Petri Net Modeling and Wafer Sojourn Time Analysis of Single-Arm Cluster Tools with Residency Time Constraint and Activity Time Variation

Yan Qiao and NaiQi Wu
Department of Industrial Engineering
School of Mechatronics Engineering
Guangdong University of Technology
Guangzhou 510006, P. R. CHINA
nqwu@gdut.edu.cn

MengChu Zhou
Dept. of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102-1982, USA
Zhou@njit.edu

Abstract: Cluster tools are increasingly adopted in semiconductor fabrication. With wafer residency time constraints, it is crucial to schedule a cluster tool such that the wafer sojourn time in a processing module is in a given range. However, because of the activity time variation in wafer fabrication by cluster tools, a feasible schedule obtained under the assumption of deterministic activity times may become infeasible. Thus, it is a great challenge to schedule cluster tools. To solve this problem, it is critically important to reveal the wafer sojourn time change with bounded activity time variation. This work targets at single-arm cluster tools. They are modeled by a Petri net (PN) to describe the fabrication processes. Based on the net, a real-time control policy is proposed such that its use offsets the effect of the activity time variation as much as possible. Then, the wafer sojourn time delay is analyzed and analytical expressions are given to compute the upper bound. With the proposed method, we can check if a given schedule is feasible under bounded activity time variation. Examples are given to show the applications of the research results.

Keywords: Cluster tools, Semiconductor manufacturing, Petri net, scheduling.

I. INTRODUCTION

As a kind of integrated equipment in semiconductor manufacturing, cluster tools are increasingly adopted to produce wafers one by one with single-wafer processing technology. A cluster tool is composed of a number of processing modules (PM), an aligner, a wafer handing robot, and two loadlocks for wafer cassette loading/unloading. They provide a flexible, reconfigurable, and efficient environment for semiconductor manufacturing [Bader, et al., 1990; and Burggraaf, 1995], resulting in higher yield [Newboe, 1990], shorter cycle time [McNab, 1990; Newboe, 1990; and Singer, 1995], better utilization of space [Burggraaf, 1995; and Singer, 1995], and lower capital cost [Singer, 1995].

In general, a cassette has 25 wafers with an identical recipe [Kim et al., 2003; and Lee and Park, 2005]. Raw wafers are loaded into the system through a loadlock, visit one or more PMs in a pre-specified order (each wafer should stay in a PM for a minimum time in each step to get processed), and return to the loadlock where they come from [Wu et al., 2008]. The robot can be a single or dual-arm one as shown in Fig. 1.1.

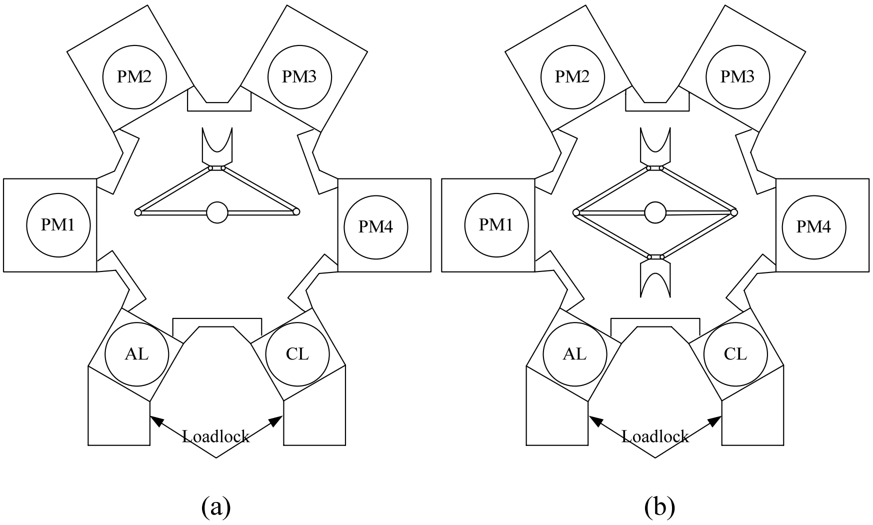

Figure 1.1. The cluster tools: (a) single-arm robot; (b) dual-arm robot

To operate cluster tools effectively, important effort has been made in modeling and performance evaluation of cluster tools [Venkatesh et al., 1997; Ding et al., 2006; Perkinson, et al., 1994 and 1996; Yi et al., 2008; Zuberek, 2001; Chan et al., 2011; and Wu and Zhou, 2010a and 2010b]. With these models, it is found that the operations of a cluster tool under the steady state are divided into two different regions: transport and process-bound ones. In the former, the robot is always busy and the system cycle time is determined by the time for robot tasks, while in the latter, the robot has idle time and the processing times in PMs determine the cycle time. It is also shown that PM activities follow the robot tasks [Shin et al., 2001]. Hence, scheduling robot tasks is very important. With these properties, dispatching or priority rules are developed to schedule them [Venkatesh et al., 1997; and Jevtic, 1999]. It is known that, for cluster tools, the robot moving time from one PM to another can be treated as a constant and are much shorter than the wafer processing time [Kim et al., 2003]. Thus, for single-arm cluster tools, backward scheduling is optimal [Lee et al., 2004; and Lopez and Wood, 2003]. However, these results are obtained based on the assumption that there is no limitation on how long a wafer can stay in a PM after being processed.

For some wafer fabrication processes, such as low pressure chemical-vapor deposition (LPCVD), there is a strict constraint on the wafer sojourn time in a PM [Kim et al., 2003; Lee and Park, 2005; Rostami et al., 2001; and Yoon and Lee, 2005]. It is referred to as residency time constraint [Rostami et al., 2001]. It means that a wafer should be unloaded from a PM within a limited time after it is processed, and otherwise, the wafer would be scrapped. Since there is no immediate buffer between PMs, it is very complicated to coordinate and schedule cluster tools to meet wafer residency time constraints. In this sense, scheduling cluster tools is somehow similar to hoist scheduling [Che and Chu, 2005 and 2007; Chen et al., 1998; and Kats et al., 1999]. However, differences exist between them due to multiple PMs for each step, and unique chamber operation for wafer fabrication. By considering residency time constraints, the methods [Kim et al., 2003; Lee and Park, 2005; and Rostami et al., 2001] find the optimal periodical schedule for dual-arm cluster tools. To improve their computational efficiency, necessary and sufficient conditions are presented for both single and dual-arm cluster tools in [Wu et al., 2008; and Wu and Zhou, 2010a]. If schedulable, closed-form scheduling algorithms are proposed to find the optimal cyclic schedules.

All the aforementioned studies are conducted under the assumption that the activity times are deterministic and known in advance. However, in practice, some abnormal events may occur in operating a cluster tool, leading to the activity time disturbance. For instance, wafer alignment failure and retrial, processing delay, and computer processing delay may take place. Thus, besides the residency time constraints, the process or robot tasks are subject to random variation or abrupt random disturbances [Kim and Lee, 2003; and Kim and Lee, 2006], which result in the wafer residency time fluctuation such that a feasible schedule obtained under the assumption of deterministic activity time becomes infeasible in practice. Therefore, it is necessary to search for an effective scheduling methodology to adapt to the abrupt random disturbances caused by abnormal events. The work [Shin et al., 2001] presents a real-time scheduling approach to deal with abnormal events by modeling the system with finite state machines. In [Kim and Lee, 2003], it is found that by increasing the robot cycle time the system can be stabilized to a steady state after a disturbance. Based on the earliest starting rule and swap strategy, Kim and Lee [2008] study the schedulability problem by considering bounded activity time variations for dual-arm cluster tools. They identify so called always schedulable and never schedulable cases by using Petri nets and branching technique. Wu et al. [2008] improve their result by showing that some never schedulable cases identified in [Kim and Lee, 2008], in fact, are always schedulable by using their new real-time controller. They analyze the effect of activity time variation on the wafer sojourn time delay in a PM in a dual-arm cluster tool by a Petri net model in [Wu and Zhou, 2010c]. Then, closed-form scheduling algorithms are proposed to find the optimal schedule for them with wafer residency time constraint and bounded activity time variation in [Wu and Zhou, 2011]. Notice that their results are applicable to only dual-arm cluster tools under a swap strategy. In a single-arm cluster tools, swapping is impossible. Thus, the results obtained for dual-arm cluster tools are not applicable to single-arm cluster tools. According to [Wu et al., 2008; Wu and Zhou, 2010c; and Wu and Zhou, 2011], to cope with the activity time variation, one needs to offset the activity time fluctuation and balance the workloads among the operation steps by properly scheduling the robot waiting times. For dual-arm cluster tools, only the robot waiting time during swapping affects the workload at a processing step and all the others have no effect on the workload. Unfortunately, for single-arm cluster tools, any scheduled robot waiting time has its contribution to the workload at a processing step to be shown later, which is totally different from the dual-arm one. Notice that offsetting activity time variation and balancing workloads among the operation steps depend on the assignment of robot waiting times. Thus, with wafer residency time constraint and activity time variation, a single-arm cluster tool is much more complex and difficult to analyze and schedule than the dual-arm one is. Up to now, this problem remains untouched. This work is the first one that aims to solve it to the authors' best knowledge.

In the next section, a Petri Net model is developed to describe the system. Then, Section 3 presents a real-time control policy and conducts the wafer sojourn time delay analysis. Two illustrative examples are used to show the results in Section 4. Finally, conclusions are given in Section 5.

II. SYSTEM MODELING

Due to their capability of dealing with concurrent activities, Petri nets (PN) are widely used as a modeling and analysis tools for flexible manufacturing system [Zhou and DiCesare, 1991; Zhou et al., 1992; Viswanadham et al., 1990; Wu, 1999; Wu and Zhou, 2001, 2004, and 2005], semiconductor manufacturing [Zhou and Jeng, 1998; Wu and Zhou, 2002 and 2006; and Liao et al., 2004], and cluster tools [Kim and Lee, 2003; Kim et al., 2003; and Zuberek, 2001]. In this section, we model single-arm cluster tools by extending the resource-oriented Petri net (ROPN) proposed in [Wu et al., 2008].

A. Finite Capacity PN

To operate an automated manufacturing system (AMS) well, such as a cluster tool, is to effectively allocate the finite resources to the production jobs [Wu et al., 2008]. PNs are powerful in modeling the behavior of resource allocation. Because of the resource limitations in an AMS, a finite capacity PN is an ideal choice to model them. Its concept is based on [Zhou and Venkatesh, 1998; and Murata, 1989]. It is a particular kind of directed graphs containing places and transitions, PN = (P, T, I, O, M, K) where $P = \{p_1, p_2, ..., p_m\}$ is a finite set of places; $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$; $I : P \times T \rightarrow \mathbb{N}$ is an input function; O: $P \times T \rightarrow \mathbb{N}$ is an output function, where $\mathbb{N} = \{0, 1, 2, \ldots\}$; M: $P \rightarrow \mathbb{N}$ is a marking representing the numbers of tokens in places with M_0 being the initial marking; and K: $P \rightarrow \{1, 2, 3, \ldots\}$ is a capacity function where $K(p)$ represents the largest number of tokens that p can hold at any time.

