machines

Article

Fault Detection, Diagnostics, and Treatment in Automated Manufacturing Systems Using Internet of Things and Colored Petri Nets

Husam Kaid [1,*], Abdulrahman Al-Ahmari [2,3,*] and Khaled N. Alqahtani [1]

[1] Industrial Engineering Department, College of Engineering, Taibah University, Medina 42353, Saudi Arabia
[2] Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[3] Raytheon Chair for Systems Engineering (RCSE Chair), Advanced Manufacturing Institute, King Saud University, Riyadh 11421, Saudi Arabia
* Correspondence: hkaid@taibahu.edu.sa (H.K.); alahmari@ksu.edu.sa (A.A.-A.)

Abstract: Internet of things (IoT) applications, which include environmental sensors and control of automated manufacturing systems (AMS), are growing at a rapid rate. In terms of hardware and software designs, communication protocols, and/or manufacturers, IoT devices can be extremely heterogeneous. Therefore, when these devices are interconnected to create a complicated system, it can be very difficult to detect and fix any failures. This paper proposes a new reliability design methodology using "colored resource-oriented Petri nets" (CROPNs) and IoT to identify significant reliability metrics in AMS, which can assist in accurate diagnosis, prognosis, and resulting automated repair to enhance the adaptability of IoT devices within complicated cyber-physical systems (CPSs). First, a CROPN is constructed to state "sufficient and necessary conditions" for the liveness of the CROPN under resource failures and deadlocks. Then, a "fault diagnosis and treatment" technique is presented, which combines the resulting network with IoT to guarantee the reliability of the CROPN. In addition, a GPenSIM tool is used to verify, validate, and analyze the reliability of the IoT-based CROPN. Comparing the results to those found in the literature shows that they are structurally simpler and more effective in solving the deadlock issue and modeling AMS reliability.

Keywords: automated manufacturing systems; Petri net; reliability; deadlock; internet of things; simulation

Citation: Kaid, H.; Al-Ahmari, A.; Alqahtani, K.N. Fault Detection, Diagnostics, and Treatment in Automated Manufacturing Systems Using Internet of Things and Colored Petri Nets. *Machines* **2023**, *11*, 173. https://doi.org/10.3390/machines11020173

Academic Editors: Ratna Babu Chinnam and Murat Yildirim

Received: 22 December 2022
Revised: 15 January 2023
Accepted: 24 January 2023
Published: 27 January 2023

1. Introduction

The growth of the "internet of things" (IoT) and "information and communications technology" (ICT) [1] has made it possible for "cyber-physical systems" [2] to be used to control and manage AMS. The IoT is a set of real objects equipped with "sensors", "software", and other ICT systems that enable them to interact with other systems and devices over the internet. ICT, on the other hand, is any type of "computing/communication" equipment, "networking devices", and "information systems" that make it easier to interact in the digital world. The IoT is an essential component of the next generation of information methods; it is the internet that connects physical objects, which in turn connect the internet with the real world. In accordance with the agreement, physical objects are linked to the internet for information communication and exchange via "radio frequency identification" (RFID), "sensors", and a few other tools. Therefore, things can be intelligently identified, located, and controlled [3].

Cyber-physical systems (CPS) integrate sensor networks and embedded computing to control and monitor the physical environment in the context of manufacturing systems, including control loops that enable environmental stimuli to activate the control, communication, or computing automatically [4]. The development of the IoT and ICT provides the capability to control, monitor, and manage CPS via the prompt delivery of real-time data from the manufacturing system. In CPS of manufacturing systems, machine/resource failures are common. Managers are frequently concerned with whether the planned due dates of orders can be achieved if machines fail, i.e., if the AMS is robust against failures. Even though IoT and ICT can detect real-time information in CPS, including "resource failures", poor response to "resource failures" can damage CPS and reduce performance. The CPS can only be robust when the real-time data presented by "IoT" and "ICT" is employed to enhance decision-making and adapt to a dynamic environment in a timely way. Information from IoT and ICT about "resource failures" in real time is only relevant if an efficient system exists to assess and analyze the impacts of "resource malfunctions" and deal with them [3]. Therefore, the CPS requires a strategy to evaluate the effects of "resource failures", support decision-making, and take the necessary actions in response to "resource failures". This motivates us to design a technique to evaluate the effects of "resource failures" on CPS operations and provide the necessary decision support.

Several manufacturers have concentrated their efforts on improving the effectiveness of computer numerical control (CNC) machine tools with high accuracy and superior surface quality. In CNC machines, managing frequent failures, such as "tool wear", "noise", and "tool breakage", is a problem. Tool wear is a complex process to control because it is not a linear operation; tools degrade quickly at first, then at a reasonable rate for a duration of time, and finally at an increasing rate until complete failure. Tool wear is a common and unavoidable problem during most machining processes. It presents a variety of obstacles to the quality and productivity of automated machining operations, impeding the possibility of intelligent and integrated industrial companies [5].

Nowadays, the construction of an intelligent online method for condition monitoring based on industrial IoT for manufacturing systems has become a solution for advanced predictive maintenance systems that continuously monitor and record the status of CNC machine tools [6].

Recently, effective fault diagnosis and detection approaches based on IoT, CPS, and artificial machine learning have been developed. The authors of [7] developed a new IoT architecture for online monitoring of induction motor faults. The developed IoT architecture recognizes the fault categories of the motor using effective machine learning methods. In addition, cyber-attacks are considered, and the proposed IoT architecture can detect and avoid cyber-attacks. The results highlight the advantages of the proposed IoT infrastructure to accurately identify motor faults and cyberattacks. The authors of [8] proposed a new smart IoT system for vibration monitoring and control of the milling CNC machine center. The milling machine was connected to a force sensor to record the vibrations caused by the cutting process. The IoT system employing the "message queue telemetry transport" (MQTT) protocol was built to connect the sensor node to the cloud server in order to record the cutting conditions. Deep learning neural networks are used to classify cutting states such as "stable cutting", "unstable cutting", and "fake cutting". Moreover, if there is a cyberattack, the proposed method can switch the IoT system automatically to the backup broker to keep cutting operations safe and reliable. The authors of [9] developed and prototyped an "event-driven tool condition monitoring" (EDTCM) system that is activated "just in time" after the workpiece begins processing. The authors of [10] proposed a flexible and simple tool condition monitoring system in which the algorithms use repetitive machining processes in mass production and employ similarity analysis between the data of known conditions and received signals. The authors of [11] reviewed and evaluated direct and indirect sensing approaches for tool condition monitoring in milling, and discussed the strengths, drawbacks, and future prospects of such measurement techniques. The authors of [12] conducted research on energy harvesting systems for "wireless sensor nodes" (WSNs), directly addressing the issue of wireless sensor batteries' sustainability.

The authors of [13] presented a practical lightweight model EPN and a definition “language EPNML” based on Petri nets (PNs) to represent the IoT services’ behavior and business processes, such as process control and logic relationship architecture. To improve the efficiency and adaptability of IoT services, this model was developed. The authors of [14] modeled the connection between different sensors and the “geographic environment” by employing “colored Petri nets” (CPNs). As data and processing services, they categorize the “IoT service”, the “geographic entity”, and the “Geographic Information System” (GIS) service. In order to do this, CPNs and an algebra were used to represent and evaluate “geo characteristics”, “IoT”, and “Geographic Information System” services, and the connection activity between a “sensor” node and its “geographic environment”. Using three different case studies, they illustrated the validity of the CPN-based assessment method. The authors of [15] presented a CPN technique to represent the global behavior of “wireless sensor networks”, which contains the energy usage of the network. CPNs and the principle of “component-oriented modeling” form the basis of their method. Two components are highlighted: the “radio system” and the “MAC Protocol”. Models for each subnet and interface were then developed separately. The study evaluated the network based on its energy usage, allowing them to estimate its life. In [16], CPNs were used to represent the “1-wire protocol” that is employed to connect a master device with many small, inexpensive devices, including “digital thermometers” and “weather sensors”, throughout a single shared bus with “low data rates” and “large ranges”.

As aforementioned in the literature, manufacturers use careful operational technology to avoid these failures in order to avoid costly tool wear and damage [17]. This, unfortunately, leads to less productive and costly operations due to the requirement to replace the tool early, the downtime of a machine, and replacement costs. To solve this problem, manufacturers can apply sensors to monitor tool conditions and manage the system in real time. These sensors include acoustic emission sensors [18], accelerometers and current sensors [19]. Because a single sensor cannot accurately determine the status of a tool under varying operational conditions, combining the data from multiple sensors can be the main obstacle. Using information that is partially redundant, this data acquisition can provide decision-making information for a process with less uncertainty. Earlier solutions attempted to derive a tool’s status from data by extracting relevant characteristics. Therefore, this study aims to propose a “colored resource-oriented Petri nets” (CROPNs)-based IoT for sensor systems in order to detect and remedy failures. Using the main CROPN properties, we are able to construct a system with a huge number of sensors. The study’s purpose is to present a technique for modeling reliability, based on CROPN and IoT, to identify critical reliability measures in AMS, including “mean time to failure”, “mean time to repair”, and “availability”. First, a CROPN that considers “resource failures” is designed to guarantee the liveness of the CROPN. Then, a method is developed for “fault detection and treatment” that integrates IoT and CROPN to ensure the system’s reliability. Moreover, the system’s reliability is evaluated using CROPN and IoT. The following is a list of the significant contributions of the study:

(1) The development of a new approach for modeling the reliability and deadlock control of complicated AMS.

(2) The IoT-based CROPN is used to present tool-wear monitoring on a milling machine as an example of an actual system that is generally difficult to model, evaluate performance, and monitor tool status on various machines.

(3) The GPenSIM tool [20–24] is used to propose a comprehensive simulation program for the IoT-based CROPN that is then employed to simulate, verify, and evaluate results.

The rest of this paper is structured as follows: Section 2 shows the development of the CROPN and its “deadlock avoidance” strategy. The CROPN and IoT combination for “fault detection and treatment” and the reliability model are illustrated in Section 3. In

Section 4, we present the actual AMS example that demonstrates the experimental outcomes of the designed methodology. In Section 5, the developed CROPN based on the IoT is validated and compared to other existing approaches. Conclusions and research directions are presented in Section 6.

2. Synthesis Method of CROPN

Based on its stated processing paths, a ROPN defines a component production process as a component sequentially visiting the resources. ROPN expresses machines and buffers as H-resources [25–28] of the same resource type. Each resource is created in a single location. The paths a product travels to obtain its resources define its production processes. If each component's production process is constructed in an efficient manner, the model is highly compressed and demonstrates a number of important structural characteristics. Moreover, the ROPN considers material handling equipment to be a special type of resource, known as G-resources [25–28]. Due to the restricted resources available in the production system, a limited capacity PN with its related firing rules is employed. A transition in a ROPN indicates the movement of a part from one resource to another. The AMS can be modeled by the ROPN in two steps: (1) designing the component manufacturing process regardless of component transportation, and (2) designing the component handling operations using the model constructed in step 1.

