bullet}^+$ .

Note that, although in Definition 15, only  $p_p$  of  $N$  is replaced by a subnet  $N_p$ , this refinement operation can be extended to the place set refinement operation. Let the place set  $P_{set}$  have a finite number of places. Every place of  $P_{set}$  may not be in conflict with other places of  $P_{set}$ , i.e.,  $\forall p_1, p_2 \in P_{set}, \bullet p_1 \cap \bullet p_2 = \emptyset$  and  $p_1 \bullet \cap p_2 \bullet = \emptyset$  (where  $p_1 \neq p_2$ ). In other words, every place of  $P_{set}$  may not share output transitions and input transitions with other places of  $P_{set}$ .

**Supposition 2.** Suppose (i)  $P_{set} = \{p_1, p_2, \dots, p_m\} (m \geq 2)$  is a place set of  $N$ , with  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$  for  $\forall p_i, p_j \in P_{set}$  (where  $i \neq j$ ); (ii)  $N_{set} = (N_{p1}, N_{p2}, \dots, N_{pm})$ , where  $N_{pi} (i = 1, 2, \dots, m)$  is the subnet of  $N$ .

**Definition 16.** PIREs+ place set refinement operation  $Re f_{p_{set}}(P_{set}, N_{set})$ : let the refined PIREs+  $N' = (P', T', F_I', F_O', I', M_0')$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using place subnets  $N_{p1}, N_{p2}, \dots, N_{pm}$  to replace  $p_1, p_2, \dots, p_m$  of  $P_{set}$ , respectively.

Note that, since every place of  $P_{set}$  may not share output transitions and input transitions with other places of  $P_{set}$ , the specific execution process of this place set operation is as follows. First, let  $N_1 = (P_1, T_1, F_{I1}, F_{O1}, I_1, M_{1,0})$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using  $N_{p1}$  to replace  $p_1$  of  $N$ ; second, let  $N_2 = (P_2, T_2, F_{I2}, F_{O2}, I_2, M_{2,0})$  be obtained from  $N_1 = (P_1, T_1, F_{I1}, F_{O1}, I_1, M_{1,0})$  using  $N_{p2}$  to replace  $p_2$  of  $N_1$ ; ... ; last, let  $N' = (P', T', F_I', F_O', I', M_0')$  be obtained from  $N_{m-1} = (P_{m-1}, T_{m-1}, F_{I,m-1}, F_{O,m-1}, I_{m-1}, M_{m-1,0})$  using  $N_{pm}$  to replace  $p_m$  of  $N_{m-1}$ .

#### 4. Preservation of Reachability, Timing, and Functionality for PIREs+

In this section, we investigate the preservation of reachability, functionality, and timing of the refined PIREs+ model by using the refinement operations. First, we propose the concepts of the property preservations of PIREs+. Second, we study the reachability, timing, and functionality preservations of PIREs+.

**Definition 17.** PIREs+ subnets  $N_{p1}$  and  $N_{p2}$  are said to have the same reachability iff:

- (i) The input place's number and the output place's number of  $N_{p1}$  are the same as those of  $N_{p2}$ ;
- (ii) The tokens' number within the input places of  $N_{p1}$  is equal to that of  $N_{p2}$ , and when these tokens go through  $N_{p1}$  and  $N_{p2}$ , respectively, the number of tokens of  $N_{p1}$  is the same as that of  $N_{p2}$ .

**Definition 18.** PIREs+ subnets  $N_{p1}$  and  $N_{p2}$  are said to have the same functionality iff:

- (i) The reachability of  $N_{p1}$  is the same as that of  $N_{p2}$ ;
- (ii) If when the token type of tokens in the input places of  $N_{p1}$  is equal to that of  $N_{p2}$ , then the token type of tokens in the output places of  $N_{p1}$  is equal to that of  $N_{p2}$ .

**Definition 19.** PIREs+ subnets  $N_{p1}$  and  $N_{p2}$  are said to have the same timing iff:

- (i) The reachability of  $N_{p1}$  is the same as that of  $N_{p2}$ ;
- (ii) The token time of tokens in the input places of  $N_{p1}$  is equal to that of  $N_{p2}$ , and the token time of tokens in the output places of  $N_{p1}$  is equal to that of  $N_{p2}$ .

In the following, reachability, functionality, and timing preservations of these refinement operations will be investigated.

**Theorem 1.** Let  $PIRES+ N' = (P', T', F'_I, F'_O, I', M'_0)$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using  $PIRES+$  refinement operation  $Re f_p(p_p, N_{pp})$ . Then,  $N'$  and  $N$  have the same reachability, functionality, and timing.

**Proof.** Suppose subnet  $N_1 = (P_1, T_1, F_{I1}, F_{O1}, I_1, M_{1,0})$ , where  $P_1 = \{\bullet(\bullet p_p), p_p, (p_p \bullet)\bullet\}$ ,  $T_1 = \{\bullet p_p\} \cup \{p_p \bullet\}$ ,  $F_{I1} = \{(p_p, t) | t \in p_p \bullet\} \cup \{(p, t) | p \in \bullet(\bullet p_p) \wedge t \in \bullet p_p\}$ ,  $F_{O1} = \{(t, p) | t \in p_p \bullet \wedge p \in (p_p \bullet)\bullet\} \cup \{(t, p_p) | t \in \bullet p_p\}$ ,  $I_1 = \emptyset$ ,  $M_{pp,0} = M_0|_{P_{pp}}$ .

Suppose subnet  $N_2 = (P_2, T_2, F_{I2}, F_{O2}, I_2, M_{2,0})$  where  $P_2 = \{\bullet(\bullet p_p), (p_p \bullet)\bullet\} \cup P_{pp}$ ,  $T_2 = \{\bullet(\bullet p_p), (p_p \bullet)\bullet\} \cup T_{pp}$ ,  $F_{I2} = F_{Ipp} \cup \{(p, t) | t \in \bullet p_p \wedge p \in \bullet(\bullet p_p)\} \cup \{(p_{out}, t) | t \in p_p \bullet\}$ ,  $F_{O2} = F_{Opp} \cup \{(t, p_{in}) | t \in \bullet p_p\} \cup \{(t, p) | p \in (p_p \bullet)\bullet \wedge t \in p_p \bullet\}$ ,  $I_2 = I_{pp} \cup I_1$ . If  $p \in P_{pp}$ , then  $M_{2,0}(p) = M_{pp,0}(p)$ . If  $p \in \{\bullet(\bullet p_p), (p_p \bullet)\bullet\}$ , then  $M_{2,0}(p) = M_{pp,0}(\{(p_p \bullet)\bullet \cup \bullet(\bullet p_p)\}(p))$ .  $\square$

We can see that  $N_1$  and  $N_2$  have the same input places and output places. According to Definition 15, the number of tokens in input places of  $N_1$  is the same as that of  $N_2$ . Since  $PIRES+ N'$  is obtained from  $N$  by using  $N_2$  to replace  $p_p$ , by Supposition 1, Definitions 12, 15, the number of tokens in output places of  $N_1$  is the same as that of  $N_2$ . By Definition 17,  $N'$  and  $N$  have the same reachability.