The preset of transition t is the set of all input places to t, i.e., ${}^\bullet t = \{p: p \in P \text{ and } I(p, t) > 0\}$. Its postset is the set of all output places from t, i.e., $t^\bullet = \{p: p \in P \text{ and } O(p, t) > 0\}$. Similarly, p's preset $^\bullet p = \{t \in T: O(p, t) > 0\}$ and postset $p^\bullet = \{t \in T: I(p, t) > 0\}$.

Definition 2.1: A transition $t \in T$ in a finite capacity PN is enabled if $\forall p \in P$,

$$M(p) \geq I(p, t) \tag{2.1}$$

and

$$K(p) \geq M(p) - I(p, t) + O(p, t) \tag{2.2}$$

If a transition is enabled, it can fire. Firing an enabled transition t in marking M yields

$$M'(p) = M(p) - I(p, t) + O(p, t) \tag{2.3}$$

Definition 2.1 means that t is enabled and can fire if there are enough tokens in all the places in $^\bullet t$ and at the same time there are enough free spaces in all the places in $t^\bullet$. Thereafter, when condition (2.1) is met, t is said to be process-enabled. When condition (2.2) is met, t is said to be resource- enabled. Thus, t is enabled only if it is both process and resource-enabled. A sequence of transition firings results in a sequence of markings. M_i is said to be reachable from M_0 if there exists a sequence of transition firings that transforms M_0 to M_i. The set of all markings reachable from M_0 is denoted by $R(M_0)$. A transition in a PN is live if it can fire at least once again in some firing sequence for every marking M reachable from M_0. A PN is live if every transition is live. The liveness of a PN assures that all events or activities in the model can happen.

B. Modeling the Wafer Flow

We model a single-arm cluster tool with wafer residency time constraint and activity variation by extending the ROPN developed in [Wu et al., 2008]. According to [Kim et al., 2003], for a wafer fabrication process without wafer revisiting, the wafer flow pattern can be denoted as $(m_1, m_2, \ldots, m_n)$, where n is the number of steps for processing the wafer and m_i is the number of PMs used to process the wafers for Step i.

We use timed place p_i to model the PMs of Step i with $K(p_i) = m_i$. The loadlocks are treated just as a processing step, called Step 0, and are modeled by p_0. Because the loadlocks can hold all the wafers in the tool, we have $K(p_0) = \infty$. Then, there are n+1 steps for the system, and let $\mathbf{\Omega} = \{0, 1, \ldots, n\}$ be the set of steps including Step 0 and $\mathbb{N}_\mathbf{n} = \{1, \ldots, n\}$ be the set of wafer processing steps. The robot is modeled by place r with $K(r) = 1$ indicating that it has only one arm. When there is a token in r it implies that the robot arm is available for delivering a wafer.

If the system operates in the process-bound region, the robot needs to wait before or after a robot task is executed. It follows from the results obtained in [Wu et al., 2008] that if the activity times are deterministic the robot waiting times can be exactly scheduled. However, with activity time variation, it is not realistic to do so.

Thus, there are two types of robot waiting times: scheduled and unscheduled waiting times. The key is to decide after what task is executed the robot should wait. With wafer residency time constraint and activity time variation, the robot waiting should be used to balance the workload among the steps and offset the activity time variation to the largest extent. It is known from [Wu et al., 2008] that, to balance the workload, the robot can wait before unloading a wafer from or loading a wafer to a step. However, if the robot waits before unloading a wafer from or loading a wafer to a step only, some activity time variation cannot be offset. Thus, we schedule the system such that the robot waits before both unloading a wafer from and loading a wafer to a step. In the PN model, we introduce places q_{i2} and q_{i1} to model the scheduled robot waiting before unloading a wafer from and loading a wafer to Step i, respectively. Because of the activity time uncertainty, when the robot arrives at Step i for unloading a wafer, the wafer may not be completed. Hence, an unscheduled robot waiting is necessary. Place q_{i3} is used to model such an unscheduled robot waiting at Step i. Noticed that, places p_i, q_{i1}, q_{i2}, and q_{i3} are timed places. Every transition in the model represents a robot task, and thus all the transitions are timed. Transitions s_{i1} and s_{01} model the loading of a wafer into a PM at Step i modeled by p_i and a loadlock modeled by p_0, respectively. Transitions s_{i2}, $i = 1, 2, \ldots, n$-1, model robot tasks of unloading a wafer from a PM at p_i and moving to p_{i+1}. Transition s_{02} models robot tasks of unloading a wafer from a loadlock at p_0 and moving to p_1. Transition s_{n2} models robot tasks of unloading a wafer from p_n and moving to a loadlock. Transitions y_{i1}, $i = 0, 1, \ldots, n$-2, represent that the robot moves from place p_{i+2} to p_i without carrying a wafer. Transitions $y_{(n-1)1}$ and y_{n1} represent that the robot moves from place p_0 to p_{n-1} and place p_1 to p_n, respectively. Pictorially, places p_i's and q_{ij}'s are denoted by ◎, and r is denoted by ⬭. Then, the PN model for a system with n steps is shown in Fig. 2.1.

In operating a cluster tool, there is an initial transient process that transfers the tool from the idle to steady state. At most of time, the tool operates in the steady state in a periodical way. When it needs to stop the tool, there is a final transient process that transfers the tool from the steady to idle state. In this paper, we address the periodical scheduling problem, which involves only the steady state. Thus, we need to model the steady state only. Because the robot task times are much shorter than the wafer processing times, when the system reaches its maximal production rate in the steady state, there are $\sum_{i=1}^{n} m_i$ wafers being processed concurrently. This implies that m_i wafers are under processing at step i, $i \in \mathbb{N}_{\mathbf{n}}$. Thus, without loss of generality, we let $M_0(p_i) = m_i$, $i \in \mathbb{N}_{\mathbf{n}}$, $M_0(r) = 1$ to indicate that the robot is idle, and $M_0(p_0) = n$ to indicate that there are always wafers to be processed.

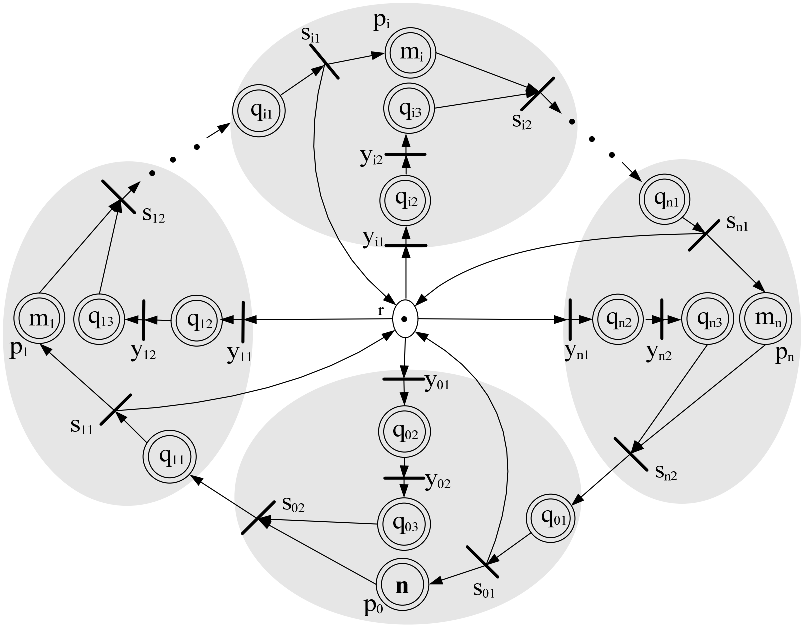

Figure 2.1. A generic PN model for a single-arm cluster tool with *n* wafer processing steps

The PN model developed above is deadlock-prone. For example, when a marking M is reached such that $M(p_i) = m_i$, $i \in \mathbb{N}_\mathbf{n}$ and $M_0(r) = 1$. At this marking, y_{01} is enabled and can fire. However, firing y_{01} makes the PN dead. Thus, it is necessary to avoid deadlocks in the PN. To do so, a control policy can be introduced to make the PN live as done in [Wu et al, 2008].

Definition 2.2 [Wu et al., 2008]: At marking M, transition y_{i1}, $i \in \mathbb{N}_\mathbf{n-1} \cup \{0\}$ is said to be control-enabled if $M(p_{i+1}) = m_{i+1} - 1$; and transition y_{n1} is said to be control-enabled if $M(p_i) = m_i$, $i \in \mathbb{N}_\mathbf{n}$.

With the control policy given in Definition 2.2, the system starts from M_0, y_{n1} fires first and followed by s_{n2}, and s_{01}. At this time, marking M_1 is reached, $y_{(n-1)1}$ fires and followed by $s_{(n-1)2}$, and s_{n1}. By doing so, at some time marking M_k is reached such that only $y_{(n-k)1}$ is enabled, $y_{(n-k)1}$ fires and followed by $s_{(n-k)2}$, and $s_{(n-k+1)1}$. In this way, finally, marking M_n is reached such that only y_{01} is enabled. Thus, y_{01} fires and followed by s_{02}, and s_{11}. Then, M_{n+1} is reached such that M_{n+1} is equivalent to M_0, this implies that a cycle is completed and the PN model becomes deadlock-free.

C. Modeling Activity Times

In scheduling cluster tools with wafer residency constraints, the starting time for each activity is crucial. Thus, for the purpose of scheduling, the model should describe the temporal aspect of activities. In the proposed PN, time is associated with both places and transitions that represent actions. The key here is how to model the activity time variation. To solve this problem, we use a time duration $[\zeta_1, \zeta_2]$ to denote a robot task time meaning that time ζ is needed to complete a task, where $\zeta \in [\zeta_1, \zeta_2]$ is any number in $[\zeta_1, \zeta_2]$. The wafer processing time is denoted as $([\zeta_1, \zeta_2], \chi)$ which means that after the completion of processing this task with ζ time units the longest time delay in a PM must not be greater than χ. If ζ is associated with a transition t, it means that firing t takes ζ time units. If ζ is associated with a place p, a token in p must stay in p at least for ζ time units before it can enable an output transition of p. It should also be pointed out that, for a robot task or wafer process at a PM, $\zeta \in [\zeta_1, \zeta_2]$ is obtained by observing in the real-time operation. If $\zeta \in [\zeta_1, \zeta_2]$ represents a scheduled robot waiting time, ζ can be scheduled to be any number in $[\zeta_1, \zeta_2]$. It implies that it is controlled by the scheduler, and thus controllable. However, if $\zeta \in [\zeta_1, \zeta_2]$ represents an unscheduled robot waiting time, ζ is determined by the activity time variation, and hence not controllable. The time durations for different transitions and places are shown in Table 2.1. Durations $\mu_{y_{i1}}$, $\mu_{y_{(n-1)1}}$, and $\mu_{y_{n1}}$ are needed for the robot to move from Step $i + 2$ to Step i, $i \in \mathbb{N}_{n-2}$, from Step 0 to Step n-1, and from Step 1 to Step n, respectively. They can be any number in $[\alpha, \beta]$. The time λ_{i2} is taken for the robot unloading a wafer from a PM at p_i and moving to p_{i+1}. The time λ_{n2} is taken for the robot unloading a wafer from p_n and moving to a loadlock. It is assumed that the times taken for λ_{i2} and λ_{n2} are same. They can be any number in $[c, d]$. However, transition s_{02} models the robot unloading a wafer from a loadlock, aligning and moving to p_1 together, and thus has the time $\lambda_{02} \in [c_0, d_0]$ which is different from λ_{i2}, $i \in \mathbb{N}_n$.