A CROPN can be expressed by $N = (P, T, C, I, O, K, M)$, where

(1) $P = \{p_o\} \cup \{p_r\} \cup P_R$ is a number of model places, where p_o is a load/unload idle place, p_r is a global place that represents material handling equipment resources, and $P_R = \cup_{i\in m}\{p_i\}$, represents a set of model resource places, $m > 0$;

(2) $T = \cup_{j\in n}\{t_j\}$ represents a number of model transitions, where $n > 0$, $P \cap T = \phi$, and $P \cup T \neq \phi$;

(3) $C(p)$ and $C(t)$ are, respectively, the sets of color related to CROPN places and transitions, where

(4) $\forall p_i \in P$, $C(p_i) = \cup_{i\in m}\{a_{iu}\}$, $u = | C(p_i) |$

(5) $\forall t_j \in T$, $C(t_j) = \cup_{j\in n}\{b_{jv}\}$, $v = | C(t_j) |$;

(6) $I(p, t)$: $C(p) \times C(t) \rightarrow \mathbf{IN}$ represents the function of the input of the CROPN, and $\mathbf{IN} = \{0, 1, 2, \ldots\}$;

(7) $O(p, t)$: $C(p) \times C(t) \rightarrow \mathbf{IN}$ represents the function of the output of the CROPN;

(8) K: $P \rightarrow \mathbf{IN}$ denotes the capacity function, which allocates the maximum tokens for each $p_i \in P$ and is represented as $K(p_i)$;

(9) M: $P \rightarrow N$ indicates the function of the CROPN state (marking) that adds tokens to each place in a CROPN. $M(p_i)$ represents the tokens in place p_i regardless of its color. $M(p_i, a_{ij})$ represents the tokens in place p_i with color a_{ij}. M_o denotes the initial state of the CROPN.

Assuming that $F= I(p_i, t_j) \cup O(p_i, t_j)$, $\forall(p_i, t_j) \in F$, the preset (input) and postset (output) transitions of place p_i can be expressed as "$^\bullet p_i = \{t_j \in T \mid (t_j, p_i) \in F\}$" and "$p_i^\bullet = \{t_j \in T \mid (p_i, t_j) \in F\}$", respectively. Similarly, the preset (input) and postset (output) places of transition t_j can be, respectively, expressed as "$^\bullet t_j = \{p_i \in P \mid (p_i, t_j) \in F\}$" and " $t_j^\bullet = \{p_i \in P \mid (t_j, p_i) \in F\}$". A CROPN can be called

(1) ordinary when "$I(p_i, t_j)(a_{ih}, b_{jk}) = 1$", $\forall(p_i, t_j) \in F$, $p_i \in P$, $t_j \in T$;

(2) weighted when "$I(p_i, t_j)(a_{ih}, b_{jk}) > 1$", $(p_i, t_j) \in F$, $\exists p_i \in P$, and $\exists t_j \in T$;

(3) self-loop-free when "$I(p_i, t_j)(a_{ih}, b_{jk}) > 0$", "$O(p_i, t_j)(a_{ih}, b_{jk}) = 0$", and $\forall(p_i, t_j) \in P \cup T$;

(4) self-loop when "$I(p_i, t_j)(a_{ih}, b_{jk}) > 0$", "$O(p_i, t_j)(a_{ih}, b_{jk}) > 0$", and $\forall(p_i, t_j) \in P \cup T$.

In a CROPN, a $t_j \in T$ is "process-resource-enabled" [28] if

$$\text{“}M(p_i, a_{ih}) \geq I(p_i, t_j)(a_{ih}, b_{jk}),\ \forall p_i \in P,\ \forall p_i \in {}^\bullet t_j,\ a_{ih} \in C(p_i),\ b_{ik} \in C(t_j),\text{”} \quad (1)$$

and

$$\text{“}K(p_i) \geq M(p_i, a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk}) - I(p_i, t_j)(a_{ih}, b_{jk}),\ \forall p_i \in P,\ \forall p_i \in t_j^\bullet,\ a_{ih} \in C(p_i),\ b_{ik} \in C(t_j).\text{”} \quad (2)$$

At state M, if a transition $t_j \in T$ is enabled, it fires, and the state changes from M to M', denoted as $M[t_j\rangle M'$ and stated as

$$\text{“}\forall p_i \in P,\ a_{ih} \in C(p_i),\ b_{ik} \in C(t_j),\ M'(p_i, a_{ih}) = M(p_i, a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk}) - I(p_i, t_j)(a_{ih}, b_{jk}).\text{”} \quad (3)$$

A CROPN has multiple circuits due to its high connectivity. Production process circuits (PPCs) are unique circuits in a CROPN that play a significant influence in its liveness. There are no idle places in PPCs, which are denoted as PPCs = $\{e_1, e_2, \ldots, e_r\}$, $r > 0$, where e_r is a circuit in PPCs. When an e_r flows from a node a to multiple nodes and then back to the originating node a without duplicating any nodes, it said to be an elementary circuit. An e_r is non-elementary if it does not return to the initial node a. Moreover, in the e_r, the sets transitions must equal the sets of places, such that “$|P(e_r)| = |T(e_r)|$”, where $P(e_r)$ represents the number of places in the e_r and $T(e_r)$ denotes the number of transitions in the e_r, respectively, and the input places of transition t_j for an e_r “$({}^\bullet t_j \in P(e_r),\ p_i \in {}^\bullet t_j)$” must be included in the e_r. If the t_j in a circuit e_r fires and the tokens leave the e_r, the tokens are described as the “departing tokens” and the “cycling tokens” when they do not leave a circuit e_r, and can be formulated as

$$p_i \in P(e_r),\ M(e_r) = \Sigma M(p_i, e_r) \quad (4)$$

where $M(p_i, e_r)$ indicates the number of tokens in p_i, which enables t_j in an e_r.

It is said that a circuit e_r^c is an interactive subnet, which contains c PPCs, $c > 0$, if its transitions and places are shared with at least one other PPC and its connections are strong. If a transition t_j in e_r^c ($p_i \in P(e_r^c)$ and $t_j^\bullet \in (p_i^\bullet \cap T(e_r^c))$) fires and the tokens leave the e_r^c, where $P(e_r^c)$ represents the set of places and $T(e_r^c)$ denotes the set of transitions in an e_r^c, the tokens in e_r^c are known as the “departing tokens”; otherwise, they are said to be the “cycling tokens”, and can be stated as

$$p_i \in P(e_r^c),\ M(e_r^c) = \Sigma M(p_i, e_r^c) \quad (5)$$

where $M(p_i, e_r^c)$ denotes the number of tokens in p_i, which enables t_j in an e_r^c.

In a CROPN,

(1) if the t_j satisfies conditions in Equations (1) and (2), it said to be a controlled transition;
(2) if there is at least one controlled transition t_j, then the CROPN is called a controlled net;
(3) if a circuit e_r satisfies the conditions in Equations (1) and (2), then it is said to be enabled;
(4) if the $t_j \in T(e_r)$ is live (deadlock-free), the t_j is called live;
(5) the t_j is called an input and output transition of an e_r^c if the $t_j \not\subset T(e_r^c)$, $t_j^\bullet \in P(e_r^c)$, and ${}^\bullet t_j \in P(e_r^c)$.

The following are the conditions necessary for a deadlock-free CROPN:

1. **Condition 1** [28]**:** A CROPN is not deadlock-free if

$$\text{“}M_o(p_o) \geq K(e_r).\text{”} \quad (6)$$

2. **Condition 2** [28]**:** At any state $M \in R(N, M_o)$, a circuit e_r is live if

$$\text{“}S'(e_r) \geq 1,\text{”} \quad (7)$$

where $R(N, M_o)$ represents the CROPN reachable states, and $S'(e_r)$ is the current number of spaces in an e_r, and can be expressed as

$$\text{“}S'(e_r) = K(e_r) - M(e_r).\text{”} \quad (8)$$

3. **Condition 3:** At any state $M \in R(N, M_o)$, when Condition 2 is satisfied, transitions in $T(e_r)$ and $T_I(e_r)$ are controlled, where $T_I(e_r)$ represents the set of input transitions in an e_r (see DFC-Policy in [29]);

4. **Condition 4** [28]**:** At any state $M \in R(N, M_o)$, a circuit e_r^c is deadlock-free if

$$\text{“}S'(e_r) \geq 1,\text{”}\ \forall e_r \tag{9}$$

and

$$\text{“}\eta(e_r^c, M) \geq 1,\text{”} \tag{10}$$

where $\eta(e_r^c, M)$ denotes the enabled PPCs in an e_r^c

5. **Condition 5** [28]**:** At any marking M, an e_r^c is deadlock-free if
   (a) any $t_j \in T_I(e_r^c)$ is controlled, where $T_I(e_r^c)$ denotes the set of input transitions in the e_r^c;
   (b) before a controlled t_j fires, each $e_r \in V_{en}(t_j)$, $S'(e_r) \geq 2$, where $V_{en}(t_j)$ represents the set of PPCs in the CROPN and the t_j denotes the input transition of these PPCs;
   (c) the state M can be modified to M' after the firing of the t_j, such that “$\eta(e_r^c, M) \geq 1$”.
6. **Condition 6** [28]**:** If a CROPN has no PPC, then it is always live.

Based on the above conditions, Algorithm 1 shows the deadlock avoidance algorithm for a CROPN model.

Algorithm 1: *Policy for avoiding deadlock in the CROPN.*

```
Input: The CROPN model;
Output: The controlled CROPN model;
Compute PPCs = {$e_1$,$e_2$, …, $e_r$}, $r > 0$ of the CROPN model;
Compute all reachable markings $R (N, M_o)$ of the CROPN model;
if the PPCs $\neq \phi$, then
 for ($1 \leq$ | PPCs | $\leq r$++), do
  if an $e_r$ is not an interactive and $p \in P(e_r)$, then
      for ($0 \leq$ | $R (N, M_o)$| $\leq k$++), do
        $K(e_r) = \Sigma K(p, e_r)$;
        $M_k(e_r) = \Sigma M_w(p, e_r)$;
        $S'(e_r) = K(e_r) - M_k(e_r)$;
        if $S'(e_r) \geq 1$, then
            The $e_r$ has no deadlock states;
        else if
            The $e_r$ has deadlock states;
            Implement the DFC-Policy in [29] to avoid deadlock states;
      end for
  else if an $e_r$ is an interactive and $p \in P(e_r)$, then
      for ($0 \leq$ | $R (N, M_o)$| $\leq j$++) do
        $K(e_r) = \Sigma K(p, e_r)$;
        $M_k(e_r) = \Sigma M_j(p, e_r)$;
        $S'(e_r) = K(e_r) - M_j(e_r)$;
        if $S'(e_r) \geq 1$ and $\eta(e_r^c, M_j) \geq 1$, then
            The $e_r$ has no deadlock states;
        else if
            The $e_r$ has deadlock states;
            Implement the condition 5 to avoid deadlock states;
            end for
        end if
    end for
else if
The $e_r$ is deadlock-free based on condition 6;
end if
Output: A controlled CROPN model
End
```

3. Hybrid CROPNs and IoT

3.1. CROPN Synthesis Based on IoT

In AMSs, the "resource failure" represents a problem with temporal uncertainty. If a "resource failure" happens, we construct a "recovery subnet" that is capable of repairing it. The resource is then reusable. Moreover, for AMS to work effectively, reliably, and safely, "fault detection and treatment" should occur rapidly [21–23,30]. In this section, formal definitions are presented in order to build "recovery", "detection", and "treatment" networks for faults in AMS using IoT. Assume the system uses sensors to detect resource failures. The sensors transmit the data to the internet. Thus, the collected data can be viewed from any location in the world. Once the data have been sent to the internet, a computer will download them and use them to find and fix failures. Thus, if one of the resources fails, online operations can continue with the remaining resources. The value measured by the sensors is used as a "set point" or "threshold value" to help the system decide how to control itself. The flowchart and structure of the implemented systems are depicted in Figure 1.