According to Definition 15, the type of tokens in input places of  $N_1$  is the same as that of  $N_2$ . Since  $f_{pp} \circ f_{\bullet p_{in}} = f_{\bullet p_p}$ ,  $f_{p_{out} \bullet} = f_{p_p \bullet}$ , then the type of tokens in output places of  $N_1$  is the same as that of  $N_2$ . By Definition 18,  $N'$  and  $N$  have the same functionality.

By Definition 15, the time of tokens in input places of  $N_1$  is the same as that of  $N_2$ . Since  $d_{\bullet p_{in}}^- + d_{p_p}^- = d_{\bullet p_p}^-$ ,  $d_{p_{in}}^+ + d_{p_p}^+ = d_{\bullet p_p}^+$ ,  $d_{p_{out} \bullet}^- = d_{p_p \bullet}^-$  and  $d_{p_{out} \bullet}^+ = d_{p_p \bullet}^+$ , by Definition 15, the time of tokens in output places of  $N_1$  is the same as that of  $N_2$ . By Definition 19,  $N'$  and  $N$  have the same timing.

Since  $N' - N_2 = N - N_1$ , then  $N'$  and  $N$  have the same reachability, functionality, and timing.  $\square$

Note that, by Supposition 2, Definitions 15, 16, and Theorem 1, we can analyze the reachability, timing, and functionality preservations of the place set refinement operation.

**Theorem 2.** Suppose  $P_p = \{p_1, p_2, \dots, p_m\}$  ( $m \geq 2$ ) is the place set of  $PIRES+ N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ . Let  $N' = (P', T', F'_I, F'_O, I', M'_0)$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using place set refinement operation  $Re f_{P_p}(P_p, N_p)$ . Then,  $N'$  and  $N$  have the same reachability, functionality, and timing.

**Proof.** Since  $P_p = \{p_1, p_2, \dots, p_m\}$  ( $m \geq 2$ ) is the place set of  $N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ , we can refine these places one by one. First, let  $N_1 = (P_1, T_1, F_{I1}, F_{O1}, I_1, M_{1,0})$  be obtained from  $N = (P, T, F_I, F_O, I, M_0)$  using  $N_{pp1}$  to replace  $p_1$  of  $N$ , by the proof process of Theorem 1,  $N_1$  and  $N$  have the same reachability, functionality, and timing. Second, let  $N_2 = (P_2, T_2, F_{I2}, F_{O2}, I_2, M_{2,0})$  be obtained from  $N_1 = (P_1, T_1, F_{I1}, F_{O1}, I_1, M_{1,0})$  using  $N_{pp2}$  to replace  $p_2$  of  $N_2$ , by the proof process of Theorem 1,  $N_2$  and  $N_1$  have the same reachability, functionality, and timing. In the case of  $p_3, \dots, p_m$ , we can prove it in the same way. Thus,  $N'$  and  $N$  have the same reachability, functionality, and timing.  $\square$

### 5. Preservation of Liveness and Boundedness for PIRES+

In Petri net system, liveness and boundedness are the two important behavior properties. The liveness reflects that the net system can complete any part of work in any state. This means that there is no local deadlock in the Petri net system. The boundedness reflects the overflow resistance of the Petri net system.

Some studies have been conducted in the field of Petri net refinement aiming to realize the preservation of liveness and boundness [11,16,17,22]. Huang et al. [16] proposed several refinement approaches for Place/Transition Petri nets to preserve liveness and boundedness. Xia [22] presented a refinement operation for ordinary Petri nets and investigated the preservation of reversibility, boundedness, and liveness of the refined nets. In [11], we analyzed a kind of transition refinement method for PRES+ to preserve liveness and boundedness.

In this section, we investigate the refinement method of PIREs+, and propose certain constraints to preserve liveness and boundedness of PIREs+.

To explore the preservation of liveness and boundedness of the refinement approaches, the concepts of the state, liveness, and boundedness of PIREs+ should be presented.

**Definition 20.**  $S = (M_v, J)$  is said to be a state of PIREs+ model  $N = (P, T, F_I, F_O, I, M_0)$ , where  $M_v \in R(M_0)$  (where  $R(M_0)$  is the marking reachable set), and  $J : T \rightarrow R^+ \cup \{\#\}$  (where  $\#$  denotes the unusable status).

**Definition 21.**  $S_0 = (M_0, J_0)$  with  $J_0(t) = \begin{cases} 0 & M_0(p) \geq W(p, t) \forall p \in \bullet t \\ \#, & \text{otherwise} \end{cases}$  (where  $W(p, t)$  denotes the weight function on  $(p, t)$ ) is called the initial state of  $N$ .

**Definition 22.** Let  $Z = (P, T, F_I, F_O, I)$  be the skeleton of  $N = (P, T, F_I, F_O, I, M_0)$ .  $\Sigma = (Z, S_0)$  (where  $S_0 = (M_0, J_0)$ ) is called the PIREs+ net system of  $N = (P, T, F_I, F_O, I, M_0)$ .

**Definition 23.** Let  $S$  be a reachable state, and  $t \in T$ . If  $\forall S' \in R_N(S), \exists S'' \in R_N(S')$ , such that  $S'' \xrightarrow{t}$ , then  $t$  is said to be live. If  $\forall t, t$  is live, then  $\Sigma = (Z, S_0)$  is said to be live.

**Definition 24.** If  $S \in R_N(S_0), \exists K > 0$  (where  $K$  is a natural number) such that  $M(p) \leq K$ , then  $p$  is called bounded. For  $\forall p$ , if  $p$  is bounded, then the PIREs+ net system  $\Sigma = (Z, S_0)$  is said to be bounded.

In order to investigate liveness and boundedness preservations of the refinement operations, the concept of the PIREs+ place-closed net system should be proposed.

**Definition 25.**  $\bar{\Sigma} = (\bar{Z}_p, \bar{S}_{p,0})$  is called a place-closed PIREs+ net system if we add a transition  $t_p$  (where its transition delay is  $[d_{t_p}^-, d_{t_p}^+]$ , and the transition function is  $f_{t_p}$ ) and arcs  $\{(p, t_p) | p \in (p_p \bullet)\}, \{(t_p, p) | p \in \bullet(p_p)\}$ , to  $(Z_p, S_{p,0})$  and the marking of  $(Z_p, S_{p,0})$  is preserved.

In the following, the preservation of liveness and boundedness for PIREs+ will be studied.