Table 2.1. The time durations associated with transitions and places

Symbol	Transition or place	Actions	Allowed time duration
λ_{i1}	$s_{i1} \in T$	Robot loads a wafer into Step i, $i \in \mathbf{\Omega}$	$[c, d]$
λ_{01}	$s_{01} \in T$	Robot loads a wafer into a loadlock	$[c, d]$
λ_{i2}	$s_{i2} \in T$	Robot unloads a wafer from Step i and moves to p_{i+1}, $i \in \mathbb{N}_{n-1}$	$[c+\alpha, d+\beta]$
	$s_{n2} \in T$	Robot unloads a wafer from Step n and moves to loadlock	
λ_{02}	$s_{02} \in T$	Robot unloads a wafer from a loadlock, aligns it and moves to p_1 together	$[c_0+\alpha, d_0+\beta]$
$\mu_{y_{i1}}$	$y_{i1} \in T$	Robot moves from Step $i + 2$ to Step i, $i \in \mathbb{N}_{n-2}$	$[\alpha, \beta]$
$\mu_{y_{(n-1)1}}$	$y_{(n-1)1} \in T$	Robot moves from Step 0 to Step n-1	$[\alpha, \beta]$
$\mu_{y_{n1}}$	$y_{n1} \in T$	Robot moves from Step 1 to Step n	$[\alpha, \beta]$
τ_i	$p_i \in P$	A wafer being processed and waiting in p_i, $i \in \mathbb{N}_n$	$([a_i, b_i], \delta_i)$
ω_{i2}	$q_{i2} \in P$	Robot waits before unloading a wafer from Step i, $i \in \mathbf{\Omega}$	$[0, \infty]$

ω_{i1}	$q_{i1} \in P$	Robot waits before loading a wafer to Step i, $i \in \mathbf{\Omega}$	$[0, \infty]$

From Definition 2.2, it is known that the PN model is deadlock-free. However, with residency time constraint, its deadlock-freeness does not mean that it is live because it requires that a token in a place p_i should stay in it within a given time window. Furthermore, the activity time variation can make the token's sojourn time longer than the desired one. Nevertheless, it is easy to verify that if for any i such that a token is allowed to stay in p_i for unlimited time, the PN is live. Let τ_i denote the sojourn time of a token in p_i. Further, let ς_i be a sample in $[a_i, b_i]$. Then, we present the liveness definition of the timed PN for single-arm cluster tools with residency time constraints.

Definition 2.3: A timed PN for single-arm cluster tools with residency time constraint is said to be live if $\forall$ $M \in R(M_0)$, $i \in \mathbb{N}_n$, $\varsigma_i \in [a_i, b_i]$, s_{i2} is enabled for any wafer in p_i at τ_i when $\tau_i - \varsigma_i \le \delta_i$.

Based on the model, to obtain a feasible schedule for a single-arm cluster tool with wafer residency time constraints and activity time variation is to determine the scheduled robot waiting times such that the timed PN is live.

III. ANALYSIS OF TEMPORAL PROPERTIES

With activity time variation, the time taken by an activity is ζ that can be any number in $[\zeta_1, \zeta_2]$ and it is obtained by on-line observation. Thus, it is not meaningful to schedule such a system by using an off-line schedule, and instead real-time scheduling is necessary. If the activity times are deterministic, a periodic schedule is used because the cluster tools operate cyclically. Thus, to schedule a cluster tool subject to activity time variation, an effective way is to find a periodic schedule based on deterministic activity times and then make it adapt to certain random activity time variation. To do so, it is important to know the effect of time variation on the system. Therefore, in this section, based on the PN model, a real-time control policy is proposed. Then, we analyze the effect of activity time variation on wafer sojourn time delay at a PM under this policy.

A. Temporal Properties under Normal Condition

As discussed before, an activity takes time $\zeta \in [\zeta_1, \zeta_2]$. To effectively analyze the system, we define the normal condition under which an activity takes $\zeta = \zeta_1$ time units. Then, in the real-time operation, the time needed for an activity can be denoted as $\zeta = \zeta_1 + \Delta\zeta$ with $\zeta_1 \le \zeta_1 + \Delta\zeta \le \zeta_2$. In this way, $\Delta\zeta \neq 0$ can be seen as a disturbance and ζ can be any number in $[\zeta_1, \zeta_2]$. Thus, activity time variation can be treated as disturbance to the normal condition. Hence, to schedule cluster tools with activity time variation, we can obtain a periodic schedule under the normal condition plus a real-time regulator to adapt to the activity time disturbance. We first analyze the properties under the normal condition.

Observe the PN model shown in Fig. 2.1, we know that, under the normal condition, to complete the processing of a wafer in Step i, $i \in \mathbb{N}_{n-1}$, the following transition firing (activities) sequence must be executed: s_{i2} (time $c+\alpha$) → robot waiting in $q_{(i+1)1}$ (time $\omega_{(i+1)1}$) →$s_{(i+1)1}$ (time c) →$y_{(i-1)1}$ (time α) → robot waiting in $q_{(i-1)2}$ (time $\omega_{(i-1)2}$) →$s_{(i-1)2}$ (time $c+\alpha$) → robot waiting in q_{i1} (time ω_{i1}) →s_{i1} (time c) → processing a wafer at Step i (time τ_i) →s_{i2} (time $c+\alpha$) again. In this way, a cycle is completed and it takes $\tau_i + 4c + 3\alpha + \omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1}$ time units. Notice that τ_i should be within $[a_i, a_i + \delta_i]$ and there are m_i PMs for Step i. When $\tau_i = a_i$ we have the lower permissive cycle time at Step i as

$$\theta_{iL} = \frac{a_i + 4c + 3\alpha + \omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1}}{m_i}, 1 < \mathrm{i} < \mathrm{n} \tag{3.1}$$

When $\tau_i = a_i + \delta_i$ we have the upper permissive cycle time at Step i as

$$\theta_{iU} = \frac{a_i + \delta_i + 4c + 3\alpha + \omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1}}{m_i}, 1 < \mathrm{i} < \mathrm{n} \tag{3.2}$$

For Step 1, the lower cycle time is

$$\theta_{1L} = \frac{a_1 + 3c + c_0 + 3\alpha + \omega_{02} + \omega_{21} + \omega_{11}}{m_1} \tag{3.3}$$

Its upper cycle is

$$\theta_{1U} = \frac{a_1 + \delta_1 + 3c + c_0 + 3\alpha + \omega_{02} + \omega_{21} + \omega_{11}}{m_1} \tag{3.4}$$

For Step n, the lower cycle time is

$$\theta_{nL} = \frac{a_n + 4c + 3\alpha + \omega_{(n-1)2} + \omega_{01} + \omega_{n1}}{m_n} \tag{3.5}$$

Its upper cycle is

$$\theta_{nU} = \frac{a_n + \delta_n + 4c + 3\alpha + \omega_{(n-1)2} + \omega_{01} + \omega_{n1}}{m_n} \tag{3.6}$$

In fact, Expressions (3.1), (3.3), and (3.5) describe the workloads among the steps when robot waiting exists. It shows that any robot waiting affects the workload. Also, it follows from Expressions (3.1) - (3.6) that the robot waiting times have effect on the permissive wafer sojourn time range for the steps. Thus, by carefully regulating the robot waiting times, it can change the permissive range among the steps. If the robot waiting time is removed from the above expressions, we can obtain the lower and upper workloads with no robot waiting for each step as follows.

$$\vartheta_{iL} = \frac{a_i + 4c + 3\alpha}{m_i}, \ 1 < \mathrm{i} \le \mathrm{n} \tag{3.7}$$

$$\vartheta_{iU} = \frac{a_i + \delta_i + 4c + 3\alpha}{m_i}, \ 1 < \mathrm{i} \le \mathrm{n} \tag{3.8}$$

$$\vartheta_{1L} = \frac{a_1 + 3c + c_0 + 3\alpha}{m_1} \tag{3.9}$$

$$\vartheta_{1U} = \frac{a_1 + \delta_1 + 3c + c_0 + 3\alpha}{m_1} \tag{3.10}$$

Expressions (3.7)–(3.10) present the workload balance information that affects the existence of a feasible schedule. It follows from (3.2) and (3.8) that $\theta_{iU} > \vartheta_{iU}$ if $\omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1} > 0$. It implies that, by making $\omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1} > 0$, the cycle time of Step i is increased without increasing the wafer sojourn time. Thus, it is possible to adjust the robot waiting times such that the permissive wafer sojourn time ranges among the steps are balanced to some extent to guarantee the feasibility. At the same time, any robot task time delay is just like robot waiting and affects the schedule's feasibility, which complicates the analysis and scheduling problem.

To schedule a single-arm cluster tool with residency time constraint under the normal condition, the key is that $a_i \le \tau_i \le a_i + \delta_i$ must be satisfied. Hence, we need to know how τ_i should be calculated. The wafer sojourn time at p_i depends on the robot cycle time and the workloads of the steps. Under the normal condition, the unscheduled robot waiting time is zero. Hence, from the PN model shown in Fig. 2.1, we know that, during a wafer stays in a PM at Step i, the following event sequence forms a robot cycle: firing s_{i2} → robot waiting in $q_{(i+1)1}$ → firing $s_{(i+1)1}$ → firing $y_{(i-1)1}$ → robot waiting in $q_{(i-1)2}$ → firing $s_{(i-1)2}$ → robot waiting in q_{i1} → ***firing s_{i1} with the k-th wafer*** → firing $y_{(i-2)1}$ → robot waiting in $q_{(i-2)2}$ → firing $s_{(i-2)2}$ → robot waiting in $q_{(i-1)1}$ → firing $s_{(i-1)1}$ → …→ firing s_{21} → firing y_{01} → robot waiting in q_{02} → firing s_{02} → robot waiting in q_{11} → firing s_{11} → firing y_{n1} → robot waiting in q_{n2} → firing s_{n2} → robot waiting in q_{01} → firing s_{01} →…→ ***firing s_{i1} with the (k+1)-th wafer*** → firing $y_{(i-2)1}$ → robot waiting in $q_{(i-2)2}$ → firing $s_{(i-2)2}$ → robot waiting in $q_{(i-1)1}$ → firing $s_{(i-1)1}$ → …→ firing s_{21} → firing y_{01} → robot waiting in q_{02} → firing s_{02} → robot waiting in q_{11} → firing s_{11} → firing y_{n1} → robot waiting in q_{n2} → firing s_{n2} → robot waiting in q_{01} → firing s_{01} →…→ ***firing s_{i1} with the (k+m_i-1)-th wafer*** → firing $y_{(i-2)1}$ → robot waiting in $q_{(i-2)2}$ → firing $s_{(i-2)2}$ → robot waiting in $q_{(i-1)1}$ → firing $s_{(i-1)1}$ → …→ firing s_{21} → firing y_{01} → robot waiting in q_{02} → firing s_{02} → robot waiting in q_{11} → firing s_{11} → firing y_{n1} → robot waiting in q_{n2} → firing s_{n2} → robot waiting in q_{01} → firing s_{01} →…→ ***firing s_{i2} with the k-th wafer*** →

robot waiting in $q_{(i+1)1}$ → firing $s_{(i+1)1}$ → firing $y_{(i-1)1}$ → robot waiting in $q_{(i-1)2}$ → firing $s_{(i-1)2}$ → robot waiting in q_{i1} → ***firing s_{i1} with the $(k+m_i)$-th wafer.***