The "open systems interconnection model" (OSI) application layer communication protocols, including "dynamic host configuration protocol" (DHCP) and "hypertext transfer protocol" (HTTP), were implemented for the capture and data processing system using C++ libraries. "Transmission Control Protocol" is implemented at the transport layer. The wireless connection between the sensors and the router is performed by employing the "Wi-Fi libraries" of the capture and data processing system to detect the "service set identifier" (SSID) and password for the "Wi-Fi protected access" or "wired equivalent privacy" security protocol, which is linked to the "Wi-Fi shield" [31]. Computer software is used to transfer all data to the cloud.

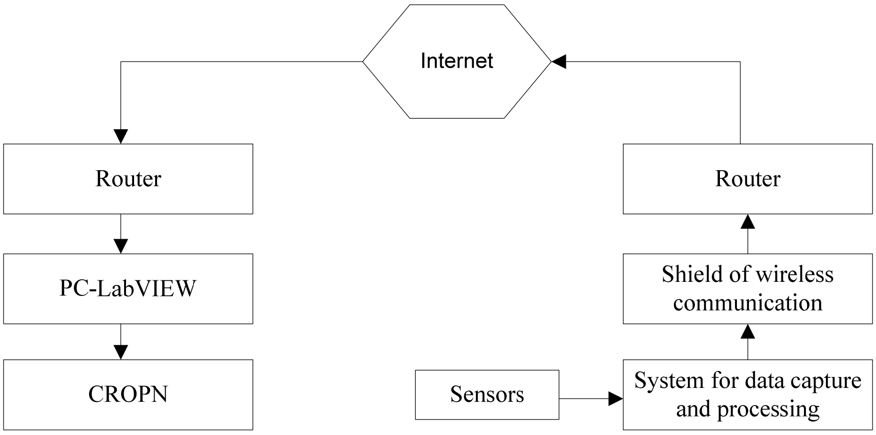
Internet
Router
Router
PC-LabVIEW
Shield of wireless communication
CROPN
Sensors
System for data capture and processing

Figure 1. Flowchart of the resource failure sensing.

Let N be a CROPN with $N = (P, T, C, I, O, K, M_0)$. Let $N_{IoT} = (\{p_i, p_{Sij}, p_{DT}, p_{RT}, p_X, p_{PC}\}, \{t_{fi}, t_{ri}, t_{DT}, t_{RT}, t_X, t_{PC}\}, F_{IoT}, c_{rni})$ be a "recovery net" of p_i based on IoT and M_{IoTo} represents initial states of the N_{IoT}, and $F_{IoT} = \{(p_i, t_{fi}), (t_{fi}, p_{Sij}), (p_{Sij}, t_{DT}), (t_{DT}, p_{DT}), (p_{DT}, t_{RT}),(t_{RT}, p_{RT}), (p_{RT},$

$t_X)$, (t_X, p_X), (p_X, t_{PC}), (t_{PC}, p_{PC}), $(p_{PC}, t_{ri})\}$, and $M_{IoTo}(p_i) \geq 0$, where p_{Sij}, $j > 0$ is the sensor's system for resource p_i failure sensing, p_{DT} represents the "data capture with wireless shield", and p_{RT} denotes the router. The sensors p_{Sj} transmit the data to the Internet (represented by p_X), namely to the servers of Xively ("a platform designed for the IoT"). The collected data can be viewed from PC-LabVIEW (represented by p_{PC}). The transitions t_{fi}, t_{ri}, t_{DT}, t_{RT}, t_X, and t_{PC} indicate a "failure transition" of the resource p_i, a "recovery transition" of the resource p_i, a transition of the data capture, a transition of the router, a transition of the data capture, the transition of the Xively, and a transition of the PC/lab view, respectively. An unreliable net is constructed if CROPN (N, M_o) is combined with the recovery net (N_{IoT}, M_{IoTo}); this is represented by the expression $(N_U, M_{Uo}) = (N_{IoT}, M_{IoTo}) \parallel (N, M_o)$, where $\parallel$ represents the integration of (N_{IoT}, M_{IoTo}) and (N, M_o).

Based on the above notations, Algorithm 2 presents the IoT-based synthesis of CROPN. Algorithm 2 can immediately observe the change in sensors using the monitoring function software of the resource failure monitoring system. When a significant failure occurred in any machine in the system, the sensors' amplitude also increased significantly. The software will provide a warning if the resource has a significant failure and the sensor data exceed the threshold. At this time, the machine tool number with the corresponding red symbol on the PC-LabVIEW represents an abnormal tool, while the green symbol denotes the tool is online and operating normally.

Algorithm 2: *Constructing CROPN based on IoT.*

```
Input: The controlled CROPN with N = (P, T, C, I, O, K, M0);
Output: The controlled CROPN based on IoT;
Insert  places $p_{DT}$, $p_{RT}$, $p_X$, $p_{PC}$ to the CROPN ;
Add  transitions $t_{DT}$, $t_{RT}$, $t_X$, $t_{PC}$ to the CROPN;
Draw arcs $(t_{DT}, p_{DT})$, $(p_{DT}, t_{RT})$, $(t_{RT}, p_{RT})$, $(p_{RT}, t_X)$, $(t_X, p_X)$, $(p_X, t_{PC})$, $(t_{PC}, p_{PC})$;
for $(1 \leq |P_R| \leq i{+}{+})$, do
    Add a sensor transition $t_{fi}$ of the resource $p_i$, $p_i \in P_R$;
    Add a recovery transition $t_{ri}$ of the resource $p_i$, $p_i \in P_R$;
    for $(1 \leq \psi_i \leq j{+}{+})$, do/* $\psi_i$ represents the number of sensors in $p_i$*/
        Add a sensor place $p_{sj}$ for resource $p_i$ failure sensing;
        Draw an arc from sensor transition $t_{fi}$ to a sensor $p_{sj}$;
        Add an arc from sensor $p_{sj}$ to a transition $t_{DT}$;
    end for
    Draw an arc from the PC-LabVIEW $p_{PC}$ to a transition $t_{ri}$;
    Add an arc from a transition $t_{ri}$ to the resource $p_i$;
end for
Output: A controlled CROPN based on IoT;
End
```

3.2. Reliability Design of IoT-CROPN

The proposed unreliable net based on IoT was changed so that the reliability parameters of the AMS could be estimated. Note that both the "unreliable net" and the method for estimating reliability described in [22,32] were employed to determine the system's reliability parameters. Let (N_U, M_{Uo}) be a CROPN based on IoT. Let $N_{RM} = (\{p_{up}, p_{down}, p_{uptime}, p_{downtime}, p_{failures}\}, \{t_{fi}, t_{ri}, t_{uptime}, t_{downtime}\}, F_{RM})$ be a " reliability model" of CROPN based on IoT and M_{RMo} represent the initial states of the N_{RM}, and $F_{RM} = \{(p_{up}, t_{fi}), (t_{fi}, p_{down}), (p_{down}, t_{ri}), (t_{ri}, p_{up}), (p_{up}, t_{uptime}), (t_{uptime}, p_{uptime}), (p_{down}, t_{downtime}), (t_{downtime}, p_{downtime})\}$, and $M_{RMo}(p_{up}) = 1$, $M_{RMio}(p_{down})$

= 0, $M_{RMo}(p_{uptime})$ =0, and $M_{RMo}(p_{downtime})$ = 0, where p_{up} is the "on condition place", p_{down} represents the "off condition place", p_{uptime} denotes the on time counter place, $p_{downtime}$ represents the downtime counter place, and $p_{failures}$ denotes the number of occurred failures in a system. The transitions t_{uptime} and $t_{downtime}$ represent an uptime deterministic transition and the downtime deterministic transition of the system, respectively. The reliability model of the CROPN based on IoT is constructed if a CROPN based on IoT (N_U, M_{Uo}) is combined with the reliability model (N_{RM}, M_{RMo}); this is represented by the expression (N_{RU}, M_{RUo}) = (N_{RM}, M_{RMo}) || (N_U, M_{Uo}).

The "failure transition" t_{fi} and "recovery transition" t_{ri} of the CROPN based on IoT presented in Section 3.2 are connected to two places: the "off condition place" p_{down} and "on condition place" p_{up} $\{(p_{up}, t_{fi}), (t_{fi}, p_{down}), (p_{down}, t_{ri}), (t_{ri}, p_{up})\}$, and the initial marking of the p_{up} = 1 and p_{down} = 0 that define the system's down and up conditions. When a "failure transition" t_{fi} is fired, a token is removed from the "p_{up}" and a token is added to the "p_{down}". The uptime and downtime of a system can be easily estimated by introducing a "time counter" expressed by a "test arc", a downtime deterministic transition "$t_{downtime}$" and an uptime deterministic transition "t_{uptime}", and two places a downtime place "$p_{downtime}$" and a uptime place "p_{uptime}" with arcs (p_{up}, t_{uptime}), (t_{uptime}, p_{uptime}), (p_{down}, $t_{downtime}$), ($t_{downtime}$, $p_{downtime}$), and the initial marking of the place p_{uptime} = 0 and the place $p_{downtime}$ = 0. In addition, a place "$p_{failures}$" is inserted in the net to calculate the number of actual failures (denoted as "$N_{failures}$"). The parameters of reliability can be determined as follows:

$$\text{“}T_{MTTF} = T_{uptime}/N_{failures}.\text{”} \tag{11}$$

$$\text{“}T_{MTTR} \quad = T_{downtime}/N_{failures}.\text{”} \tag{12}$$

$$\text{“}As = T_{uptime}/(T_{uptime+}\ T_{downtime}).\text{”} \tag{13}$$

where T_{MTTF}, T_{MTTR}, and As represent the "mean time to failure", the "mean time to repair", and the "availability" of the IoT-based unreliable CROPN, respectively.