**Theorem 3.** Suppose PIREs+ net system  $\Sigma' = (Z', S_0')$  is obtained from  $\Sigma = (Z, S_0)$  by the refinement operation  $Re f_p(p_p, N_p)$ . If  $\bullet(\bullet p_{pin}) \in \{p | M'_0(p) = 0 \wedge p \in P'\}$ , then  $\Sigma' = (Z', S_0')$  is live iff  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma} = (\bar{Z}_p, \bar{S}_{p,0})$  are live.

**Proof.** (If)  $\forall t' \in T'$ , there exists  $t' \in T$  or  $t' \in T_p$  in  $\Sigma' = (Z', S_0')$ . According to Supposition 1, for  $\forall S' \in R(M_0')$  (where  $S' = [M_{(P-\{p_p\})}, M_p]$ ), there exists  $M \in R(M_0)$  and  $M_p \in R(M_{p,0})$ . If  $t' \in T$ , according to Definition 23,  $\exists \bar{M} \in R(M), \bar{S} \xrightarrow{t'}$  for  $M \in R(M_0)$ . According to Definition 23, Definition 15, and Supposition 1,  $\exists \bar{S}' = ((M_{(P-\{p_p\})}, M_p), J_0') \in R(M')$ , such that  $\bar{S} \xrightarrow{t'}$  (where  $\bar{M} \in R(M), \bar{M}_p \in R(M_p)$ ). So, in  $\Sigma' = (Z', S_0')$ ,  $t'$  is live. If  $t' \in T_p$ , according to Definition 23,  $\exists \bar{M}_p \in R(M_p)$ , such that  $\bar{S}_p \xrightarrow{t'}$ . According to Definition 23, Definition 15, and Supposition 1,  $\exists \bar{S}'' = ((M'_{(P-\{p_p\})}, M_p), J_0') \in R(M')$ , such that  $\bar{S}'' \xrightarrow{t'}$ . Then,  $t'$  is live in  $\Sigma' = (Z', S_0')$ .

(Only-if) Let  $\Sigma' = (Z', S_0')$  be live. Suppose that  $\Sigma = (Z, S_0)$  is not live, that is,  $\exists S \in R(S_0), \exists t \in T, \forall \bar{S} \in R(S)$ , such that  $\neg(\bar{S} \xrightarrow{t})$ . Suppose  $S_{\{P-p_p\},0} \xrightarrow{\sigma, \tau} S \xrightarrow{\bar{\sigma}, \bar{\tau}} \bar{S}$ , where  $\bar{\sigma}, \bar{\tau} \in T$ , and  $S_{\{P-p_p\},0}$  is the projection of  $S''$  on  $P - \{p_p\}$ . Next, we add corresponding transition step  $\sigma_p$  of  $\Sigma_p = (Z_p, S_{p,0})$  and obtain  $\sigma', \sigma^* \in T'', S_{\{P-p_p\},0} \xrightarrow{\sigma, \tau} S_{\{P-p_p\}} \xrightarrow{\bar{\sigma}, \bar{\tau}} S_{\{P-p_p\}}$ . According to Definition 23, Definition 15, and Supposition 1,  $S_0' \xrightarrow{\sigma', \tau'} S'' \xrightarrow{\sigma'', \tau''} \bar{S}''$ , the projection of  $S''$  on  $\bar{S}$  is  $\bar{S}$ , then  $\exists t' \in T', S'' \in R(S_0'')$ ,  $\forall \bar{S}'' \in R(S')$ , such that  $\neg(\bar{S} \xrightarrow{t'}) \rightarrow \neg(\bar{S}'' \xrightarrow{t'})$ . That conflicts with the liveness of  $\Sigma' = (Z', S_0')$ . Hence, there exist  $S_0'', S_0''' \in R(S_0')$ , such that  $\Sigma = (Z, S_0)$  obtained from  $(Z', S_0'')$  is live and  $\Sigma_p = (Z_p, S_{p,0})$  obtained from  $(Z'', S_0''')$  is live. Since  $\bullet(\bullet p_{pin}) \in \{pM_0'(p) > 0 \wedge p \in P'\}$ , then  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma} = (\bar{Z}_p, \bar{S}_{p,0})$  are live.  $\square$

**Theorem 4.** Suppose that the PIREs+ net system  $\Sigma' = (Z', S_0')$  is obtained from  $\Sigma = (Z, S_0)$  by refinement operation  $Re f_p(p_p, N_p)$ .  $\Sigma' = (Z', S_0')$  is bounded iff  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma} = (\bar{Z}_p, \bar{S}_{p,0})$  are bounded.

**Proof.** (If) Since  $\Sigma = (Z, S_0)$  is bounded,  $\forall p \in P, \exists k_1 > 0$  such that  $\forall S \in R(S_0), M(p) \leq k_1$ . Obviously,  $\forall p \in P - \{p_p\}, M_{(P-\{p_p\})}(p) \leq k_1$ . Since  $\bar{\Sigma} = (\bar{Z}_p, \bar{S}_{p,0})$  is bounded, then for every  $p \in P_p, \exists k_2 > 0$ , such that  $\forall S_p \in R(S_{p,0}), M_p(p) \leq k_2$ . Let  $k = k_1 + k_2$ , by Definitions 12–15 and Supposition 1,  $\forall p \in P', \forall S' \in R(S_0'), M'(p) = (M_{(P-\{p_p\})}, M_p)(p) \leq k$ . By Definition 24,  $\Sigma' = (Z', S_0')$  is bounded.

(Only-if) Without loss of generality, suppose  $\Sigma = (Z, S_0)$  is not bounded, such that  $\exists p \in P, \forall k > 0, \exists S \in R(S_0), M(p) > k$ . By Definitions 12–14, and Supposition 1,  $\forall k > 0, \exists S' \in R(S_0'), M'(p) > k$ . That conflicts with the fact that  $\Sigma' = (Z', S_0')$  is bounded.

Note that, by Supposition 2, Definitions 15, 16, and Theorems 3 and 4, we can investigate the preservation of liveness and boundedness of the PIREs+ place set refinement operation.  $\square$

**Theorem 5.** Suppose  $P_p = \{p_1, p_2, \dots, p_m\} (m \geq 2)$  is the PIREs+ place set of  $N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ . Let  $\Sigma' = (Z', S_0')$  be obtained from  $\Sigma = (Z, S_0)$  using the PIREs+ place set refinement operation  $Ref_{p_{set}}(P_{set}, N_{set})$ . If  $\bullet(\bullet p_{p_{jin}}) \in \{p|M_0'(p) > 0 \wedge p \in P'\}$  (where  $j = 1, 2, 3, \dots, m$ ), then  $\Sigma' = (Z', S_0')$  is live if and only if  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma}_{pj} = (\bar{Z}_{pj}, \bar{S}_{pj0})$  (where  $j = 1, 2, 3, \dots, m$ ) are live.