With this event sequence, it is known that when s_{i1} fires for the $(k+m_i)$-th time for loading the $(k+m_i)$-th wafer into p_i while s_{i2} fires for the $(k+m_i)$-th time unloading the k-th wafer from p_i and moving to p_{i+1}. In other words, the k-th wafer is loaded into p_i by the k-th firing of s_{i1} and unloaded from p_i and moving to p_{i+1} by the $(k+m_i)$-th firing of s_{i2}, or the sojourn time duration of the k-th wafer in p_i is between the k-th firing of s_{i2} and the $(k+m_i)$-th firing of s_{i1}. During this time, the robot undergoes m_i cycles except that transitions s_{i2}, $s_{(i+1)1}$, y_{i-1}, $s_{(i-1)2}$, and s_{i1} fire $(m_i - 1)$ times, and the robot waits in $q_{(i-1)2}$, $q_{(i+1)1}$, and q_{i1} (m_i-1) times. Let ζ^j denote the time taken by the j-th occurrence of an event. Notice that, under the normal condition, ω_{i2} and ω_{i1}, $i \in \mathbf{\Omega}$, are the scheduled waiting times and are constant, while ω_{i3}, $i \in \mathbf{\Omega}$, should be zero. Then, the wafer sojourn time in p_i is

$$\tau_i = \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\mu^j_{y_{d1}} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}(\lambda^j_{d1}+\lambda^j_{d2}) + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega^j_{d2} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega^j_{d1} - (\lambda^k_{i1} + \lambda^k_{i2} + \lambda^k_{(i+1)1}$$

$$+ \mu^k_{y_{(i-1)1}} + \lambda^k_{(i-1)2} + \omega^k_{(i-1)2} + \omega^k_{i1} + \omega^k_{(i+1)1} \tag{3.11}$$

Under the normal condition, $\lambda^j_{02} = c_0 + \alpha$, for all j; $\lambda^j_{i2} = c + \alpha$, for all j and $i \in \mathbb{N}_{\mathbf{n}}$; $\mu^j_{y_{d1}} = \alpha$, $\omega^j_{i2} = \omega_{i2}$, $\omega^j_{i1} = \omega_{i1}$ and $\lambda^j_{i1} = c$ for all j and i. Thus, at any steady state marking M, the wafer sojourn time in p_i is

$$\tau_1 = m_1 \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum\nolimits_0^n \omega_{d2} + \sum\nolimits_0^n \omega_{d1}] - (3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21})$$

$$= m_1 \times \psi - (3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21}) \tag{3.12}$$

$$\tau_i = m_i \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum\nolimits_0^n \omega_{d2} + \sum\nolimits_0^n \omega_{d1}] - (4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1})$$

$$= m_i \times \psi - (4c + 3\alpha + \omega_{(i-1)2} + \omega_{(i+1)1} + \omega_{i1}),\ i = 2, 3, \ldots, n-1 \tag{3.13}$$

$$\tau_n = m_n \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum\nolimits_0^n \omega_{d2} + \sum\nolimits_0^n \omega_{d1}] - (4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01})$$

$$= m_n \times \psi - (4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01}) \tag{3.14}$$

Notice that, under the normal condition, the robot cycle time is

$$\psi = 2(n+1)\alpha + (2n+1)c + c_0 + \sum\nolimits_0^n \omega_{d2} + \sum\nolimits_0^n \omega_{d1} = \psi_1 + \psi_2 \tag{3.15}$$

where $\psi_1 = 2(n+1)\alpha + (2n+1)c + c_0$ is a constant and known in advance and $\psi_2 = \sum\nolimits_0^n \omega_{d2} + \sum\nolimits_0^n \omega_{d1}$ is to be determined by a scheduler. It should also be noticed that ψ is independent of the wafer processing times.

Let $\theta_1 = (\tau_1 + 3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21})/(m_1)$, $\theta_i = (\tau_i + 4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1})/(m_i)$, $i = 2, 3, \ldots,$

n -1, and $\theta_n = (\tau_n + 4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01})/(m_n)$ denote the cycle time for step i, $i \in \mathbb{N}_\mathbf{n}$. Further, let θ be the production cycle of the system. Because the steady-state process of single-arm cluster tools is a series process, the production rate is the same for all the steps and this production rate is the one for the system. Thus, we have the following proposition.

Proposition 3.1: A single-arm cluster tool should be scheduled such that when the steady-state process is reached, all processing steps have the same cycle time, or we have

$$\theta = \theta_1 = \theta_2 = \ldots = \theta_n. \tag{3.16}$$

With the PN model shown in Fig. 2.1, we can analyze the relationship between the production cycle and the robot cycle. Assume that wafer W_k is loaded into step i at time τ_k and W_{k+1} is loaded into it at τ_{k+1}. Then, $[\tau_k, \tau_{k+1}]$ forms a processing cycle for step i. During this time, s_{i1} fires twice, and the robot completes the following activities: firing s_{i1} → firing $y_{(i-2)1}$ → waiting in $q_{(i-2)2}$ → firing $s_{(i-2)2}$ → waiting in $q_{(i-1)1}$ → firing $s_{(i-1)1}$ → firing $y_{(i-3)1}$ → …→ firing y_{01} → waiting in q_{02} → firing s_{02} → waiting in q_{11} → firing s_{11} → firing y_{n1} → waiting in q_{n2} → firing s_{n2} → waiting in q_{01} → firing s_{01} → firing $y_{(n-1)1}$ → …→ firing y_{i1} → waiting in q_{i2} → firing s_{i2} →waiting in $q_{(i+1)1}$ → firing $s_{(i+1)1}$→ firing $y_{(i-1)1}$ → waiting in $q_{(i-1)2}$ → firing $s_{(i-1)2}$ → waiting in q_{i1} → firing s_{i1} again. It can be seen that, during this time, the robot completes exactly one cycle. Thus, we have the following proposition.

Proposition 3.2: Under the normal condition, a single-arm cluster tool should be scheduled such that when the steady state is reached the wafer production cycle is equal to the robot cycle, or we have

$$\theta = \theta_1 = \theta_2 = \ldots = \theta_n = \psi \tag{3.17}$$

For the robot cycle time in (3.15), α, c, and c_0 are all deterministic, only ω_{d2} and ω_{d1}, $d \in \mathbf{\Omega}$, are changeable, or ψ_1 is deterministic and ψ_2 can be regulated. Thus, based on the PN, to schedule a single-arm cluster tool with residency time constraints under the normal condition is to appropriately regulate ω_{d2}, and ω_{d1}, $d \in \mathbf{\Omega}$, such that (3.17) holds and at the same time the wafer residency time constraints are satisfied.

B. Real-Time Control Policy

Under the normal condition, $\lambda_{02}^{j} = c_0 + \alpha$, for all j; $\lambda_{d2}^{j} = c + \alpha$, for all j and $d \in \mathbb{N}_\mathbf{n}$; $\mu_{y_{d1}}^{j} = \alpha$ and $\lambda_{d1}^{j} = c$ for all j and $d \in \mathbf{\Omega}$. Meanwhile, ω_{d2}, and ω_{d1}, for all $d \in \mathbf{\Omega}$, are scheduled to be constant. It follows from (3.12) - (3.14) that τ_i, $i \in \mathbb{N}_\mathbf{n}$, is constant. Hence, to obtain a feasible periodic schedule under the normal condition is to determine ω_{d2}, and ω_{d1}, for all d, such that $a_i \leq \tau_i \leq a_i + \delta_i$ for any $i \in \mathbb{N}_\mathbf{n}$. However, when the activity times are subject to random variation, the situation is different. Thus, it is necessary to analyze the effect of activity time variation on the wafer sojourn time.

The random activity time variation can be seen as random disturbance to the normal condition. Thus, we let $\mu^j_{y_{d1}} = \alpha + \sigma^j_{y_{d1}}$, $\lambda^j_{d1} = c + \rho^j_{d1}$, $\lambda^j_{02} = c_0 + \alpha + \rho^j_{02}$, and $\lambda^j_{d2} = c + \alpha + \rho^j_{d2}$, $d \in \mathbb{N}_\mathbf{n}$. Meanwhile, the wafer processing time variation causes unscheduled robot waiting. Consider that ω_{d2}, and ω_{d1}, for any $d \in \mathbb{N}_\mathbf{n}$, are given as constant for the periodic schedule under the normal condition, and ω^j_{d2} and ω^j_{d1} are adjusted dynamically to adapt to the random activity time variation. Then, taking the activity time variation into account, from (3.11) we have

$$\tau_1 = m_1 \times [2(n+1)\alpha + (2n+1)c + c_0] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\sigma^j_{y_{d1}} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d1} + \sum_{d=1}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d2} + \sum_{j=k}^{k+m_j-1}\rho^j_{02} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}(\omega^j_{d1} + \omega^j_{d2} + \omega^j_{d3}) - (3c + c_0 + 3\alpha + \omega^k_{02} + \omega^k_{11} + \omega^k_{21} + \omega^k_{03} + \rho^k_{11} + \rho^k_{12} + \rho^k_{21} + \rho^k_{02} + \sigma^k_{y_{01}} \tag{3.18}$$

$$\tau_i = m_i \times [2(n+1)\alpha + (2n+1)c + c_0] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\sigma^j_{y_{d1}} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d1} + \sum_{d=1}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d2} + \sum_{j=k}^{k+m_j-1}\rho^j_{02} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}(\omega^j_{d1} + \omega^j_{d2} + \omega^j_{d3}) - (4c + 3\alpha + \omega^k_{(i-1)2} + \omega^k_{i1} + \omega^k_{(i+1)1} + \omega^k_{(i-1)3} + \rho^k_{i1} + \rho^k_{i2} + \rho^k_{(i+1)1} + \rho^k_{(i-1)2} + \sigma^k_{y_{(i-1)1}},\ i = 2, 3, \ldots, n\text{-}1 \tag{3.19}$$

$$\tau_n = m_n \times [2(n+1)\alpha + (2n+1)c + c_0] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\sigma^j_{y_{d1}} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d1} + \sum_{d=1}^{n}\sum_{j=k}^{k+m_i-1}\rho^j_{d2} + \sum_{j=k}^{k+m_j-1}\rho^j_{02} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}(\omega^j_{d1} + \omega^j_{d2} + \omega^j_{d3}) - (4c + 3\alpha + \omega^k_{(n-1)2} + \omega^k_{n1} + \omega^k_{01} + \omega^k_{(n-1)3} + \rho^k_{n1} + \rho^k_{n2} + \rho^k_{01} + \rho^k_{(n-1)2} + \sigma^k_{y_{(n-1)1}} \tag{3.20}$$