3.3. Computational Complexity

The computation time required to execute an algorithm can be considerable, depending mainly on the complexity of the algorithm. Computer scientists have devised a method for categorizing algorithms according to the number of operations they must execute; more operations require more time. Therefore, Theorems 1 and 2 demonstrate the computational complexity of Algorithms 1 and 2, respectively.

Theorem 1. *The computational complexity of Algorithm 1 is polynomial.*

Proof: Algorithm 1 is used to show the deadlock avoidance policy for a CROPN model (N, Mo) (N, M_o) with $N = (P, T, C, I, O, K, M_o)$. Algorithm 1 contains nested loops; let y represent the number of production process circuits (PPCs) in a CROPN model, i.e., y = |PPCs|. For each state $M \in R(N, M_o)$, the liveness of each PPC will be verified using conditions 1–4; consequently, we refer to z as the CROPN reachable states, i.e., $z = |R(N, M_o)|$. The number of times "nested for loops" are conducted is yz when designing the deadlock avoidance policy for a CROPN model (N, M_o). Thus, the computational complexity of Algorithm 1 is $\boldsymbol{O}(yz)$. As the computational complexity of Algorithm 1 is polynomial, it is applicable to large-scale systems. □

Theorem 2. *The computational complexity of Algorithm 2 is polynomial.*

Proof. Algorithm 2 is used to design the CROPN-model-based IoT (N_U, M_{Uo}) with (N_U, M_{Uo}) = (N_{IoT}, M_{IoTo}) || (N, M_o), where N = (P, T, C, I, O, K, M_o) and $N_{IoT} = (\{p_i, p_{Sj}, p_{DT}, p_{RT}, p_X, p_{PC}\}, \{t_{fi}, t_{ri}, t_{DT}, t_{RT}, t_X, t_{PC}\}, F_{IoT}, c_{rni}, M_{IoTo})$. Algorithm 2 has nested loops; let β be the number of machines (resources places) in a CROPN model, i.e., β = |P_R|, where PR is the set of resource places in (N_U, M_{Uo}). For each resource place p_i, where $p_i \in P_R$, the sensors will be connected to it using steps 7–11 of Algorithm 2; thus, we define the parameter λ to be the number of sensors in resource place p_i, and this is constant for all resource places in (N_U, M_{U_0}). When designing the CROPN-model-based IoT, the number of “nested for loops” executed is $\beta\lambda$. Therefore, Algorithm 2 has a computational complexity of $\boldsymbol{O}(\beta\lambda)$, is polynomial, and is appropriate for large-scale systems. □

3.4. Illustrative Example

Consider the AMS [24] presented in Figure 2a to illustrate the CROPN construction. It consists of one CNC machine m_1 (operation resource), one robot r_1 (material handling equipment resource), and a load/unload station. The AMS produces one component type A. The AMS operation route is presented in Figure 2b.

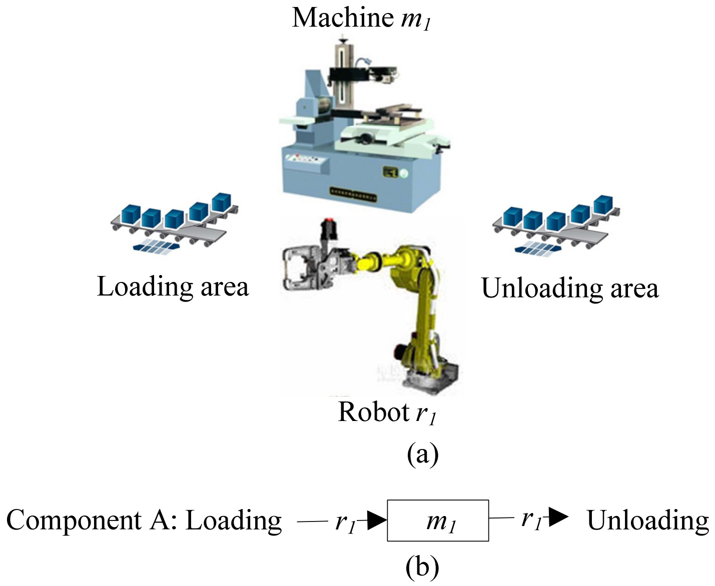

Figure 2. (**a**) Example of an AMS and (**b**) its processing route.

The developed CROPN is shown in Figure 3. As illustrated in Figure 3, places p_1 and p_r model machine m_1 and robot r_1, respectively. Transitions t_{01A} and t_{10A} model the transporting component A to/from p_1, respectively. The initial marking of the developed CROPN model is $M_o(p_o, p_1\ p_r) = M_o(c_{p1}, 0, c_{t1})$, where c_{p1} indicates that a raw component A is available at the load/unload station p_o, c_{t1} denotes robot r_1, which means that the robot r_1 is idle and can transport the raw component A, and $M_o(p_1) = 0$ means that the machine m_1 is free and available to process a raw component A. The component process route is as follows: a component A is placed at the place p_o, the robot r_1 transports a component to the machine m_1 (p_1) via a transition t_{01A}, and if the processing time of the machine m_1 is finished, the robot r_1 transports the completed component via a transition t_{10A} to the place p_0. The CROPN’s behavior is as follows: When a transition t_{01A} is enabled, it fires, and selects, respectively, from places p_o and p_r, one token c_{p1} and one token c_{t1}. If a transition t_{01A} is fired, it adds, respectively, to places p_1 and p_r, one token c_{p1} and one token c_{t1}. When the machine’s m_1 processing time is finished, a transition t_{10A} is enabled, and it selects, respectively, from places p_1 and p_r, one token c_{p1} and one token c_{t1}. If a transition t_{10A} is fired, it adds, respectively, a token c_{p1} and a token c_{t1} to the p_o and the p_r. According to Algorithm 1, the CROPN has no production process circuits that meet condition 6. Thus, it is live.

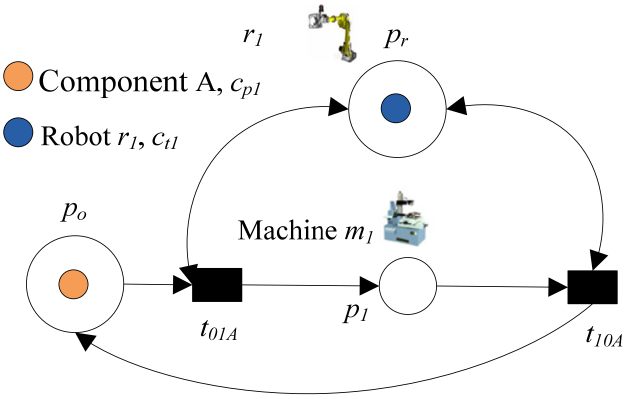

Figure 3. The CROPN of the system shown in Figure 2a.

Figure 4 illustrates the CROPN based on an IoT network for the system depicted in Figure 3 in order to demonstrate how to recognize and treat tool wear. We used a part of a mill dataset to evaluate the tool status under a single operational condition, as performed by [33]. This data set was collected through controlled laboratory experiments of milling machine operations performed using a variety of parameters, including depth of cut, feed rate, and material type. The collection of data was sampled by three different types of sensors ("acoustic emission sensor", "vibration sensor", and "current sensor") at various positions [33].

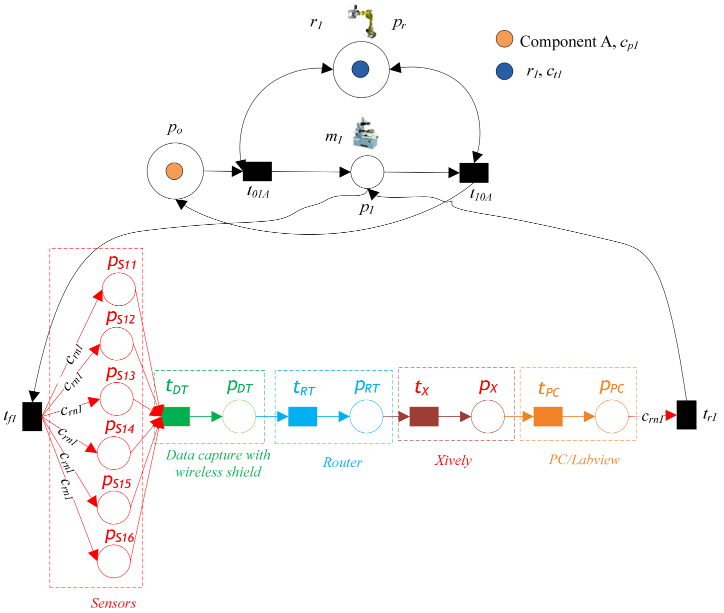

Figure 4. CROPN model based on IoT for the system presented in Figure 3.

In [33], a high-speed data acquisition board with a maximum sampling rate of 100 kHz was used to transmit the data. The output of the sampled data was utilized by the signal processing software. Multiple sensor signals were preprocessed. In most instances, the signal was amplified to meet equipment threshold requirements. Specifically, the data from the "acoustic emission sensors" and the "vibration sensors" were amplified to be within the range of ±5 V for maximum load, taking into account the equipment's maximum permitted range. The signals were screened using a "high-pass filter", and the "vibration sensor" signals were further screened using a "low-pass filter". The corner frequencies were selected based on the noise noticed on an "oscilloscope". On the oscilloscope, 180 Hz of periodic noise was observed for the vibration signal, matching the third harmonic of the primary power source. Consequently, 400 Hz was chosen as the corner frequency for the "low-pass filter". The 1 kHz "high-pass filter" was selected. The range of the "acoustic emission sensor" ends at 8 kHz. Thus, observations exceeding this frequency cannot be attributable to any machining process occurrence. Because it unnecessarily muddies the signal, it was filtered out.

As seen in Figure 4, the system uses six sensors for tool failure sensing: the current sensors p_{s1} and p_{s2}, which, respectively, gauge variations in an alternating current (AC) and a direct current (DC) spindle motor; the accelerometer sensors p_{s3} and p_{s4}, which, respectively, gauge table and spindle vibrations; and the acoustic emission sensors p_{s5} and p_{s6}, which, respectively, measure acoustic stress wave effects at the table and spindle for the diagnosis of a tool break. There are multiple phases to the process of transmitting data to the Internet. The sensors p_{s1}, p_{s2}, p_{s3}, p_{s4}, p_{s5}, and p_{s6} first take the readings and send them to the "Wi-Fi wireless shield" (represented by t_{DT} and p_{DT}). The shield then transfers the data to a "wireless router" (denoted by t_{RT} and p_{RT}). Integrating objects into a communication network is an important component of the IoT. In other words, it symbolizes a new method in which all objects are integrated into the Internet and interact in real time. Practically, it is built on the integration of sensors, equipment, and "household" items with the internet via "wired or wireless networks". The use of this method is achievable, and its implementation will be cost-effective due to the Internet's global reach. It will enable the "sensors" to be connected in "homes", "workplaces", and various "automation operations". Thus, every object can be linked to a web environment in order to save all of its data and display it in the actual world. The objective of this research is to introduce the PC-LabVIEW-based system designed to communicate Internet data to the CROPN. Using a PC equipped with Labview (represnted by t_{PC} and p_{PC}), the cloud-stored data is downloaded for processing and transmission to the CROPN. Through the OPC communication standard, the CROPN is responsible for executing the PC-LabVIEW-programmed fault detection and treatment control (communication between the CROPN and PC-LabVIEW). Figure 5 shows the reliability design of the net illustrated in Figure 4.