**Proof.** Since  $P_p = \{p_1, p_2, \dots, p_m\} (m \geq 2)$  is the place set of  $N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ , we can refine the places of  $P_p$  one by one. First, let  $\Sigma_1 = (Z_1, S_{1,0})$  be obtained from  $\Sigma = (Z, S_0)$  using  $N_{p_1}$  to replace  $p_1$  of  $\Sigma = (Z, S_0)$ . Since  $\bullet(\bullet p_{p_{1in}}) \in \{pM_0'(p) > 0 \wedge p \in P'\}$ , by the proof process of Theorem 3,  $\Sigma_1 = (Z_1, S_{1,0})$  is live if and only if  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma}_{p_1} = (\bar{Z}_{p_1}, \bar{S}_{p_10})$  are live. Second, let  $\Sigma_2 = (Z_2, S_{2,0})$  be obtained from  $\Sigma_1 = (Z_1, S_{1,0})$  using  $N_{p_2}$  to replace  $p_2$  of  $\Sigma_1 = (Z_1, S_{1,0})$ . Since  $\bullet(\bullet p_{p_{2in}}) \in \{pM_0'(p) > 0 \wedge p \in P'\}$ , by the proof process of Theorem 3,  $\Sigma_2 = (Z_2, S_{2,0})$  is live if and only if  $\Sigma_1 = (Z_1, S_{1,0})$  and  $\bar{\Sigma}_{p_2} = (\bar{Z}_{p_2}, \bar{S}_{p_20})$  are live. In the case of  $p_3, \dots, p_m$ , we can prove it in the same way. So,  $\Sigma' = (Z', S_0')$  is live if and only if  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma}_{pj} = (\bar{Z}_{pj}, \bar{S}_{pj0})$  (where  $j = 1, 2, 3, \dots, m$ ) are live.  $\square$

**Theorem 6.** Suppose  $P_p = \{p_1, p_2, \dots, p_m\} (m \geq 2)$  is the place set of  $N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ . Let  $\Sigma' = (Z', S_0')$  be obtained from  $\Sigma = (Z, S_0)$  using the PIREs+ place set refinement operation  $Ref_{p_{set}}(P_{set}, N_{set})$ .  $\Sigma' = (Z', S_0')$  is bounded if and only if  $\Sigma = (Z, S_0)$  and  $\bar{\Sigma}_{pj} = (\bar{Z}_{pj}, \bar{S}_{pj0})$  (where  $j = 1, 2, 3, \dots, m$ ) are bounded.

**Proof.** Since  $P_p = \{p_1, p_2, \dots, p_m\} (m \geq 2)$  is the place set of  $N$ , and  $\forall p_i, p_j \in P_p$  (where  $i \neq j$ ),  $\bullet p_i \cap \bullet p_j = \emptyset$  and  $p_i \bullet \cap p_j \bullet = \emptyset$ , we can refine these places  $p_1, p_2, \dots, p_m$  one by one. First, let  $\Sigma_1 = (Z_1, S_{1,0})$  be obtained from  $\Sigma = (Z, S_0)$  using  $N_{p_1}$  to replace  $p_1$  of  $\Sigma = (Z, S_0)$ . By the proof process of Theorem 4,  $\Sigma_1 = (Z_1, S_{1,0})$  is bounded if and only if  $\Sigma = (Z, S_0)$  and  $\overline{\Sigma}_{p_1} = (\overline{Z}_{p_1}, \overline{S}_{p_10})$  are bounded. Second, let  $\Sigma_2 = (Z_2, S_{2,0})$  be obtained from  $\Sigma_1 = (Z_1, S_{1,0})$  using  $N_{p_2}$  to replace  $p_2$  of  $\Sigma_1 = (Z_1, S_{1,0})$ . By the proof process of Theorem 4,  $\Sigma_2 = (Z_2, S_{2,0})$  is bounded if and only if  $\Sigma_1 = (Z_1, S_{1,0})$  and  $\overline{\Sigma}_{p_2} = (\overline{Z}_{p_2}, \overline{S}_{p_20})$  are bounded. In the case of  $p_3, \dots, p_m$ , we can prove it in the same way. So,  $\Sigma' = (Z', S_0')$  is bounded if and only if  $\Sigma = (Z, S_0)$  and  $\overline{\Sigma}_{p_j} = (\overline{Z}_{p_j}, \overline{S}_{p_j0})$  (where  $j = 1, 2, 3, \dots, m$ ) are bounded.  $\square$

## 6. Application

The contribution of this work is to propose the place refinement operation and the place set refinement operation and verify the relevant important properties. To demonstrate the effectiveness of these refinement approaches, we use the refinement approaches for PIRES+ to model and analyze an embedded control system.

With the rapid development of communication technology, the frequency of communication technology innovation is gradually accelerating. 5G, as a new generation cellular mobile network, has its advantages of excellent wide area coverage, good mobility, low price, and high sharing of WLAN. In the actual network, cellular communication network and WLAN are two independent systems.

As we all know, WLAN is the product of the combination of computer network and wireless communication technology. It uses radio frequency technology (RF) to replace the old twisted pair to form a local area network. WLAN has many corresponding access specifications, all of which are in IEEE 802.11 standard. There is mainly a/b/ac/ad/n/ac. At present, 802.11.ac is widely used. When a piece of information is received through the wireless network, these information data are processed and decoded by the physical layer, data link layer, network layer, transport layer, and application layer, then restored to the most original information transmitted by the sender. We have simplified the structure of the transmission process, analyzed the whole information processing process from a macro perspective.

In order to effectively realize the real integration between heterogeneous networks, this section applies the refinement method of PIRES+ proposed above to the modeling and analysis of mobile terminal network communication system. The refined operation of the network system can effectively achieve the integration of heterogeneous networks and accurately respond to the system workflow, and facilitate the future improvement and migration of the system. Next, the cellular mobile communication, WLAN network, and digital signal processing process of the mobile terminal network communication system will be modeled and analyzed.

### 6.1. Constructing a PIRES+ Model

We will construct an embedded control system's PIRES+ model using the place set refinement operation. First, an abstract PIRES+ model of an embedded control subsystem will be presented. Second, the refined PIRES+ model is obtained using the PIRES+ refinement operations. The embedded control system's abstract model ( $N$ ) is illustrated in Figure 3.

In Figure 3,  $p_1$ : cellular network signal processing part;  $p_7$ : digital signal processing part;  $p_{11}$ : WLAN signal processing part.  $t_1$ : convert electromagnetic wave to RF-Radio Frequency;  $t_2$ : the baseband information is obtained by adjusting the filter;  $t_3$ : start digital signal processing;  $t_4$ : finish digital signal processing;  $t_5$ : pulse code modulation;  $t_6$ : verify WLAN signal password;  $t_7$ : the signal received from the antenna is converted into AC signal.