It follows from (3.19) that if $\omega^j_{d2} = \omega_{d2}$, and $\omega^j_{d1} = \omega_{d1}$, a nonzero value of $\sigma^j_{y_{d1}}$, ρ^j_{d2}, and ρ^j_{d1} causes a delay of τ_i. This may result in the violation of wafer residency constraints if it takes a_i time units for the k-th wafer to be completed in p_i. To reduce the effect on the delay of τ_i caused by $\sigma^j_{y_{d1}}$, ρ^j_{d2} and ρ^j_{d1}, we can adjust ω_{d2} and ω_{d1} dynamically. To do so, we can observe $\sigma^j_{y_{d1}}$, ρ^j_{d2}, and ρ^j_{d1} in real-time. If there exists a nonzero value of $\sigma_{y_{d1}}^{j}$, ρ_{d2}^{j}, and ρ_{d1}^{j}, the robot waiting time in q_{d2} and q_{d1} can be shortened by adjusting ω_{d2} and ω_{d1} in real-time. Let $\omega_{i2}^{j} = max\{(\omega_{i2} - \sigma_{y_{i1}}^{j}),\ 0\}$. Further, let $\omega_{i2}^{j} + \sigma_{y_{i1}}^{j} = \omega_{i2} + \eta_{i2}^{j}$. Surely, we have $\eta_{i2}^{j} = max\{(\sigma_{y_{i1}}^{j} - \omega_{i2}),\ 0\}$. In this way, the effect of $\sigma_{y_{d1}}^{j}$ on τ_i can be made as small as possible. Let $\omega_{01}^{j} = max\{(\omega_{01} - \rho_{n2}^{j}),\ 0\}$, and $\omega_{i1}^{j} = max\{(\omega_{i1} - \rho_{(i-1)2}^{j}),\ 0\}$. Further, let $\omega_{01}^{j} + \rho_{n2}^{j} = \omega_{01} + \eta_{01}^{j}$, $\omega_{i1}^{j} + \rho_{(i-1)2}^{j} = \omega_{i1} + \eta_{i1}^{j}$. Surely, we have $\eta_{01}^{j} = max\{(\rho_{n2}^{j} - \omega_{01}),\ 0\}$, and $\eta_{i1}^{j} = max\{(\rho_{(i-1)2}^{j} - \omega_{i1}),\ 0\}$. In this way, the effect of ρ_{d2}^{j} can be made as small as possible. The dynamical regulation of ω_{i2}^{j} and ω_{i1}^{j} can be summarized as the following real-time operation policy.

Control Policy 3.1 (Real-time Operation Policy): The following rules are applied for the real-time operation of the cluster tool.

1) Under the normal condition find a periodic schedule by determining ω_{i2}, and ω_{i1}, $i \in \mathbf{\Omega}$.
2) Transition s_{01} is enabled if the j-th token stays in q_{01} for $\omega_{01}^{j} = max\{(\omega_{01} - \rho_{n2}^{j}),\ 0\}$, and s_{i1}, $i \in \mathbb{N}_{\mathbf{n}}$, is enabled if the j-th token stays in q_{i1} for $\omega_{i1}^{j} = max\{(\omega_{i1} - \rho_{(i-1)2}^{j}),\ 0\}$.
3) Transition y_{i2}, $i \in \mathbf{\Omega}$, is enabled if the j-th token stays in q_{i2} for $\omega_{i2}^{j} = max\{(\omega_{i2} - \sigma_{y_{i1}}^{j}),\ 0\}$.
4) Transitions s_{i2} and y_{i1}, $i \in \mathbf{\Omega}$, fire once enabled.

By Control Policy 3.1, s_{i2} can fire when there is a token in q_{i3} and a wafer (token) in p_i is completed. This implies that the token waiting time ω_{i3}^{j} in q_{i3} (or the firing of s_{i2}) depends on whether a wafer in p_i is completed or not. Based on such policy, we can rewrite τ_i as follows.

$$\begin{aligned}
\tau_1 = {} & m_1 \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} + \\
& \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21} + \omega_{02}^{k} + \omega_{11}^{k} + \omega_{21}^{k} - \omega_{02} - \omega_{11} - \\
& \omega_{21} + \omega_{03}^{k} + \rho_{11}^{k} + \rho_{12}^{k} + \rho_{21}^{k} + \rho_{02}^{k} + \sigma_{y_{01}}^{k}) \\
= {} & \Lambda_1 + \Theta_1 \qquad (3.21)
\end{aligned}$$

$$\tau_i = m_i \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} +$$

$$\sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1} + \omega_{(i-1)2}^{k} + \omega_{i1}^{k} + \omega_{(i+1)1}^{k} - \omega_{(i-1)2} - \omega_{i1} - \omega_{(i+1)1} + \omega_{(i-1)3}^{k} + \rho_{i1}^{k} + \rho_{i2}^{k} + \rho_{(i+1)1}^{k} + \rho_{(i-1)2}^{k} + \sigma_{y_{(i-1)1}}^{k})$$

$$= \Lambda_i + \Theta_i,\ 1 < \mathrm{i} < \mathrm{n} \tag{3.22}$$

$$\tau_n = m_n \times [2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}] + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01} + \omega_{(n-1)2}^{k} + \omega_{n1}^{k} + \omega_{01}^{k} - \omega_{(n-1)2} - \omega_{n1} - \omega_{01} + \omega_{(n-1)3}^{k} + \rho_{n1}^{k} + \rho_{n2}^{k} + \rho_{01}^{k} + \rho_{(n-1)2}^{k} + \sigma_{y_{(n-1)1}}^{k})$$

$$= \Lambda_n + \Theta_n \tag{3.23}$$

where $\Lambda_1 = m_1 \times \psi - (3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21})$, $\Lambda_i = m_i \times \psi - (4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1})$, $i = 2, 3, \ldots, n-1$, and $\Lambda_n = m_n \times \psi - (4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01})$ are the scheduled sojourn time under the normal condition given by (3.12) - (3.14), and are constant when the periodic schedule is determined.

We have the sojourn time disturbance as $\Theta_1 = \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (\omega_{02}^{k} + \omega_{11}^{k} + \omega_{21}^{k} - \omega_{02} - \omega_{11} - \omega_{21} + \omega_{03}^{k} + \rho_{11}^{k} + \rho_{12}^{k} + \rho_{21}^{k} + \rho_{02}^{k} + \sigma_{y_{01}}^{k})$, $\Theta_i = \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (\omega_{(i-1)2}^{k} + \omega_{i1}^{k} + \omega_{(i+1)1}^{k} - \omega_{(i-1)2} - \omega_{i1} - \omega_{(i+1)1} + \omega_{(i-1)3}^{k} + \rho_{i1}^{k} + \rho_{i2}^{k} + \rho_{(i+1)1}^{k} + \rho_{(i-1)2}^{k} + \sigma_{y_{(i-1)1}}^{k})$, $i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\}$, and $\Theta_n = \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^{j} + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^{j} - (\omega_{(n-1)2}^{k} + \omega_{n1}^{k} + \omega_{01}^{k} - \omega_{(n-1)2} - \omega_{n1} - \omega_{01} + \omega_{(n-1)3}^{k} + \rho_{n1}^{k} + \rho_{n2}^{k} + \rho_{01}^{k} + \rho_{(n-1)2}^{k} + \sigma_{y_{(n-1)1}}^{k})$. In Θ_i, η_{d2}^{j}, η_{d1}^{j}, ρ_{i1}^{k}, ρ_{i2}^{k}, $\rho_{(i+1)1}^{k}$, $\rho_{(i-1)2}^{k}$, and $\sigma_{y_{i-1}}^{k}$ are obtained by real-time observation. It should be pointed out that ω_{d1} and ω_{d2} in Θ_i are determined by an off-line schedule and are known in advance. However, ω_{d3}^{j} is uncontrollable and is caused by activity time variation and the real-time regulation of ω_{d2} and ω_{d1}. Thus, to evaluate Θ_i, the key is to analyze the effect of activity variation on ω_{d3}^{j}.

C. Wafer Sojourn Time Delay Analysis

It follows from the last sub-section that the delay of sojourn time τ_i for the j-th wafer in p_i is equal to ω_{i3}^{j} in q_{i3}. Now, based on the PN and above analysis, we analyze the delay of wafer sojourn time τ_i in p_i by deriving the upper bound of ω_{i3}^{j}.

With wafer flow pattern $(m_1, m_2, \ldots, m_n)$, there are $m = m_1 + m_2 + \ldots + m_n$ wafers in processing at any time. Without loss of generality, they can be numbered as $W_1, W_2, \ldots, W_m$. Let WP_i denote the set of wafers being processed in p_i. Further, let $E1 = m_2 + \ldots + m_n + 1$, $L1 = m$, $Ei = m_{i+1} + \ldots + m_n + 1$, $Li = m_i + \ldots + m_n$, $i \in \mathbf{N_{n-1}}$, $En = 1$, and $Ln = m_n$, such that W_{Ei} and W_{Li} are the earliest and the latest wafers released into p_i, respectively. Hence, we have $WP_1 = \{W_{E1}, W_{E1+1}, \ldots, W_{L1}\}, \ldots, WP_i = \{W_{Ei}, W_{Ei+1}, \ldots, W_{Li}\}, \ldots, WP_n = \{W_{En}, W_{En+1} \ldots, W_{Ln}\}$. Let ω_{i3}^{1} be the robot waiting time in q_{i3} before the robot unloads the completed wafer W_{Ei} at p_i, and ω_{i3}^{2} before the robot unloads W_{Ei+1} at p_i, and so on. Then, the wafer sojourn time delay is analyzed step by step as follows.

Theorem 3.1: Assume that wafers $W_{E(i+1)}$, $i \in \mathrm{N}_{\mathbf{n-1}}$, and all the robot tasks are performed under the normal condition, and the processing of wafer W_{Ei}, $i \in \mathrm{N}_{\mathbf{n-1}}$, takes $\upsilon_i \in (a_i, b_i]$ time units. Then, $\omega_{i3}^{1} = max\{(\upsilon_i - \Lambda_i), 0\}$, $i \in \mathrm{N}_{\mathbf{n-1}}$.