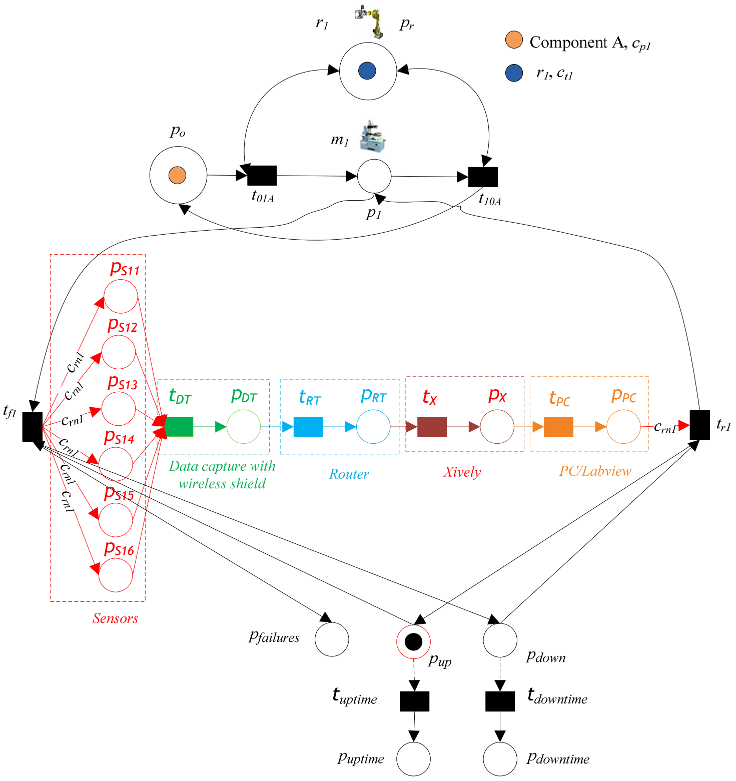

Figure 5. Reliability design of the net illustrated in Figure 4.

The system depicted in Figure 6 is the "graphical user interface" (GUI) that enables communication with the sensor system and presents the collected information. Consequently, it supervises and monitors the operation. If the sensor data exceed the threshold, the CROPN executes the system's action.

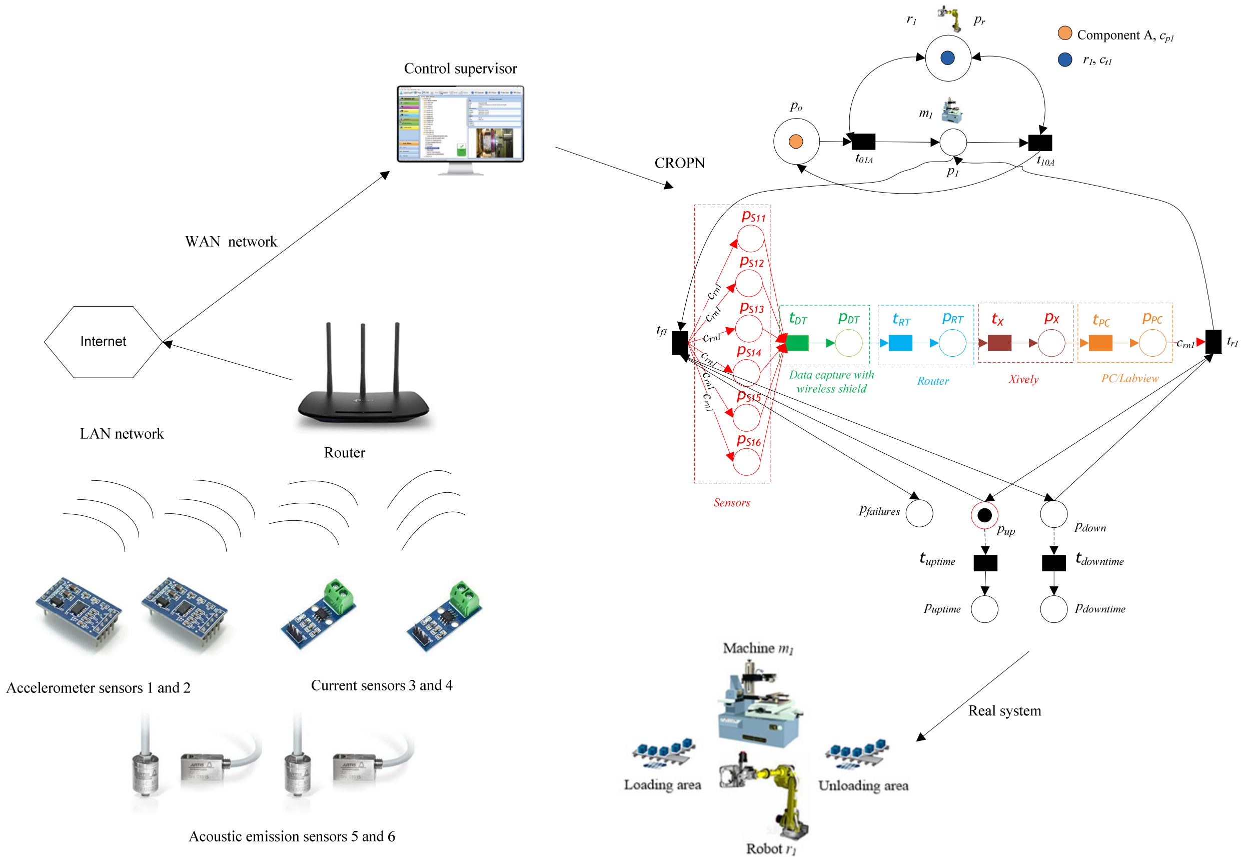

Figure 6. Graphical user interface based on IoT for an AMS presented in Figure 4.

The information gathered by the "sensors" and sent to the Internet is depicted in Figure 7. It enables the filtering and graphing of data to identify potential patterns or important events. Regardless of the exact location of the sensors, the data can be accessed from anywhere on Earth.

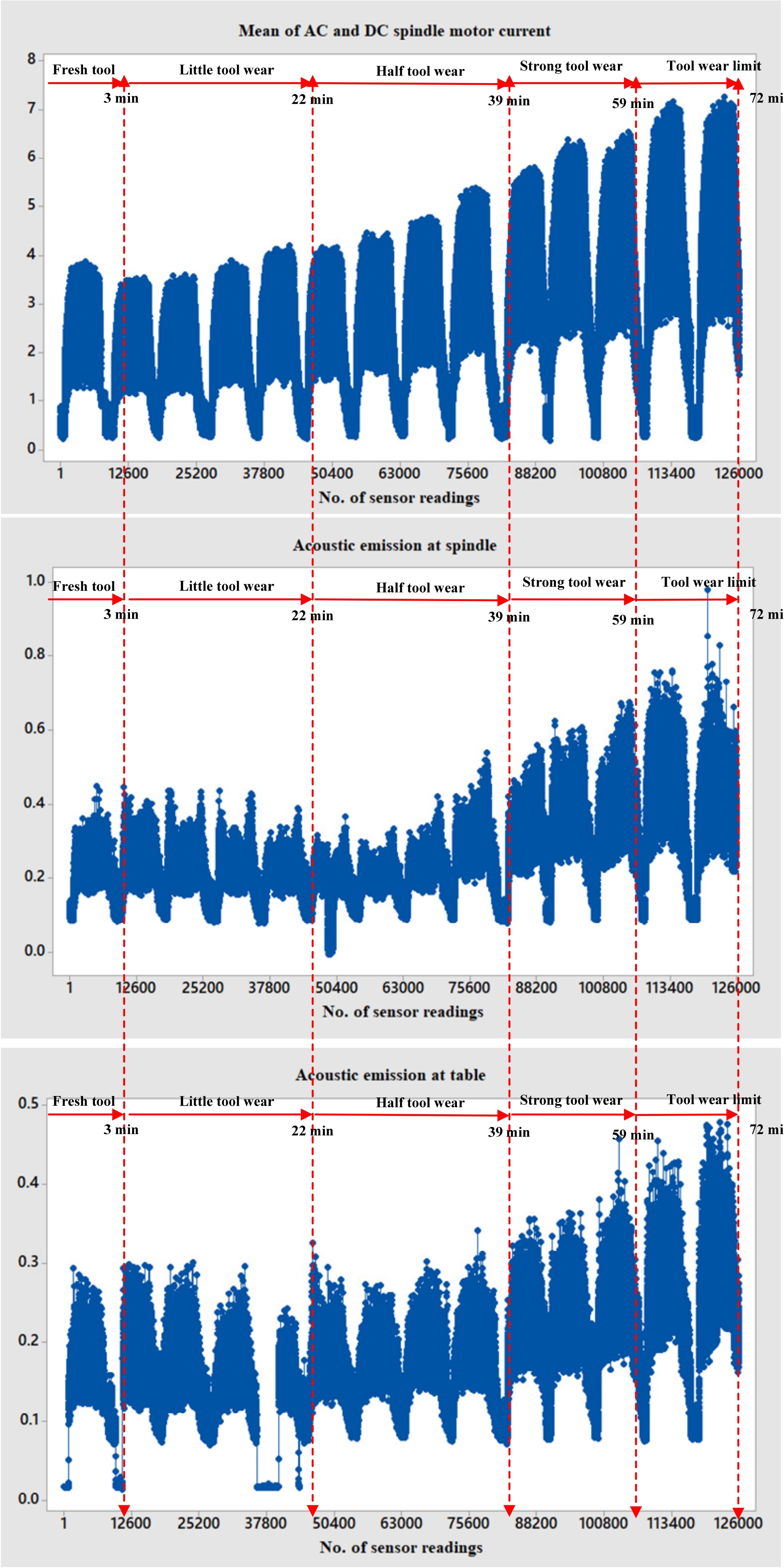

Figure 7. Sensor-gathered data for monitoring tool wear status for the GUI shown in Figure 6.

4. Experimental Results

This section uses an existing automated manufacturing system in [21,24] to describe the implemented sensor system. The system is illustrated in Figure 8. It includes three machines, m_1–m_3 (resources for operations); two robots, r_1 and r_2 (resources for material handling equipment); and a load/unload station. The AMS produces two component types, A and B. The experimental conditions and parameters were determined by industry applicability and manufacturer-recommended settings. The conditions and parameters of the system were as follows: For machine 1, the cutting speed was set to 200 m/min, which corresponds to 826 rev/min, the depth of cut was 1.5 mm, and the feed rate was 0.5 mm/rev, which translates to 413 mm/min. For machine 2, the cutting speed was set to 200 m/min, the depth of cut was 0.75 mm, and the feed rate was 0.25 mm/rev, which translates to 206.5 mm/min. Cast iron (component A) and stainless steel J45 (component B) were used as the materials. The collection of data was sampled by three different types of sensors (“acoustic emission sensor”, “vibration sensor”, and “current sensor”) at various positions.