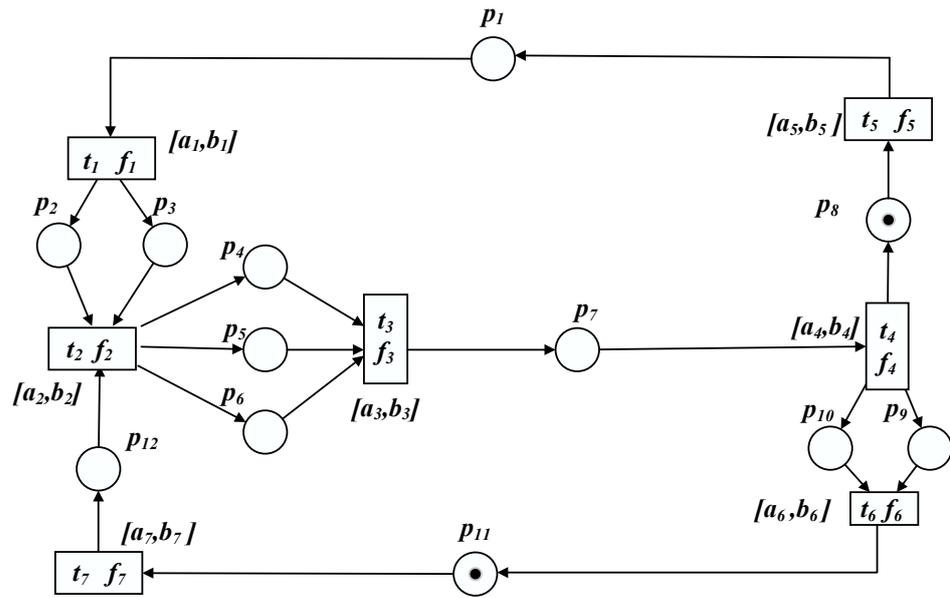


Figure 3. The abstract model  $N$ .

The subsystem  $N_1$  is displayed in Figure 4.  $N_1$  is the PIREs+ model of cellular network signal processing part.  $t_{p11}$ : adjust audio;  $t_{p12}$ : perform human-computer interaction;  $t_{p13}$ : finish current signal processing and initialize the system;  $t_{p14}$ : the antenna receives the electromagnetic wave signal sent by the base station. There exist transition functions  $f_{p11}, f_{p12}, f_{p13}, f_{p14}$  associated with transitions  $t_{p11}, t_{p12}, t_{p13}, t_{p14}$ , respectively. There exists transition set function  $f_{p1}$  of subnet  $N_1$ , where  $f_{p1} = f_{p11} \circ f_{p12} \circ f_{p13} \circ f_{p14}$ .  $[a_{p11}, b_{p11}], [a_{p12}, b_{p12}], [a_{p13}, b_{p13}], [a_{p14}, b_{p14}]$ , are transition delay intervals associated with transitions  $t_{p11}, t_{p12}, t_{p13}, t_{p14}$  respectively.  $[d_{p1}^-, d_{p1}^+]$  is the transition delay interval of subnet  $N_1$ , where  $d_{p1}^- = a_{p11} + a_{p12} + a_{p13} + a_{p14}, d_{p1}^+ = b_{p11} + b_{p12} + b_{p13} + b_{p14}$ .

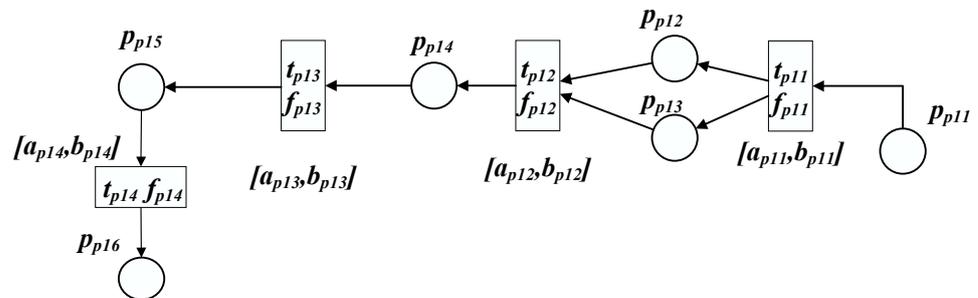


Figure 4. The subnet model  $N_1$ .

The subsystem  $N_2$  is displayed in Figure 5.  $N_2$  is the PIREs+ model of the digital signal processing part, where  $t_{p21}, t_{p22}, t_{p23}, t_{p24}, t_{p25}, t_{p26}$ , represent a series of signal processing processes, such as acquisition, filtering, transformation, spectrum analysis, estimation, recognition, etc. There exist transition functions  $f_{p21}, f_{p22}, f_{p23}, f_{p24}, f_{p25}, f_{p26}$ , associated with transitions  $t_{p21}, t_{p22}, t_{p23}, t_{p24}, t_{p25}, t_{p26}$ , respectively. There exists transition set function  $f_{p2}$  of subnet  $N_2$ , where  $f_{p2} = f_{p21} \circ ((f_{p23} \parallel f_{p22}) \circ (f_{p24} \parallel f_{p25})) \circ f_{p26}$ .  $[a_{p21}, b_{p21}], [a_{p22}, b_{p22}], [a_{p23}, b_{p23}], [a_{p24}, b_{p24}], [a_{p25}, b_{p25}], [a_{p26}, b_{p26}]$  are transition delay intervals associated with transitions  $t_{p21}, t_{p22}, t_{p23}, t_{p24}, t_{p25}, t_{p26}$ , respectively. There exists transition delay interval  $[d_{p2}^-, d_{p2}^+]$  of subnet  $N_2$ , where  $d_{p2}^- = a_{p21} + \max(a_{p23} + a_{p24}, a_{p22} + a_{p25}) + a_{p26}, d_{p2}^+ = b_{p21} + \max(b_{p22} + b_{p25}, b_{p23} + b_{p24}) + b_{p26}$ .

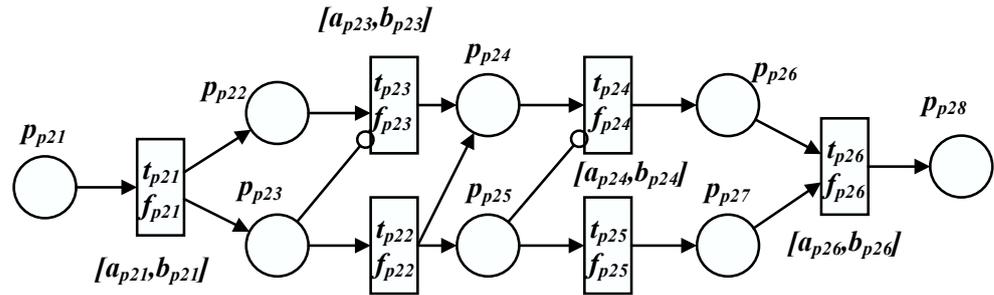


Figure 5. The subnet model  $N_2$ .