Proof: By assumption, under the normal condition, the system is scheduled such that $\omega_{i3} = 0$, $i \in \mathbf{\Omega}$, and the delay in processing W_{Ei}, $i \in \mathrm{N}_{\mathbf{n-1}}$, has no effect on the processing of $W_{E(i+1)}, \ldots, W_{En}$. Thus, $\omega_{d3}^{1} = 0$, $d = i+1, \ldots, n$, must hold. It is known that Λ_i is the scheduled wafer sojourn time at p_i. Then it follows from Control Policy 3.1 that $\omega_{i3}^{1} = max\{(\upsilon_i - \Lambda_i), 0\}$, $i \in \mathrm{N}_{\mathbf{n-1}}$, must hold. ■

We call $max\{(\upsilon_i - \Lambda_i), 0\}$ the time delay caused by a delay in processing a wafer. With Theorem 3.1, we have the following corollary when there is a delay in processing a wafer at p_n.

Corollary 3.1: Assume that all the robot tasks are performed under the normal condition, and the processing of wafer W_{En} takes $\upsilon_n \in (a_n, b_n]$ time units. Then, $\omega_{n3}^{1} = max\{(\upsilon_n - \Lambda_n), 0\}$.

Theorem 3.2: Assume that: 1) wafers $W_{E(i+1)}$, $i \in \{\mathrm{N}_{\mathbf{n-1}} - \{1\}\}$, and all the robot tasks are completed under the normal condition; 2) the processing of wafer W_{Ei}, $i \in \{\mathrm{N}_{\mathbf{n-1}} - \{1\}\}$, takes $\upsilon_i \in (a_i, b_i]$ time units; 3) wafers $W_{E(i-1)}$, $W_{E(i-2)}, \ldots, W_{E(i-k+1)}$ are completed under the normal condition with $1 \le k \le i-1$; and 4) the processing of wafer $W_{E(i-k)}$, $i \in \{\mathrm{N}_{\mathbf{n-1}} - \{1\}\}$, takes $\upsilon_{i-k} \in (a_{i-k}, b_{i-k}]$ time units. Then, $\omega_{(i-k)3}^{1} = max\{(\upsilon_{i-k} - \Lambda_{i-k} - \omega_{i3}^{1}), 0\}$, $i \in \{\mathrm{N}_{\mathbf{n-1}}$ -

{1}}.

Proof: Because the robot waits in q_{i3} for ω_{i3}^{1} time units, there is ω_{i3}^{1} time-unit delay in addition to the scheduled one when it arrives at $q_{(i-k)3}$. Thus, the robot needs to wait for $max\{(\upsilon_{i-k} - \Lambda_{i-k} - \omega_{i3}^{1}), 0\}$ time units for the completion of $W_{E(i-k)}$ at p_{i-k}. ■

Similarly to Corollary 3.1, with Theorem 3.2, we have the following corollary.

Corollary 3.2: Assume that: 1) all the robot tasks are completed under the normal condition; 2) the processing of wafer W_{En} takes $\upsilon_n \in (a_n, b_n]$ time units; 3) wafers $W_{E(n-1)}$, $W_{E(n-2)}$, …, $W_{E(n-k+1)}$ are completed under the normal condition with $1 \leq k \leq n$-1; and 4) the processing of wafer $W_{E(n-k)}$ takes $\upsilon_{n-k} \in (a_{n-k}, b_{n-k}]$ time units. Then, $\omega_{(n-k)3}^{1} = max\{(\upsilon_{n-k} - \Lambda_{n-k} - \omega_{n3}^{1}), 0\}$.

It follows from Theorem 3.2 that $\omega_{i3}^{1} + \omega_{(i-k)3}^{1} = max\{(\upsilon_i - \Lambda_i), (\upsilon_{i-k} - \Lambda_{i-k}), 0\}$. This implies that at the time when the robot leaves $q_{(i-k)3}$, the total robot activity time delay, or the accumulated robot waiting time in $q_{(i-k)3}$, is determined by the larger delay in processing wafers W_{Ei} and $W_{E(i-k)}$. The key is that after the delay in processing wafer W_{Ei} at p_i, one wafer is processed at p_{i-1}, p_{i-2}, …, and p_{i-k+1}. Thus, it can be seen that wafers W_{Ei}, $W_{E(i-1)}$, …, and $W_{E(i-k+1)}$ are processed in the same cycle. Thereafter, we think that it is a cycle when the robot leaves q_{i3}, performs a series of tasks, and comes back to q_{i3} again. By this observation, we have the following corollary immediately.

Corollary 3.3: Assume that: 1) wafers $W_{E(i+1)}$, $i \in \mathbb{N}_{\mathbf{n-1}}$, and all the robot tasks are completed under the normal condition; 2) the processing of wafer W_{Ei}, $i \in \mathbb{N}_{\mathbf{n-1}}$, takes $\upsilon_i \in (a_i, b_i]$ time units; 3) wafers $W_{E(i-1)}$, $W_{E(i-2)}$, …, W_{E1}, …, W_{En+1}, …, and $W_{E(k+1)+1}$ are completed under the normal condition with $i \leq k \leq n-1$; and 4) the processing of wafer W_{Ek+1} takes $\upsilon_{k+1} \in (a_{k+1}, b_{k+1}]$ time units. Then, $\omega_{(k+1)3}^{2} = max\{(\upsilon_{k+1} - \Lambda_{k+1} - \omega_{i3}^{1}), 0\}$.

From Theorem 3.2 and Corollary 3.3, it is easy to verify that if there are delays in processing more than two wafers in a cycle, the accumulated robot waiting time is the largest one among them. From (3.21) through (3.23), it is known that it is more important to obtain the accumulated robot time delay than the robot waiting time at a single place. In fact, Θ_i denotes the accumulated robot time delay. We use γ_{i3}^{j} to denote the accumulated robot time delay when the robot leaves place q_{i3} in the j-th cycle.

Theorem 3.3: Assume that: 1) it takes $\upsilon_i^j \in (a_i, b_i]$ time units to process wafers W_{Ei+j-1}, $j \in \mathbb{N}_{\mathbf{k}}$ and $i \in \mathbb{N}_{\mathbf{n}}$, with $k \leq m_i$ at p_i, respectively; 2) all the robot tasks are completed under the normal condition; and 3) all other wafers before W_{Ei+k-1} are completed under the normal condition. Then,

$$\gamma_{i3}^{k} = max\{0, max\{(\upsilon_i^j - \Lambda_i), j \in \mathbb{N}_{\mathbf{k}}\}\} \tag{3.24}$$

Proof: It follows from Theorem 3.1 that before unloading W_{Ei} the robot waiting time at q_{i3} is ω_{i3}^1. Noticed that there are m_i parallel PMs at p_i, hence, before the robot unloads wafer W_{Ei}, wafers W_{Ei+1}, …, and W_{Ei+k-1} are already in processing. Hence, when the robot arrives at q_{i3} again after a cycle for unloading wafer W_{Ei+1}, wafer W_{Ei+1} has been in p_i for max $\{\upsilon_i^1, \Lambda_i\}$ time units. Thus, if $\upsilon_i^2 \le \upsilon_i^1$, the robot can unload W_{Ei+1} immediately, or $\omega_{i3}^2 = 0$. This means (3.17) holds. If $\upsilon_i^2 > \upsilon_i^1$ we have: 1) if $\upsilon_i^2 \le \Lambda_i$, $\omega_{i3}^2 = 0$; 2) if $\upsilon_i^1 \le \Lambda_i < \upsilon_i^2$, $\omega_{i3}^2 = \upsilon_i^2 - \Lambda_i$; and 3) if $\Lambda_i < \upsilon_i^1$, $\omega_{i3}^2 = \upsilon_i^2 - \upsilon_i^1$. Then, we have $\gamma_{i3}^2 = max\{(\upsilon_i^1 - \Lambda_i), (\upsilon_i^2 - \Lambda_i), 0\}$. When the robot arrives at q_{i3} for the third time for unloading W_{Ei+2}, the situation is similar. By doing so, we know that (3.24) holds. ■

The key for this theorem is that the wafers W_{Ei}, W_{Ei+1}, …, and W_{Ei+k-1} with $k \le m_i$ at p_i are processed in parallel. In fact, γ_{i3}^k in (3.24) presents the accumulated robot waiting time caused by the delays in processing wafers at p_i when $k \le m_i$. To present the following theorem, we define a function f_d as

$$f_d = \begin{cases} 1, d \in N_j \\ 0, otherwise \end{cases}$$

Theorem 3.4: Assume that: 1) it takes $\upsilon_i^j \in (a_i, b_i]$ time units to process wafers W_{Ei+j-1}, $j \in \mathbb{N}_{\mathbf{k}}$ and $i \in \mathbb{N}_{\mathbf{n}}$, with $m_i < k \le 2m_i$ at p_i, respectively; 2) all the robot tasks are completed under the normal condition; 3) all other wafers before W_{Ei+k-1} are completed under the normal condition; and 4) the k wafers are numbered as 1, 2, …, m_i, m_i+1, .., m_i+j, when $k < 2m_i$; and 1, 2, …, m_i, m_i+1, .., $2m_i$, when $k = 2m_i$. Then,

$$\gamma_{i3}^{k} = max\{0, \max\{(\upsilon_i^d - \Lambda_i) + f_d \times (\upsilon_i^{m_i+d} - \Lambda_i), d = 1, 2, \ldots, m_i\}\}, \text{ if } k < 2m_i \tag{3.25}$$

$$\gamma_{i3}^{k} = max\{0, max\{(\upsilon_i^j - \Lambda_i) + (\upsilon_i^{m_i+j} - \Lambda_i), j = 1, 2, \ldots, m_i\}\}, \text{ if } k = 2m_i \tag{3.26}$$

Proof: It follows from Theorem 3.3 that $\gamma_{i3}^{m_i} = max\{(\upsilon_i^1 - \Lambda_i), \ldots, (\upsilon_i^{m_i} - \Lambda_i), 0\}$. Because wafers W_{Ei} and W_{Ei+m_i} are processed by the same PM in a sequential way, after a cycle, when the robot arrives at q_{i3} again for unloading wafer W_{Ei+m_i}, there is a time delay $max\{(\upsilon_i^1 - \Lambda_i) + (\upsilon_i^{m_i+1} - \Lambda_i), 0\}$ for the processing of W_{Ei} and W_{Ei+m_i}. Thus, $\gamma_{i3}^{m_i+1} = max\{0, max\{(\upsilon_i^1 - \Lambda_i) + (\upsilon_i^{m_i+1} - \Lambda_i), ((\upsilon_i^f - \Lambda_i), f = 2, \ldots, m_i)\}\}$. Similarly, we have

$\gamma_{i3}^{m_i+2} = max\{0, max\{(\upsilon_i^1 - \Lambda_i) + (\upsilon_i^{m_i+1} - \Lambda_i), (\upsilon_i^2 - \Lambda_i) + (\upsilon_i^{m_i+2} - \Lambda_i), ((\upsilon_i^f - \Lambda_i), f = 3, \ldots, m_i)\}\}$. Continue this process, we can obtain (3.25) and (3.26). ■

The key here is that wafers W_{Ei}, …, W_{Ei+m_i-1}, and wafers W_{Ei+m_i}, …, W_{Ei+2m_i-1} are processed in parallel by different PMs, respectively, but W_{Ei} and W_{Ei+m_i}, W_{Ei+1} and W_{Ei+m_i+1}, …, and W_{Ei+m_i-1} and W_{Ei+2m_i-1} are sequentially processed by the same PM, respectively. Based on the above discussion, we have the following corollary.