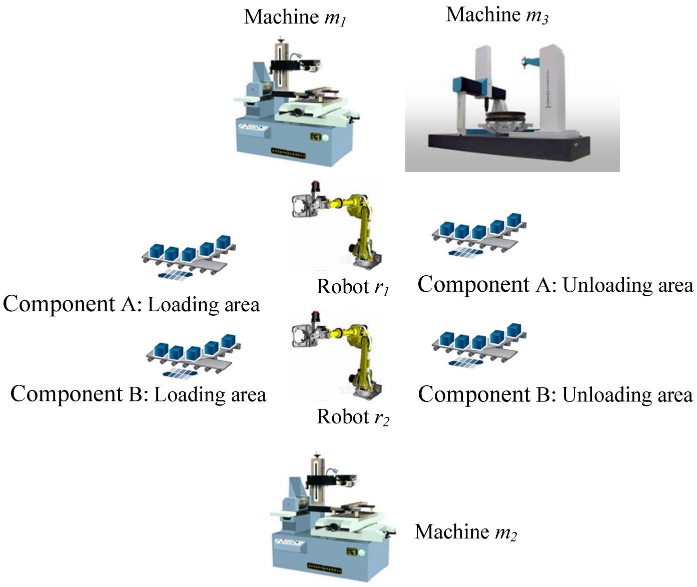
Machine m_1
Machine m_3
Robot r_1
Component A: Loading area
Component A: Unloading area
Component B: Loading area
Component B: Unloading area
Robot r_2
Machine m_2

Figure 8. The automated manufacturing system in [21,24].

The description of places, transitions, and colors in the CROPN model presented in Figure 9 is as follows: Places p_1, p_2, p_r, and p_0 represent a machine m_1, a machine m_2, a transportation place for robots r_1 and r_2, and the loading and unloading station, respectively. Transitions t_{01A}, t_{12A}, t_{20A}, t_{01B}, t_{13B}, t_{31B}, and t_{30B} model, a transporting of the component A from the p_0 to the machine m_1 by robots r_1, a transporting of the component A from the machine m_1 to the machine m_2 by robots r_1, a transporting of the component A from the machine m_2 to the p_0 by robots r_1, a transporting of the component B from the p_0 to the machine m_1 by robots r_2, a transporting of the component B from the machine m_1 to the machine m_3 by robots r_2, a transporting of the component B from the machine m_3 to the machine m_1 by robots r_2, and a transporting of the component B from the machine m_3 to the p_0 by robots r_2, respectively. The colored tokens c_{p1}, c_{p2}, c_{t1}, and c_{t2} represent component A, a component B, a robot r_1, and a robot r_2, respectively.

The initial states of the developed CROPN are $M_o(p_o, p_1, p_2, p_3, p_r) = M_o(\{c_{p1}, c_{p2}\}, 0, 0, 0, \{c_{t1}, c_{t2}\})$, where c_{p1} and c_{p2} indicate, respectively, that raw components A and B are available at load/unload station p_o; c_{t1} and c_{t2} denote, respectively, robots r_1 and r_1, which mean that the robot r_1 is idle and can transport the raw component A and the robot r_2 is idle and can transport the raw component B; and $M_o(p_1) = M_o(p_2) = M_o(p_3) = 0$, which mean that the machines m_1–m_3 are free and available to process raw components. In addition, the following describes the process routes of the components A and B:

(1) Component A is available in the place p_o and the robot r_1 transports it to the place p_1 (m_1) via a transition t_{01A}. If the processing time of the m_1 is finished, the r_1 transports the component from m_1 to the place p_2 (m_2) via a transition t_{12A}. If the processing time of the m_2 is finished, the robot r_1 transports the completed component via a transition t_{20A} to the place p_o.

(2) Component B is available in the place p_o and the robot r_2 transports it to the place p_1 (m_1) via a transition t_{01B}. If the processing time of the m_1 is finished, the r_2 transports the component from to the place p_3 (m_3) via a transition t_{13B}, when the component has no defects, the r_2 places the completed component via a transition t_{30B} to the place p_o; otherwise the component returns to the place p_1 (m_1) via a transition t_{31B}, then to a transition t_{13B}, a place p_3, a transition t_{30B}, and a place p_o.

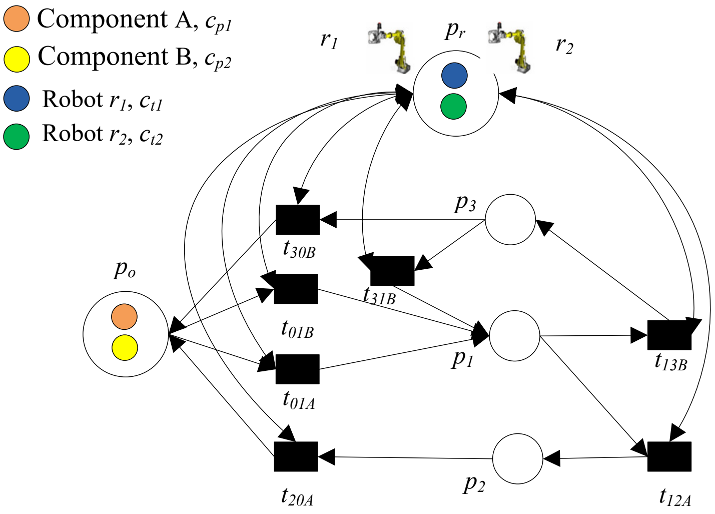

Figure 9. CROPN model of the system presented in Figure 8.

To determine if the developed CROPN is live, reconsider the net presented in Figure 9. Figure 10 depicts its reachability markings and can be stated as

$$M_0 = (\{c_{p1}, c_{p2}\}, 0, 0, 0, \{c_{t1}, c_{t2}\}).$$

$$M_1 = (c_{p2}, c_{p1}, 0, 0, \{c_{t1}, c_{t2}\}).$$

$$M_2 = (c_{p1}, c_{p2}, 0, 0, \{c_{t1}, c_{t2}\}).$$

$$M_3 = (c_{p2}, 0, c_{p1}, 0, \{c_{t1}, c_{t2}\}).$$

$$M_4 = (c_{p1}, 0, 0, c_{p2}, \{c_{t1}, c_{t2}\}).$$

The CROPN consists of a PPC: $e_1 = \{p_1, t_{13B}, p_3, t_{31B}\}$. We assume that p_1, p_2, and p_3 each have a capacity of one. The control condition 6 stated in Equation (8) is employed to determine the number of available spaces in an e_1, which are displayed in Table 1. As illustrated in Table 1, the condition 6 was achieved for all markings displayed in Figure 10, and the subnet e_1 was live. Thus, we can concluded that the developed CROPN was live.

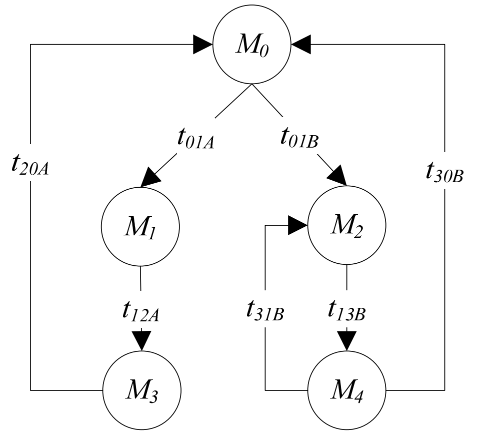

Figure 10. Reachability states of the CROPN presented in Figure 9.

Table 1. The free spaces in an e_1.

State	$K(e_1) = \Sigma K(p_i, e_1)$	$M(e_1) = \Sigma M_k(p_i, e_1)$	$S'(e_1) = K(e_1) - M(e_1)$
0	2	0	2 ≥ 1
1	2	1	1 ≥ 1
2	2	1	1 ≥ 1
3	2	0	2 ≥ 1
4	2	1	1 ≥ 1

Next, Figure 11 illustrates the IoT-based CROPN and reliability model for the system depicted in Figure 9 in order to demonstrate how to recognize and treat faults. As seen in Figure 11, we have the following nomenclature: Places p_{s11} and p_{s21} represent the “current sensors”, which gauge changes in an AC of the spindle motor for machines m_1 and m_2, respectively; p_{s12} and p_{s22} model the “current sensors”, which gauge changes in a DC of the spindle motor for machines m_1 and m_2, respectively; p_{s13} and p_{s23} represent accelerometer sensors that gauge table vibration for machines m_1 and m_2, respectively; p_{s14} and p_{s24} model the “accelerometer sensors” that gauge spindle vibration for machines m_1 and m_2, respectively; p_{s15} and p_{s25} represent the “acoustic emission sensors” that gauge “acoustic stress wave” impacts at the table for machines m_1 and m_2, respectively; p_{s16} and p_{s26} model the “acoustic emission sensors” that gauge “acoustic stress wave” impacts at the spindle for machines m_1 and m_2, respectively. In addition, places p_{DT}, p_{RT}, p_X, p_{PC}, p_{up}, p_{down}, p_{uptime}, $p_{downtime}$, and $p_{failures}$ represent the “data capture with wireless shield,” the router, the transmitting the data from sensors p_{Sij} to the Internet, the PC-LabVIEW, the “on condition place” for machines m_1 and m_2, the “on condition place” for machines m_1 and m_2, the “off condition place” for machines m_1 and m_2, the “uptime place” for machines m_1 and m_2, the “downtime place” for machines m_1 and m_2, and number of actual failures in machines m_1 and m_2, respectively. Finally, transitions t_{f1}, t_{f2}, t_{r1}, t_{r2}, t_{DT}, t_{RT}, t_X, t_{PC}, $t_{downtime}$, and t_{uptime} represent the failure of the machine m_1, the failure of the machine m_2, the recovery of the machine m_1, the recovery of the machine m_2, the data capture, the router, the Xively, the PC-LabVIEW, the downtime deterministic for machines m_1 and m_2, and the uptime deterministic for machines m_1 and m_2, respectively.