The subsystem  $N_3$  is displayed in Figure 6.  $N_3$  is the PIRES+ model of the WLAN signal processing part, where  $t_{p31}$ : human-computer interaction part;  $t_{p32}$ : finish the current signal processing and initialize the system;  $t_{p33}$ : the terminal receives WLAN signals from different sources. There exist transition functions  $f_{p31}$ ,  $f_{p32}$ ,  $f_{p33}$ , associated with transitions  $t_{p31}$ ,  $t_{p32}$ ,  $t_{p33}$ , respectively. There exists transition set function  $f_{p3}$  of subnet  $N_3$ , where  $f_{p3} = f_{p31} \circ f_{p32} \circ f_{p33}$ . There exist transition delay intervals  $[a_{p31}, b_{p31}]$ ,  $[a_{p32}, b_{p32}]$ ,  $[a_{p33}, b_{p33}]$  associated with transitions  $t_{p31}$ ,  $t_{p32}$ ,  $t_{p33}$ , respectively.  $[d_{p3}^-, d_{p3}^+]$  is the transition delay interval of subnet  $N_3$ , where  $d_{p3}^- = a_{p31} + a_{p32} + a_{p33}$ ,  $d_{p3}^+ = b_{p31} + b_{p32} + b_{p33}$ .

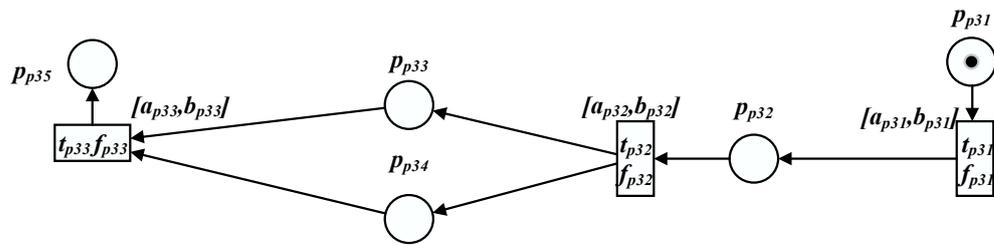


Figure 6. The subnet model  $N_3$ .

The refined PIRES+ model  $N'$  (Figure 7) is obtained using the refinement method. Place  $p_1$ ,  $p_7$ , and  $p_{11}$  of  $N$  (Figure 3) are replaced by place subnet  $N_1$ ,  $N_2$ , and  $N_3$ , respectively.

In Figure 7, corresponding to place subnet  $N_1$ , the transition function of  $t_5$  is  $f_5'$ , where  $f_{p1} \circ f_5' = f_5$ . There exists a transition delay interval  $[a_5', b_5']$  associated with transition  $t_5$ , where  $d_{p_{11}}^- + d_{N_1}^- = d_{p_1}^-$ ,  $d_{p_{11}}^+ + d_{N_1}^+ = d_{p_1}^+$ , that is  $a_5' + d_{N_1}^- = a_5$ ,  $b_5' + d_{N_1}^+ = b_5$ . By place subnet  $N_2$ , the transition function of  $t_3$  is  $f_3'$ , where  $f_{p2} \circ f_3' = f_3$ . There exists a transition delay  $[a_3', b_3']$  associated with transition  $t_3$ , where  $d_{p_{21}}^- + d_{N_2}^- = d_{p_7}^-$ ,  $d_{p_{21}}^+ + d_{N_2}^+ = d_{p_7}^+$ , that is,  $a_3' + d_{N_2}^- = a_3$ ,  $b_3' + d_{N_2}^+ = b_3$ . Corresponding to place subnet  $N_3$ , the transition function of  $t_6$  is  $f_6'$ , where  $f_{p6} \circ f_6' = f_6$ . There exists a transition delay interval  $[a_6', b_6']$  associated with transition  $t_6$ , where  $d_{p_{31}}^- + d_{N_3}^- = d_{p_{11}}^-$ ,  $d_{p_{31}}^+ + d_{N_3}^+ = d_{p_{11}}^+$ , that is  $a_6' + d_{N_3}^- = a_6$ ,  $b_6' + d_{N_3}^+ = b_6$ .

## 6.2. Property Analysis of the PIRES+ Model

The preservation of several properties will be analyzed in this section.

### 6.2.1. Preservation of Reachability, Functionality, and Timing

Let PIRES+ subnet  $N_{11} = (P_{11}, T_{11}, F_{11}, F_{O1}, I_1, M_{1,0})$  of  $N$  (Figure 3), where  $P_{11} = \{p_1, p_2, p_3, p_8\}$ ,  $T_{11} = \{t_1, t_5\}$ ,  $F_{11} = \{(p_1, t_1), (p_8, t_5)\}$ ,  $F_{O1} = \{(t_1, p_2), (t_1, p_3), (t_5, p_1)\}$ ,  $I_1 = \emptyset$ ,  $M_{1,0}(p_8) \neq \emptyset$ . Let  $N_{12} = (P_{12}, T_{12}, F_{12}, F_{O2}, I_2, M_{2,0})$ , where  $P_{12} = \{p_{p11}, p_{p12}, p_{p13}, p_{p14}, p_{p15}, p_{p16}, p_2, p_3, p_8\}$ ,  $T_{12} = \{t_1, t_{p11}, t_{p12}, t_{p13}, t_{p14}, t_5\}$ ,  $F_{12} = \{(p_{p11}, t_{p11}), (p_{p12}, t_{p12}), (p_{p13}, t_{p12}), (p_{p14}, t_{p13}), (p_{p15}, t_{p14}), (p_{p16}, t_1), (p_8, t_5)\}$ ,  $F_{O2} = \{(t_1, p_2), (t_1, p_3), (t_5, p_{p11}), (t_{p11}, p_{p12}), (t_{p11}, p_{p13}), (t_{p12}, p_{p14}), (t_{p13}, p_{p15}), (t_{p14}, p_{p16})\}$ ,  $I_2 = \emptyset$ ,  $M_{2,0}(p_8) \neq \emptyset$ .