Corollary 3.4: In Theorem 3.4, if $k = d \times m_i$, we have

$$\gamma_{i3}^{k} = max\{0, max\{\sum_{f=0}^{d-1}(\upsilon_i^{j+f\times m_i} - \Lambda_i), j = 1, \ldots, m_i\}\} \tag{3.27}$$

Up to now, we have discussed how to calculate γ_{i3}^{j} that is caused by the delay in processing wafers. Based on the above analysis, we can now discuss how τ_i changes with the activity time variation.

Assume that: 1) when the robot loads wafer W_{Ei} into p_i, there is a time delay ρ_i for executing the loading task; and 2) it takes $\upsilon_i \in (a_i, b_i]$ time units to complete W_{Ei}. Then, in the sense of time delay, it is equivalent that it takes $\upsilon_i + \rho_i$ time units to complete the processing of W_{Ei} with no delay in the loading operation. Thus, it results in a time delay $max\{(\upsilon_i + \rho_i - \Lambda_i), 0\}$. Let $\rho = d - c$ and $H_i = max\{(b_i + \rho - \Lambda_i), 0\}$ be the longest time delay caused by processing a wafer at p_i and $H = max\{H_i, i \in \mathbb{N}_\mathbf{n}\}$ be the largest one. Further let $\eta_{11} = max\{(d_0 - c_0) + (\beta - \alpha) - \omega_{11}, 0\}$, $\eta_{i1} = max\{(d - c) + (\beta - \alpha) - \omega_{i1}, 0\}$, $\eta_{i2} = max\{(\beta - \alpha) - \omega_{i2}, 0\}$ and $\Phi = \sum_{i=0}^{n}(\eta_{i1} + \eta_{i2}) + (n+1) \times \rho$ be the longest robot delay in a cycle. Notice that under the normal condition, H_i, H, and Φ are all constant. To make a schedule feasible, we concern the sojourn time τ_i of a wafer in p_i. The infeasibility of a schedule is caused by the delay of τ_i. After a wafer is loaded into p_i and before the wafer is unloaded from p_i, the robot should perform a series of activities. From (3.11) and (3.18)-(3.23), we determine the sojourn time τ_i of this wafer. Further, since a wafer needs to stay in p_i for m_i robot activity cycles before the robot comes to q_{i3} for unloading it, τ_i is equal to the accumulated time delay $\gamma_{i3}^{m_i}$ in performing these activities. Thus, we have the following result.

Theorem 3.5: If $m_i \leq m_j$ for any $i \neq j$ and $\Phi \leq H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + m_1 \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.28}$$

$$B_i = H + m_i \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i-1)2} + \eta_{(i-2)2} + 2\rho), i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.29}$$

$$B_n = H + m_n \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n-1)2} + \eta_{(n-2)2} + 2\rho) \qquad (3.30)$$

Proof: Assume that the robot has just loaded wafer W_{Li} into p_i. The time delay of any robot activity before the robot loads W_{Li} into p_i does not cause a delay of τ_i for W_{Li}. Hence, after loading W_{Li}, $\gamma^1_{(i-2)3} \le max\{\eta_{(i-2)2}, H_{i-2}\} \le max\{\Phi, H\} \le H$, $\gamma^1_{(i-3)3} \le max\{(H + \eta_{(i-1)1} + \rho + \eta_{(i-3)2}), H_{i-3}\} \le max\{(H + \eta_{(i-1)1} + \rho + \eta_{(i-3)2}), H\} \le H + \eta_{(i-1)1} + \rho + \eta_{(i-3)2}$, for $H + \eta_{(i-1)1} + \rho + \eta_{(i-3)2} \ge H$ must hold. It follows from Theorems 3.2 and 3.3 that we have $\gamma^1_{k3} \le H + \sum_{d=k+2}^{i-1}\eta_{d1} + (i\text{-}2\text{-}k)\rho + \sum_{d=k}^{i-3}\eta_{d2}$, if $0 \le k < i\text{-}3$, $\gamma^1_{k3} \le H + \sum_{d=1}^{i-1}\eta_{d1} + (i\text{-}1)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k}^{n}\eta_{d2}$, if $k = n$, $\gamma^1_{k3} \le H + \sum_{d=0}^{i-1}\eta_{d1} + i \times \rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k}^{n}\eta_{d2}$, if $k = n\text{–}1$, $\gamma^1_{k3} \le H + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1\text{+}n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$, if $i < k \le n\text{-}2$, and $\gamma^1_{i3} \le H + \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i-1)2} + \eta_{(i-2)2} + 2\rho)$. Then, after each robot cycle, when the robot arrives at q_{i3}, the time delay increases by at most Φ time units. In this way, after $m_i - 1$ robot cycles, the robot arrives at q_{i3} again for unloading W_{Li} from p_i. Thus, we can obtain (3.29). Similarly, we can show that (3.28) and (3.30) hold. ■

It follows from (3.29) that a delay in processing a wafer may result in the sojourn time delay of another wafer. It should be noticed that if $m_i > 1$, wafers W_{Ei} and W_{Ei+1} are processed simultaneously at p_i, a delay in processing W_{Ei} will lead to a sojourn time delay for W_{Ei+1}. Thus, in calculating the longest delay of Θ_i given by (3.29), H_i must be calculated as given by $max\{(b_i + \rho - \Lambda_i), 0\}$ such that we can obtain a right H. However, if $m_i = 1$, the PM for Step i processes wafers in a serial way. Thus, the sojourn time delay for a wafer processed at Step i has no effect on the sojourn time for the next wafer processed at Step i. Further, if $m_i = 1$, the accumulated time delay in q_{i3} is between loading a wafer into p_i and unloading the wafer from p_i. By observing the PN model in Fig. 2.1, after firing s_{i1} for loading a wafer into p_i, the following transition firing sequence should be executed before the wafer is unloaded from p_i: $\sigma = \{y_{(i-2)1} \to y_{(i-2)2} \to s_{(i-2)2} \to s_{(i-1)1} \to y_{(i-3)1} \to y_{(i-3)2} \to s_{(i-3)2} \to s_{(i-2)1} \to \ldots \to y_{i1} \to y_{i2}\}$. During this period, the robot does not go to p_{i-1} for unloading a wafer, and a delay in processing wafer $W_{E(i-1)}$ in p_{i-1} has no effect on the sojourn time delay of wafer W_{Ei} in p_i. Hence, to obtain the upper bound of the wafer sojourn time delay at Step i, if $m_i = 1$, $i \in \{\mathbf{N_n} - \{1\}\}$, we should set $H_i = 0$, and $H_{i-1} = 0$ and $H_j = max\{(b_j + \rho - \Lambda_j), 0\}$, $j \ne i$ and $j \ne i\text{-}1$ in calculating H. Similarly, if $m_1 = 1$, to obtain the upper bound of the wafer sojourn time delay at Step 1, we should set $H_1 = 0$ and $H_j = max\{(b_j + \rho - \Lambda_j), 0\}$, $j \ne 1$.

Theorem 3.6: If $m_i \le m_j$ for any $i \ne j$ and $H \le \Phi$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = max\{\eta_{n2}, H\} + m_1 \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.31}$$

$$B_i = max\{\eta_{(i\text{-}2)2}, H\} + m_i \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.32}$$

$$B_n = max\{\eta_{(n\text{-}2)2}, H\} + m_n \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \tag{3.33}$$

Proof: Assume that the robot has just loaded wafer W_{Li} into p_i. We have, $\gamma^1_{(i-2)3} \le max\{\eta_{(i\text{-}2)2}, H_{i\text{-}2}\} \le max\{\eta_{(i\text{-}2)2}, H\}$, $\gamma^1_{(i-3)3} \le max\{max\{\eta_{(i\text{-}2)2}, H\} + \eta_{(i\text{-}1)1} + \rho + \eta_{(i\text{-}3)2}, H_{i\text{-}3}\} \le max\{max\{\eta_{(i\text{-}2)2}, H\} + \eta_{(i\text{-}1)1} + \rho + \eta_{(i\text{-}3)2}, H\} \le max\{\eta_{(i\text{-}2)2}, H\} + \eta_{(i\text{-}1)1} + \rho + \eta_{(i\text{-}3)2}$, for $max\{\eta_{(i\text{-}2)2}, H\} + \eta_{(i\text{-}1)1} + \rho + \eta_{(i\text{-}3)2} \ge H$ must hold. Based on Theorems 3.2 and 3.3, we have $\gamma^1_{k3} \le max\{\eta_{(i\text{-}2)2}, H\} + \sum_{d=k+2}^{i-1}\eta_{d1} + (i\text{-}2\text{-}k)\rho + \sum_{d=k}^{i-3}\eta_{d2}$, if $0 \le k < i\text{-}3$, $\gamma^1_{k3} \le max\{\eta_{(i\text{-}2)2}, H\} + \sum_{d=1}^{i-1}\eta_{d1} + (i\text{-}1)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k}^{n}\eta_{d2}$, if $k = n$, $\gamma^1_{k3} \le max\{\eta_{(i\text{-}2)2}, H\} + \sum_{d=0}^{i-1}\eta_{d1} + i \times \rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k}^{n}\eta_{d2}$, if $k = n\text{-}1$, $\gamma^{m_1}_{k3} \le max\{\eta_{(i\text{-}2)2}, H\} + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1\text{+}n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$, $i < k \le n\text{-}2$, and $\gamma^1_{i3} \le max\{\eta_{(i\text{-}2)2}, H\} + \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho)$. Then, after each robot cycle, when the robot arrives at q_{i3}, γ^1_{i3} increases by at most Φ time units. In this way, after $m_i - 1$ robot cycles, the robot arrives at q_{i3} again for unloading W_{Li} from p_i. Thus, we can obtain (3.32). We can derive (3.31) and (3.33) in the similar way. ■

In Theorems 3.5 and 3.6, we consider just the situation that $m_i \le m_j$ for any $i \ne j$. However, it follows from Theorem 3.4 that if the condition that $m_i \le m_j$ for any $i \ne j$ is not true, (3.28) - (3.33) may not be applicable.