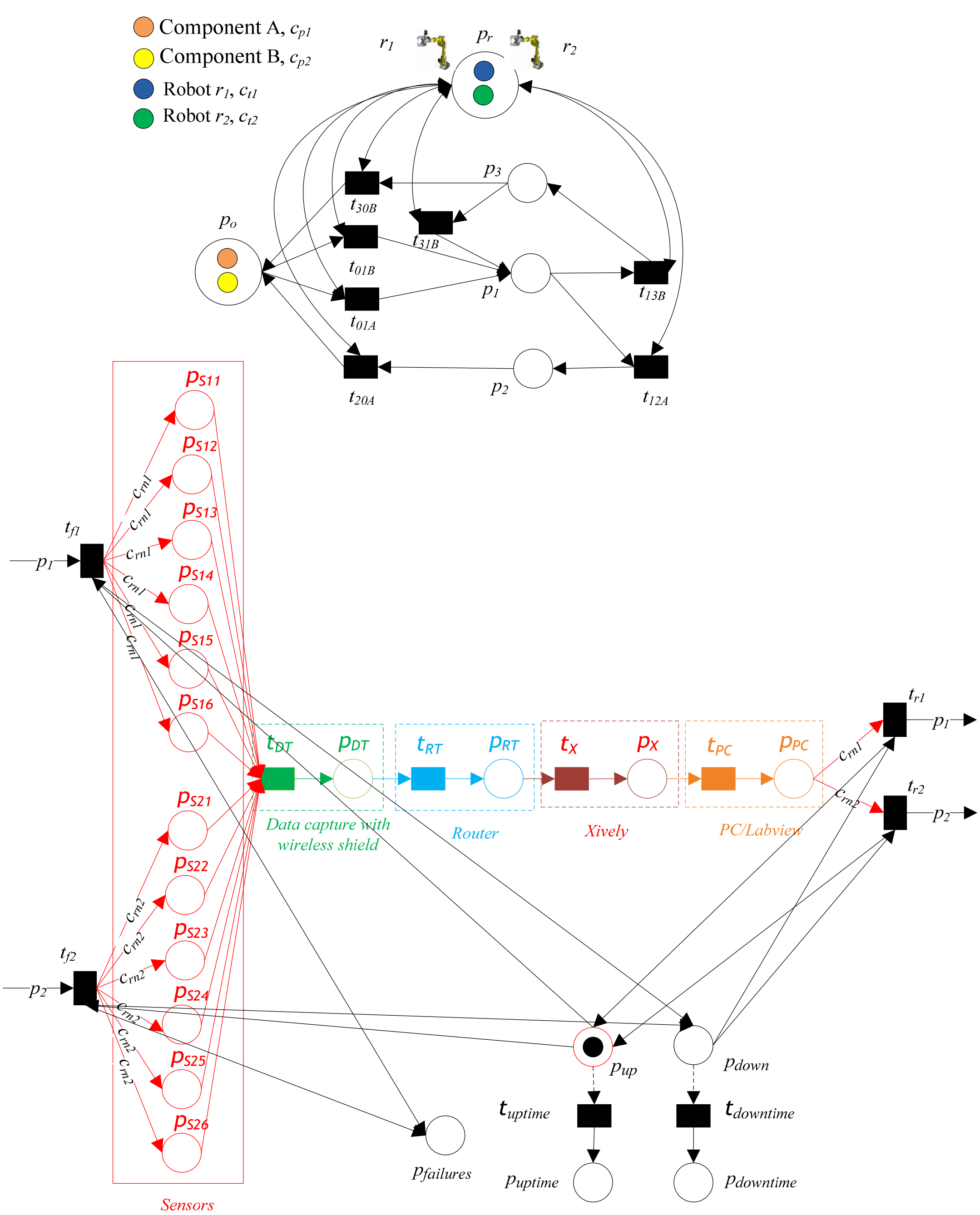

Figure 11. The unreliable CROPN based on IoT and reliability model for the model shown in Figure 9.

5. Performance Evaluation

Finally, a MATLAB-based GPenSIM tool [20,24,30,34,35] was used to execute verification and validation of the constructed IoT-based CROPN and the reliability model presented in Figure 11. The GPenSIM tool was developed in order to simulate and evaluate AMS. Using MATLAB R2015a, the developed code for the reliability model was executed. The length of the simulation (denoted as $T_{Simulation}$) was 450 min. Table 2 provides data on machine maintenance for the system depicted in Figure 10. Moreover, Table 3 summarizes the result of the simulation, which contains the total components that go through the time of simulation (denoted as $N_{Throughput}$), the throughput time for component (represented by $T_{Throughput}$), which can be expressed as

$$T_{Throughput} = \frac{T_{Simulation}}{N_{Throughput}}, \quad (14)$$

and the utilization of the machine, that can be formulated as

$$U_{Machine} = \left(\frac{T_{Machine}}{T_{Simulation}}\right) \cdot 100 \quad (15)$$

where $U_{Machine}$ and $T_{Machine}$ represent the utilization of the machine and the total occupation time of the machine, respectively.

Table 2. Data of the machine maintenance for the model illustrated in Figure 11.

Machine	**Processing Time (min)**		T_{MTTF} **(min)**		T_{MTTR} **(min)**	
	Com. A	**Com. B**	**Without Sensor System**	**With Sensor System**	t_{t1}	t_{t2}
m_1	5	7	74	Based on sensor system	10	-
m_2	6		84		-	13
m_3	-	2	-		-	-

Table 3. Comparison of the proposed IoT-CROPN shown in Figure 11 with existing methods.

	Performance						
	Throughput (Components)		**Throughput Time (min/Component)**		**Utilization (%)**		
	Com. A	**Com. B**	**Com. A**	**Com. B**	m_1	m_2	m_3
Al-Shayea *et al*. [30]	26	27	17.31	16.67	76.23	34.98	13.45
Kaid *et al*. [22]	28	29	16.07	15.52	81.61	37.67	14.35
Al-Ahmari *et al*. [23]	27	28	16.67	16.07	80.94	36.77	13.90
Alzalab *et al*. [36]	29	29	15.52	15.52	83.56	38.06	14.19
IoT-CROPN	29	30	15.52	15	84.68	39.19	14.86

As depicted in Figure 12, the results of these experiments were compared with [22,23,30,36], where the same conditions and parameters were used for all studies. In terms of "throughput", "throughput time", and "utilization", the results show that the IoT-based CROPN outperformed the other approaches. Moreover, Table 4 contains the reliability metrics of the IoT-based CROPN: "uptime", "downtime", and "availability". The availability of the model was observed to be 88.22%, outperforming two existing approaches.

Table 4. The parameters of the reliability for the IoT-CROPN shown in Figure 11.

	Reliability Parameter		
	T_{uptime} **(min)**	$T_{downtime}$ **(min)**	As **(%)**
Al-Shayea et al. [30]	347	103	77.11
Kaid et al. [22]	372	78	82.67
Al-Ahmari et al. [23]	384	66	85.53
Alzalab et al. [36]	394	56	87.55
IoT-CROPN	397	53	88.22

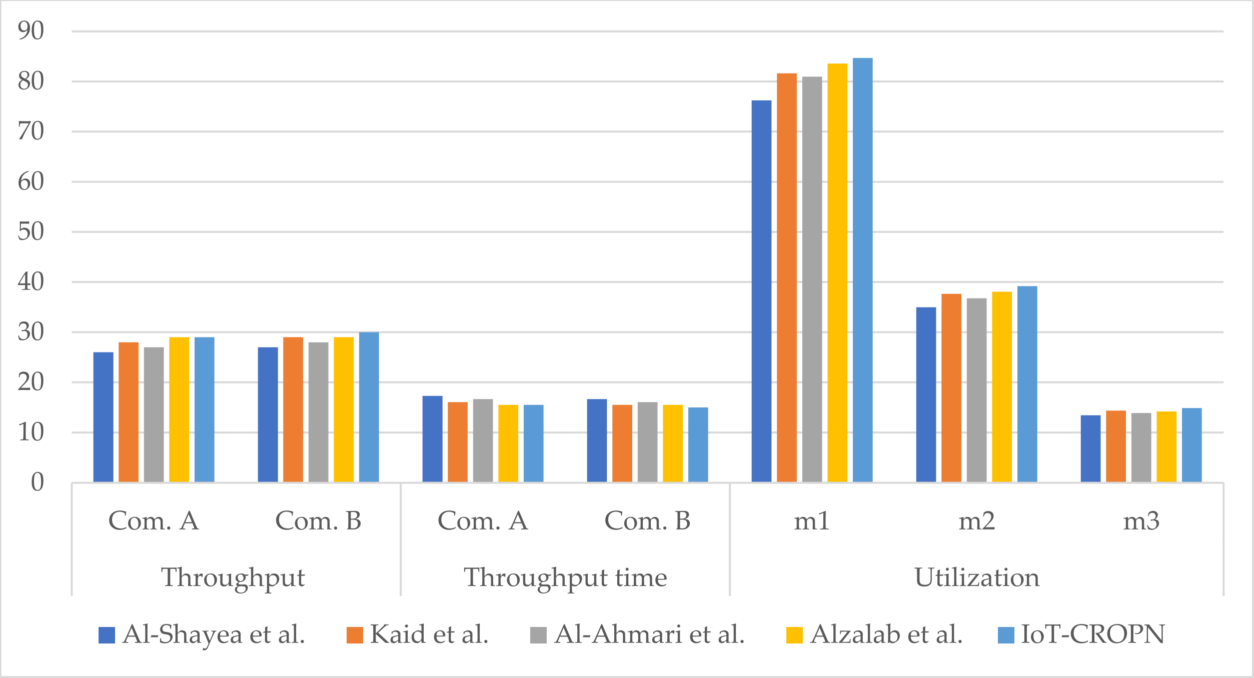

Figure 12. Comparison of the proposed IoT-CROPN shown in Figure 11 with the existing methods.

6. Conclusions

In this paper, we emphasize the significance of techniques, tools, and procedures designed to support automatic threat (i.e., fault, error, and failure) diagnosis and recovery due to the immense interest and development of IoT-based cyber-physical systems in AMS, where a certain degree of adaptability is crucial. We propose an IoT-based CROPN to facilitate self-detection and self-treatment of failures and to obtain essential reliability measurements in the IoT-based CROPN, such as "uptime", "downtime", and "availability". Existing system examples are used to demonstrate the effectiveness of the developed technique. First, the tool wear condition of a CNC milling machine was evaluated using an IoT-based CROPN that provided a systematic classification of tool wear status. This technique can lead to more effective monitoring of tool replacement due to a more effective evaluation of tool wear. First, "sufficient and necessary conditions" were presented for the liveness of the CROPN. Second, we developed a "fault diagnosis and treatment" method, which combines the obtained CROPN with IoT to guarantee the reliability of the CROPN. A simulation tool was used to verify, validate, and analyze the reliability of the IoT-based CROPN, and the results were compared with those reported in the literature.

The advantages of the IoT-based CROPN are as follows:

(1) Compared to the research in [22,23,30,36], it has been demonstrated to be simpler in configuration and more effective at overcoming the deadlock problem, "fault diagnosis and treatment", and designing CROPN reliability.
(2) It provides a hybrid model that is simultaneously simulated in real time and on the plant floor.
(3) It was experimentally validated to demonstrate how to evaluate performance and monitor tool status on various machines.