Let PIRES+ subnet  $N_{21} = (P_{21}, T_{21}, F_{13}, F_{O3}, I_3, M_{3,0})$  of  $N$  (Figure 3), where  $P_{21} = \{p_4, p_5, p_6, p_7, p_9, p_8, p_{10}\}$ ,  $T_{21} = \{t_3, t_4\}$ ,  $F_{13} = \{(p_4, t_3), (p_5, t_3), (p_6, t_3), (p_7, t_4)\}$ ,  $F_{O3} =$



$N_1, N_2$  and  $N_3, \overline{\Sigma_{p1}} = (\overline{Z_{p1}}, \overline{S_{p1,0}}), \overline{\Sigma_{p2}} = (\overline{Z_{p2}}, \overline{S_{p2,0}})$  and  $\overline{\Sigma_{p3}} = (\overline{Z_{p3}}, \overline{S_{p3,0}})$  are live and bounded. By Theorem 5 and Theorem 6,  $\Sigma' = (Z', S_0')$  is live and bounded.

### 6.3. Summary of the Modeling

The above system has clear structural hierarchy, time characteristics, and task priority requirements, so it is difficult to model with ordinary Petri net, colored Petri net, or PRES+. First, this example establishes the overall simplified model of mobile terminal communication network system, cellular mobile communication model based on embedded system, WLAN network model, and other component models. Then, the refinement operation is carried out to obtain the overall system model. Finally, through formal analysis, it is proven that the refined net system preserves the reachability, functionality, timing, liveness, and boundedness of the original net system. This modeling method solves the problem of direct modeling difficulty caused by the complexity of the system and the portability loss of some components in the overall modeling. The example effectively illustrates the feasibility and practicability of the refinement operation proposed in this paper and provides a more practical theoretical basis for the modeling of some embedded systems.

## 7. Conclusions

Embedded systems are widely used in various devices and are becoming more and more complex with increasing requirements. To facilitate the verification and design of embedded systems, PIREs+ is introduced. PIREs+ has great expressive power and many applications in verification, modeling, and analysis of embedded systems.

However, the problem of state space explosion is a disadvantage for PIREs+'s ability to model, analyze, and verify large, complex systems. In order to resolve the state space explosion of Petri nets, there are three popular transformations in the literature, namely synthesis approach, refinement approach, and reduction approach. In order to model, analyze, and verify large complex embedded systems, and to solve PIREs+'s state space explosion problem, we have proposed the place refinement approach and the place set refinement approach for PIREs+. Compared with refinement methods for ordinary Petri nets, CPNs, and PRES+, these refinement methods can nicely describe certain characteristics of embedded systems. We have also investigated the property preservation of these refinement approaches. During the refining process, under certain constraints, reachability, timing, functionality, liveness, and boundedness will be preserved. The refinement methods for PIREs+ can be used to nicely resolve design problems of embedded systems.

Of course, the refinement operation methods proposed in this paper are only applicable to the operation of place in PIREs+, without considering the refinement operation of transition in PIREs+. At the same time, this paper only carries out qualitative research from the formal aspect, and lacks certain quantitative analysis and system simulation.

In the future, some quantitative analysis and system simulation for the PIREs+'s refinement operation should be studied. Other more general refinement, synthesis, or reduction approaches for PIREs+ should also be investigated, for example, a place-bordered refinement operation, or synthesis operation methods for PIREs+ and their applications.

Fuzzy linear programming [40,41] is a generalization of classical linear programming, which blurs the boundary of linear constraints. Under loose constraints, we can obtain the optimization conditions and optimization extremum. The combination of the fuzzy linear programming and the refinement operation may become another interesting research direction in the future.

**Author Contributions:** Conceptualization, C.X. and Z.W. (Zhuangzhuang Wang); methodology, C.X.; validation, C.X., Z.W. (Zhuangzhuang Wang) and Z.W. (Zhong Wang); formal analysis, C.X. and Z.W. (Zhuangzhuang Wang); investigation, C.X. and Z.W. (Zhong Wang); resources, C.X.; writing—original draft preparation, C.X.; writing—review and editing, Z.W. (Zhuangzhuang Wang); visualization, Z.W. (Zhong Wang); supervision, C.X.; project administration, C.X.; funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Science Foundation of Shandong Province (Grant: ZR2016FM19).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rashid, M.; Anwar, M.W.; Khan, A.M. Toward the tools selection in model based system engineering for embedded systems—A systematic literature review. *J. Syst. Softw.* **2015**, *106*, 150–163. [\[CrossRef\]](#)
2. Ashjaei, M.; Khalilzad, N.; Mubeen, S. Modeling, designing and analyzing resource reservations in distributed embedded systems. In *Real-Time Modeling and Processing for Communication Systems, Lecture Notes in Networks and Systems*; Alam, M., Dghais, W., Chen, Y., Eds.; Springer: Cham, Switzerland, 2018; Volume 29, pp. 203–256.
3. Tausan, N.; Markkula, J.; Kuvaja, P.; Oivo, M. Choreography in the embedded systems domain: A systematic literature review. *Inform. Softw. Technol.* **2017**, *91*, 82–101. [\[CrossRef\]](#)
4. Stoutchini, A.; Benini, L. StreamDrive: A dynamic dataflow framework for clustered embedded architectures. *J. Signal. Process. Sys.* **2019**, *91*, 630–640.
5. Offutt, J.; Thumala, S. Testing Concurrent user behavior of synchronous web applications with Petri nets. *Softw. Syst. Model.* **2019**, *18*, 913–936. [\[CrossRef\]](#)
6. Firdaus, A.F.; Meribout, M. A new parallel VLSI architecture for real-time electrical capacitance tomography. *IEEE T. Comput.* **2016**, *65*, 30–41. [\[CrossRef\]](#)
7. Wu, D.; Schnieder, E. Scenario-based system design with colored Petri nets: An application to train control systems. *Softw. Syst. Model.* **2018**, *17*, 295–317. [\[CrossRef\]](#)
8. Gutierrez, A.; Bressan, M.; Jimenez, J.F.; Alonso, C. Real-time emulation of boost inverter using the systems modeling language and Petri nets. *Math. Comput. Simulat.* **2019**, *158*, 216–234. [\[CrossRef\]](#)
9. Cortés, L.A.; Eles, P.; Peng, Z. Modeling and formal verification of embedded systems based on a Petri net based representation. *J. Syst. Arch.* **2003**, *49*, 571–598. [\[CrossRef\]](#)
10. Karlsson, D.; Eles, P.; Peng, Z. Formal verification of component-based designs. *Des. Autom. Embed. Syst.* **2007**, *11*, 49–90. [\[CrossRef\]](#)
11. Xia, C. Property preservation of refinement for Petri net based representation for embedded systems. *Cluster Compu.* **2016**, *19*, 1373–1384. [\[CrossRef\]](#)
12. Xia, C.; Shen, B.; Zhang, H.; Wang, Y. Liveness and boundedness preservations of sharing synthesis of Petri net based representation for embedded systems. *Comput. Syst. Sci. Eng.* **2018**, *33*, 345–350. [\[CrossRef\]](#)
13. Bandyopadhyay, S.; Sarkar, D.; Mandal, C. Equivalence checking of Petri net models of programs using static and dynamic cut-points. *Acta. Inform.* **2019**, *56*, 321–383. [\[CrossRef\]](#)
14. Xia, C.; Li, C. Property preservation of Petri synthesis net based representation for embedded systems. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 905–915. [\[CrossRef\]](#)
15. Ivanov, S.; Pelz, E.; Verlan, S. Small universal non-deterministic Petri nets with inhibitor arcs. *Comput. Sci.* **2014**, *8614*, 186–197.
16. Huang, H.; Cheung, T.-y.; Mak, W.M. Structure and behavior preservation by Petri-net-based refinements in system design. *Theor. Compu. Sci.* **2004**, *328*, 245–269. [\[CrossRef\]](#)
17. Padberg, J.; Urášek, M. Rule-Based Refinement of Petri Nets: A Survey. In *Petri Net Technology for Communication-Based Systems*; Ehrig, H., Reisig, W., Rozenberg, G., Weber, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 161–196.
18. Lakos, C.; Lewis, G. Incremental state space construction of coloured Petri nets. In *International Conference on Application and Theory of Petri Nets 2021*; Colom, J.-M., Koutny, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 263–282.
19. Mejía, G.; Niño, K.; Montoya, C.; Sánchez, M.A.; Palacios, J.; Amodeo, L. A Petri net based framework for realistic project management and scheduling: An application in animation and videogames. *Comput. Oper. Res.* **2016**, *66*, 190–198. [\[CrossRef\]](#)
20. Van Hee, K.M.; Sidorova, N.; van der Werf, N.J.M. Business process modeling using Petri nets. In *Transactions on Petri Nets and Other Models of Concurrency VII*; Jensen, K., Koutny, M., Balbo, G., Wolf, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 116–161.
21. Li, J.; Zhou, M.; Dai, X. Reduction and refinement by algebraic operations for Petri net transformation. *IEEE Trans. Sys. Man Cy.—Part A Syst. Hum.* **2012**, *42*, 1244–1255. [\[CrossRef\]](#)
22. Xia, C. Analysis of properties of Petri synthesis net. In *Theory and Applications of Models of Computation*; Cai, J.-Y., Cooper, S.B., Li, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 576–587.
23. Pang, S.; Zhang, W.; Ma, T.; Gao, Q. A colony optimization algorithm to dynamic energy management in cloud datacenter. *Math. Probl. Eng.* **2017**, *2*, 1–10.
24. Choppy, C.; Mayero, M.; Petrucci, L. Coloured Petri net refinement specification and correctness proof with Coq. *Innov. Syst. Softw. Eng.* **2010**, *6*, 195–202. [\[CrossRef\]](#)

25. Ding, Z.; Jiang, C.; Zhou, M.; Zhang, Y. Preserving languages and properties in stepwise refinement-based synthesis of Petri nets. *IEEE Trans. Sys., Man Cy.—Part A Syst. Hum.* **2008**, *38*, 791–801. [[CrossRef](#)]
26. Wang, S.; You, D.; Zhou, M.; Seatzu, C. Characterization of admissible marking sets in Petri nets with uncontrollable transitions. *IEEE T. Automa. Contr.* **2016**, *61*, 1953–1958. [[CrossRef](#)]
27. Bernardinello, L.; Kılınc, G.; Mangioni, E.; Pomello, L. Modeling distributed private key generation by composing Petri nets. In *Transactions on Petri Nets and Other Models of Concurrency IX*; Koutny, M., Haddad, S., Yakovlev, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 19–40.
28. Lacheheb, M.N.; Hameurlain, N.; Maamri, R. Resources consumption analysis of business process services in cloud computing using Petri Net. *J. King Saud Univ.—Comput. Inf. Sci.* **2020**, *32*, 408–418. [[CrossRef](#)]
29. Łabiak, G.; Wegrzyn, M.; Rosado-Muñoz, A.; Bazydło, G. Dual-model approach for safety-critical embedded systems. *Microprocess. Microsyst.* **2020**, *72*, 102924. [[CrossRef](#)]
30. Berthomieu, B.; Botlan, D.L.; Zilio, S.D. Counting Petri net markings from reduction equations. *Int. J. Softw. Tools Technol. Transf.* **2020**, *22*, 163–181. [[CrossRef](#)]
31. Shah, S.A.; Bohez, E.L.J.; Shah, K.; Haq, I.U.; Azam, K.; Anwar, S. Colored Petri net model for significant reduction of invariants in flexible manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 1775–1787. [[CrossRef](#)]
32. Bourdil, P.-A.; Berthomieu, B.; Zilio, S.D.; Vernadat, F. Symmetry reduction for time Petri net state classes. *Sci. Compu. Program.* **2016**, *132*, 209–225. [[CrossRef](#)]
33. Berthomieu, B.; Botlan, D.L.; Zilio, S.D. Yakovlev. In *Model Checking Software*; Gallardo, M.M., Merino, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 65–84.
34. Chen, Y.; Li, Z.; Barkaoui, K.; Uzam, M. New petri net structure and its application to optimal supervisory control: Interval inhibitor arcs. *IEEE Trans. Syst. Man. Cy Syst.* **2014**, *44*, 1384–1400. [[CrossRef](#)]
35. Luo, J.; Liu, Z.; Wang, S.; Xing, K. Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 812–821. [[CrossRef](#)]
36. Liu, G.; Li, P.; Li, Z.; Wu, N. Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs. *IEEE Trans. Syst. Man. Cy Syst.* **2018**, *49*, 1371–1385. [[CrossRef](#)]
37. Chen, H.; Wu, N.; Li, Z.; Qu, T. On a maximally permissive deadlock prevention policy for automated manufacturing systems by using resource-oriented Petri nets. *ISA. Trans.* **2019**, *89*, 67–76. [[CrossRef](#)] [[PubMed](#)]
38. Wolf, K. Petri net synthesis with union/find. In *Application and Theory of Petri Nets and Concurrency*; Khomeño, V., Roux, O.H., Eds.; Springer: Cham, Switzerland, 2018; Volume 2, pp. 60–81.
39. Best, E.; Devillers, R.; Schlachter, U. Bounded choice free Petri net synthesis: Algorithmic issues. *Acta. Inform.* **2017**, *55*, 575–611. [[CrossRef](#)]
40. Ebrahimnejad, A. An effective computational attempt for solving fully fuzzy linear programming using MOLP problem. *J. Ind. Prod. Eng.* **2019**, *36*, 59–69. [[CrossRef](#)]
41. Ebrahimnejad, A.; Verdegay, J.L. An efficient computational approach for solving type-2 intuitionistic fuzzy numbers based Transportation Problems. *Int. J. Comput. Int. Sys.* **2016**, *9*, 1154–1173. [[CrossRef](#)]