Future research will address other discrete variables, such as product quality, overall energy utilization, reconfiguration of an AMS, and product rescheduling. In addition, operations on an "intelligent vehicle highway system" (IVHS) are highly non-linear and subject to rapid environmental change. In order to guarantee safe travel for humans, accurate diagnosis and reliable sensor data are required. Therefore, an internet-of-things-based CROPN can be employed to improve IVHS capacity and safety on the highway. Finally, in some occurrences of multiple failures, symptoms may be mutually exclusive. In large-scale problems, it may not be as simple to recognize and model these occurrences that deviate from the default model. Thus, a technique for capturing these occurrences should be designed.

Author Contributions: Conceptualization, H.K. and A.A.-A.; methodology, H.K., A.A.-A. and K.N.A.; software, H.K. and A.A.-A.; validation, H.K., A.A.-A. and K.N.A.; resources, H.K. and A.A.-A.; data curation, H.K. and A.A.-A.; writing—original draft preparation, H.K. and A.A.-A.; writing—review and editing, H.K., A.A.-A. and K.N.A.; supervision, A.A.-A.; project administration, A.A.-A.; funding acquisition, A.A.-A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are grateful to the Raytheon Chair for Systems Engineering for funding.

Data Availability Statement: All the data are available in the manuscript.

Acknowledgments: The authors would like to thank the Raytheon Chair for Systems Engineering for funding.

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

References

1. Syed, A.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 429–475. https://doi.org/10.3390/smartcities4020024.
2. Cardin, O. Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Comput. Ind.* **2019**, *104*, 11–21. https://doi.org/10.1016/j.compind.2018.10.002.
3. Hsieh, F.-S. Temporal Analysis of Influence of Resource Failures on Cyber-Physical Systems Based on Discrete Timed Petri Nets. *Appl. Sci.* **2021**, *11*, 6469. https://doi.org/10.3390/app11146469.
4. Lee, E.A. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors* **2015**, *15*, 4837–4869. https://doi.org/10.3390/s150304837.
5. Tran, M.-Q.; Doan, H.-P.; Vu, V.Q.; Vu, L.T. Machine Learning and IoT-based Approach for Tool Condition Monitoring: A Review and Future Prospects. *Measurement* **2022**, 112351. doi.org/10.1016/j.measurement.2022.112351
6. Civerchia, F.; Bocchino, S.; Salvadori, C.; Rossi, E.; Maggiani, L.; Petracca, M. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. *J. Ind. Inf. Integr.* **2017**, *7*, 4–12. https://doi.org/10.1016/j.jii.2017.02.003.
7. Tran, M.-Q.; Elsisi, M.; Mahmoud, K.; Liu, M.-K.; Lehtonen, M.; Darwish, M.M.F. Experimental Setup for Online Fault Diagnosis of Induction Machines via Promising IoT and Machine Learning: Towards Industry 4.0 Empowerment. *IEEE Access* **2021**, *9*, 115429–115441. https://doi.org/10.1109/access.2021.3105297.
8. Tran, M.-Q.; Elsisi, M.; Liu, M.-K.; Vu, V.Q.; Mahmoud, K.; Darwish, M.M.; Abdelaziz, A.Y.; Lehtonen, M. Reliable Deep Learning and IoT-Based Monitoring System for Secure Computer Numerical Control Machines Against Cyber-Attacks With Experimental Verification. *IEEE Access* **2022**, *10*, 23186–23197.
9. Wang, Y.; Zheng, L.; Wang, Y. Event-driven tool condition monitoring methodology considering tool life prediction based on industrial internet. *J. Manuf. Syst.* **2021**, *58*, 205–222.
10. Stuhr, B.; Liu, R. A Flexible Similarity-Based Algorithm for Tool Condition Monitoring. *J. Manuf. Sci. Eng.* **2022**, *144*, 031010. https://doi.org/10.1115/1.4051885.
11. Pimenov, D.Y.; Gupta, M.K.; da Silva, L.R.; Kiran, M.; Khanna, N.; Krolczyk, G.M. Application of measurement systems in tool condition monitoring of Milling: A review of measurement science approach. *Measurement* **2022**, *199*, 111503. https://doi.org/10.1016/j.measurement.2022.111503.
12. Ahmad, I.; Hee, L.M.; Abdelrhman, A.M.; Imam, S.A.; Leong, M. Scopes, challenges and approaches of energy harvesting for wireless sensor nodes in machine condition monitoring systems: A review. *Measurement* **2021**, *183*, 109856. https://doi.org/10.1016/j.measurement.2021.109856.
13. Ji, Y.; Wang, R.; Jiang, L. IoT service modeling and description based on XML and Petri-Net. *Adv. Inf. Sci. Serv. Sci.* **2013**, *5*, 271.
14. Zhang, F.; Xu, Y.; Chou, J. A Novel Petri Nets-Based Modeling Method for the Interaction between the Sensor and the Geographic Environment in Emerging Sensor Networks. *Sensors* **2016**, *16*, 1571. https://doi.org/10.3390/s16101571.
15. Zairi, S.; Mezni, A.; Zouari, B. Formal approach for modeling, verification and performance analysis of wireless sensors network. In Proceedings of the International Conference on Wired/Wireless Internet Communication, Malaga, Spain, 25–27 May 2015; pp. 381–395.
16. Cambronero, M.E.; Macia, H.; Valero, V.; Orozco-Barbosa, L. Modeling and Analysis of the 1-Wire Communication Protocol Using Timed Colored Petri Nets. *IEEE Access* **2018**, *6*, 27356–27372. https://doi.org/10.1109/access.2018.2833213.
17. Rangwala, S.; Dornfeld, D. Learning and optimization of machining operations using computing abilities of neural networks. *IEEE Trans. Syst. Man, Cybern.* **1989**, *19*, 299–314. https://doi.org/10.1109/21.31035.
18. Dornfeld, D. Acoustic emission process monitoring for untended manufacturing. In *Proceedings of the Proc. Japan-USA Symposium on Flexible Automation*; 1986; pp. 831–836.

19. Agogino, A.M.; Srinivas, S.; Schneider, K.M. Multiple sensor expert system for diagnostic reasoning, monitoring and control of mechanical systems. *Mech. Syst. Signal Process.* **1988**, *2*, 165–185. https://doi.org/10.1016/0888-3270(88)90041-6.
20. Davidrajuh, R. *Modeling Discrete-Event Systems with GPenSIM: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2018.
21. Kaid, H.; Al-Ahmari, A.; Li, Z.; Ameen, W. Deadlock Control and Fault Detection and Treatment in Reconfigurable Manufacturing Systems Using Colored Resource-Oriented Petri Nets Based on Neural Network. *IEEE Access* **2021**, *9*, 84932–84947. https://doi.org/10.1109/access.2021.3084995.
22. Al-Shayea, A.; Kaid, H.; Al-Ahmari, A.; Nasr, E.A.; Kamrani, A.K.; Mahmoud, H.A. Colored Resource-Oriented Petri Nets for Deadlock Control and Reliability Design of Automated Manufacturing Systems. *IEEE Access* **2021**, *9*, 125616–125627. https://doi.org/10.1109/access.2021.3111575.
23. Kaid, H.; Al-Ahmari, A.; Nasr, E.A.; Al-Shayea, A.; Kamrani, A.K.; Noman, M.A.; Mahmoud, H.A. Petri Net Model Based on Neural Network for Deadlock Control and Fault Detection and Treatment in Automated Manufacturing Systems. *IEEE Access* **2020**, *8*, 103219–103235. https://doi.org/10.1109/access.2020.2999054.
24. Kaid, H.; Al-Ahmari, A.; Li, Z. Colored Resource-Oriented Petri Net Based Ladder Diagrams for PLC Implementation in Reconfigurable Manufacturing Systems. *IEEE Access* **2020**, *8*, 217573–217591. https://doi.org/10.1109/access.2020.3041408.
25. Wu, N. Avoiding deadlocks in automated manufacturing systems with shared material handling system. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 20–25 April 1997; pp. 2427–2432.
26. Wu, N.; Zhou, M.; Li, Z. Resource-Oriented Petri Net for Deadlock Avoidance in Flexible Assembly Systems. *IEEE Trans. Syst. Man, Cybern. Part A: Syst. Humans* **2008**, *38*, 56–69. https://doi.org/10.1109/tsmca.2007.909542.
27. Wu, N.; Zhou, M. Process vs resource-oriented Petri net modeling of automated manufacturing systems. *Asian J. Control.* **2010**, *12*, 267–280. https://doi.org/10.1002/asjc.184.
28. Wu, N.; Zhou, M. *System Modeling and Control with Resource-Oriented Petri Nets*; CRC Press, Taylor & Francis Group: New York, NY, USA, 2009.
29. Wu, N.; Zhou, M. Intelligent token Petri nets for modelling and control of reconfigurable automated manufacturing systems with dynamical changes. *Trans. Inst. Meas. Control.* **2011**, *33*, 9–29. https://doi.org/10.1177/0142331208095622.
30. Al-Ahmari, A.; Kaid, H.; Li, Z.; Davidrajuh, R. Strict Minimal Siphon-Based Colored Petri Net Supervisor Synthesis for Automated Manufacturing Systems With Unreliable Resources. *IEEE Access* **2020**, *8*, 22411–22424. https://doi.org/10.1109/access.2020.2968469.
31. Corso, F.; Camargo, Y.;Ramirez, L. Wireless sensor system according to the concept of IoT-internet of things. In Proceedings of the 2014 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 10–13 March 2014; pp. 52–58.
32. Bertolini, M.; Bevilacqua, M.; Mason, G. Reliability design of industrial plants using Petri nets. *J. Qual. Maint. Eng.* **2006**, *12*, 397–411. https://doi.org/10.1108/13552510610705955.
33. Agogino, A.; Goebel, K. Mill Data Set. BEST lab, UC Berkeley. NASA Ames Prognostics Data Repository. NASA Ames, Moffett Field, CA, USA, 2007. Available online: http://ti.arc.nasa.gov/project/prognostic-data-repository (accessed 1 December 2022).
34. Kaid, H.; Al-Ahmari, A.; Li, Z.; Davidrajuh, R. Single Controller-Based Colored Petri Nets for Deadlock Control in Automated Manufacturing Systems. *Processes* **2020**, *8*, 21. https://doi.org/10.3390/pr8010021.
35. Kaid, H.; Al-Ahmari, A.; Li, Z.; Davidrajuh, R. Intelligent Colored Token Petri Nets for Modeling, Control, and Validation of Dynamic Changes in Reconfigurable Manufacturing Systems. *Processes* **2020**, *8*, 358. https://doi.org/10.3390/pr8030358.
36. Alzalab, E.A.; El-Sherbeeny, A.M.; El-Meligy, M.A.; Rauf, H.T. Trust-Based Petri Net Model for Fault Detection and Treatment in Automated Manufacturing Systems. *IEEE Access* **2021**, *9*, 157997–158009. https://doi.org/10.1109/access.2021.3128206.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.