Theorem 3.7: If $m_i \le m_j$, $\forall\ i \ne j$ and $m_i > m_k$, $\forall\ i \ne k$ and $k \ne j$, $\Phi \le H_k$, and $\Phi \le H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + H_k + (m_1 - 1) \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.34}$$

$$B_i = H + H_k + (m_i - 1) \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.35}$$

$$B_n = H + H_k + (m_n - 1) \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \tag{3.36}$$

Proof: Without loss of generality, $i < k \le n\text{-}2$ is assumed. Then, it follows from the proof of Theorem 3.5 that $\gamma^1_{k3} \le H + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1\text{+}n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$. At this time, wafer W_{Lk} is loaded into p_k. Then, after m_k robot cycles, the robot arrives at q_{k3} for unloading W_{Lk}. At this time, the accumulated time delay caused by the robot activities is bounded by $H + m_k \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$. Then, according to Theorem 3.4, the accumulated time delay caused by the processing of W_{Lk} is bounded by $H + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k$. Hence, $\gamma_{k3}^{m_k+1} \le max\{[H + m_k \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}], [H + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k]\}$. Notice that $H + m_k \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} = H + (m_k-1) \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + \Phi \le H + (m_k-1)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k$, we have $\gamma_{k3}^{m_k+1} \le H + (m_k-1) \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k$, for $H + (m_k-1) \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} \ge H + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$. Thus, it is easy to check that $\gamma_{i3}^{m_k+1} \le H + H_k + m_k \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i-1)2} + \eta_{(i-2)2} + 2\rho)$. After m_i - m_k -1 robot cycles, the robot arrives at q_{i3} for unloading W_{Li}. Thus, we can obtain (3.35). It follows from the proof of (3.35) that (3.34) and (3.36) hold. ■

Based on the proof of Theorem 3.7 we have the following theorem.

Theorem 3.8: If $m_i \le m_j$, $\forall\ i \ne j$ and $m_i > m_k$, $\forall\ i \ne k$ and $k \ne j$, $\Phi \ge H_k$, and $\Phi \le H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + m_1 \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.37}$$

$$B_i = H + m_i \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i-1)2} + \eta_{(i-2)2} + 2\rho),\ i \in \{\mathbb{N}_{\mathbf{n}} - \{1, n\}\} \tag{3.38}$$

$$B_n = H + m_n \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n-1)2} + \eta_{(n-2)2} + 2\rho) \tag{3.39}$$

Proof: According to Theorem 3.7, we have $\gamma_{k3}^{m_k+1} \le max\{[H + m_k \times \Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}], [H + \sum_{d=0}^{i-1}\eta_{d1} + (i-1+n-k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k]\}$. By noting that $H +$

$m_k\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} = H + (m_k\text{-}1)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + \Phi \geq H + (m_k\text{-}1)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k$, we have $\gamma_{k3}^{m_k+1} \leq H + m_k\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} + H_k$, for $H + (m_k\text{-}1)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2} \geq H + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}k)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=k+2}^{n}\eta_{d1} + \sum_{d=k}^{n}\eta_{d2}$, and $\Phi \geq H_k$. Then, it is easy to check that $\gamma_{i4}^{m_k+1} \leq H + (m_k + 1)\times\Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho)$. After m_i - m_k -1 robot cycles, the robot arrives at q_{i3} for unloading W_{Li}. Thus, we can obtain (3.38). It follows from the proof of (3.38) that (3.37) and (3.39) hold. ■

Notice that inequalities (3.37) - (3.39) are the same as (3.28) - (3.30), respectively. This implies that this wafer flow pattern has the same effect on the time delay as that in Theorem 3.5.

Theorem 3.9: If $m_i \leq m_j$, $\forall\ i \neq j$, $m_i > m_f$, $\forall\ i \neq f$ and $f \neq j$, and $m_f > m_k$, $\forall\ f \neq k$ and $k \neq i$ and j, $\Phi \leq H_f$, $\Phi \leq H_k$, and $\Phi \leq H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + H_k + H_f + (m_1 - 2)\times\Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \qquad (3.40)$$

$$B_i = H + H_k + H_f + (m_i - 2)\times\Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_{\mathbf{n}} - \{1, \mathrm{n}\}\} \qquad (3.41)$$

$$B_n = H + H_k + H_f + (m_n - 2)\times\Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \qquad (3.42)$$

Proof: Without loss of generality, $i < k < f \leq n$-2 is assumed. Then, it follows from the proof of Theorem 3.5 that $\gamma_{f3}^{1} \leq H + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}f)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=f+2}^{n}\eta_{d1} + \sum_{d=f}^{n}\eta_{d2}$. At this time, wafer W_{Lf} is loaded into p_f. From Theorem 3.7, after m_f robot cycles, the robot arrives at q_{f3} for unloading W_{Lf}. At this time, the accumulated time delay caused by the robot activities and the processing of W_{Lf} is bounded by $H + H_k + (m_f\text{-}1)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}f)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=f+2}^{n}\eta_{d1} + \sum_{d=f}^{n}\eta_{d2}$ and $H + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}f)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=f+2}^{n}\eta_{d1} + \sum_{d=f}^{n}\eta_{d2} + H_f$, respectively. Hence, we have $\gamma_{f3}^{m_f+1} \leq H + H_k + (m_f\text{-}2)\times\Phi + \sum_{d=0}^{i-1}\eta_{d1} + (i\text{-}1+n\text{-}f)\rho + \sum_{d=0}^{i-3}\eta_{d2} + \sum_{d=f+2}^{n}\eta_{d1} +$

$\sum_{d=f}^{n}\eta_{d2} + H_f$ and $\gamma_{i3}^{m_f+1} \leq H + H_k + (m_f\text{-}1) \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho) + H_f$. After m_i - m_k -1 robot cycles, the robot arrives at q_{i3} for unloading W_{Li}. Thus, we can obtain (3.41). It follows from the proof of (3.41) that (3.40) and (3.42) hold. ■

From Theorems 3.6 through 3.9, we have the following corollaries immediately.

Corollary 3.5: If $m_i \leq m_j$, $\forall\ i \neq j$, $m_i > m_f$, $\forall\ i \neq f$ and $f \neq j$, and $m_f > m_k$, $\forall\ f \neq k$ and $k \neq i$ and j, $H_f \leq \Phi$, $\Phi \leq H_k$, and $\Phi \leq H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + H_k + (m_1 - 1) \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.43}$$

$$B_i = H + H_k + (m_i - 1) \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.44}$$

$$B_n = H + H_k + (m_n - 1) \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \tag{3.45}$$

Corollary 3.6: If $m_i \leq m_j$, $\forall\ i \neq j$, $m_i > m_f$, $\forall\ i \neq f$ and $f \neq j$, and $m_f > m_k$, $\forall\ f \neq k$ and $k \neq i$ and j, $H_f \leq \Phi$, $H_k \leq \Phi$, and $\Phi \leq H$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = H + m_1 \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.46}$$

$$B_i = H + m_i \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.47}$$

$$B_n = H + m_n \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \tag{3.48}$$

Corollary 3.7: If $m_i \leq m_j$, $\forall\ i \neq j$, $m_i > m_f$, $\forall\ i \neq f$ and $f \neq j$, and $m_f > m_k$, $\forall\ f \neq k$ and $k \neq i$ and j, and $H \leq \Phi$, the wafer sojourn delay Θ_i in p_i is bounded by

$$B_1 = max\{\eta_{n2}, H\} + m_1 \times \Phi - (\eta_{21} + \eta_{11} + \eta_{02} + \eta_{n2} + 2\rho) \tag{3.49}$$

$$B_i = max\{\eta_{(i\text{-}2)2}, H\} + m_i \times \Phi - (\eta_{(i+1)1} + \eta_{i1} + \eta_{(i\text{-}1)2} + \eta_{(i\text{-}2)2} + 2\rho),\ i \in \{\mathbb{N}_\mathbf{n} - \{1, \mathrm{n}\}\} \tag{3.50}$$

$$B_n = max\{\eta_{(n\text{-}2)2}, H\} + m_n \times \Phi - (\eta_{01} + \eta_{n1} + \eta_{(n\text{-}1)2} + \eta_{(n\text{-}2)2} + 2\rho) \tag{3.51}$$

Up to now, we present the upper bound of wafer sojourn time delay caused by the activity time variation. With such bound, we can check if a given off-line schedule is feasible. This is necessary in practice when the activity time variation exists.

IV. ILLUSTRATIVE EXAMPLES

In this section, examples are presented to show the application of the proposed method to check the feasibility of a given off-line periodic schedule under the control policy.

Example 1: The wafer flow pattern is (1, 1). Under the normal condition, it takes 15 time units for the robot to unload a wafer from a loadlock (c_0 =15) and 10 time units for the robot to load a wafer into a PM or a loadlock, or unload a wafer from a PM (c = 10), 2 time units to move from p_i to p_j (α = 2). It needs 100 time units for a PM at both steps to process a wafer ($a_1 = a_2 = 100$), respectively, and after being processed, a wafer at Steps 1 and 2 can stay there for 20 time units ($\delta_1 = \delta_2 = 20$). The activity times are subject to random variation, and we have $d_0 = 20$, $d = 12$, $\beta = 3$, and $b_1 = b_2 = 104$. Under the normal condition, we have $\psi_1 = 2(n+1)\alpha + (2n+1)c + c_0 = 77$. Further, we have $\vartheta_{1L} = 151$, $\vartheta_{2L} = 146$, $\vartheta_{1U} = 171$, $\vartheta_{2U} = 166$. Let $\vartheta_{Lmax} = \max\{\vartheta_{iL}, i \in \mathbb{N}_{\mathbf{n}}\}$, thus, we have $\vartheta_{Lmax} = 151$. $\psi_1 < \vartheta_{Lmax}$ implies $\psi_2 = \vartheta_{Lmax} - \psi_1 = 74 > 0$. By using the approach presented in [Wu et al., 2008], an off-line schedule with cycle time being 151 can be obtained by setting $\omega_{11} = 0$, $\omega_{21} = 0$, $\omega_{02} = 0$, $\omega_{01} = 0$, $\omega_{12} = 0$, and $\omega_{22} = 74$. Thus, we have $\eta_{01} = 3$, $\eta_{11} = 6$, $\eta_{21} = 3$, $\eta_{02} = 1$, $\eta_{12} = 1$, and $\eta_{22} = 0$, and then $\Phi = 20$. Furthermore, with this schedule, we have $\Lambda_1 = 100$ and $\Lambda_2 = 105$.

To make a schedule feasible (or to make the PN model live) requires that $a_i \le \tau_i \le a_i + \delta_i$, $\forall i \in \mathbb{N}_{\mathbf{n}}$. Notice that $\Lambda_i \le \tau_i$ is always true. Thus, $a_i \le \tau_i$ is so. Based on the results presented in this paper, we have $\tau_i \le \Lambda_i + B_i$. Hence, if, in the worst case, $\Lambda_i + B_i \le a_i + \delta_i$ holds, the wafer residency constraint is never violated, or the schedule is feasible even if the activity times vary. For the example, we have $B_1 + (\Lambda_1 - a_1) = 7 + (100 - 100) = 7 < \delta_1$ and $B_2 + (\Lambda_2 - a_2) = 9 + (105 - 100) = 14 < \delta_2$. This implies that the schedule is feasible.

Its simulation is carried out, and the result shows that it is correct. The evolution of the system is shown in Table 4.1. PM_i and ST denote the processing module for Step i and sojourn time, respectively. Robot activity time (RAT) and processing time (PT) are randomly generated numbers.

Table 4.1. The simulation result for Example 1

NO.	Time(s)	Robot activities	Disturbance to RAT(s)	PT(s)	ST(s)	PM_1	PM_2
1	0-20	Unload W_1 from loadlock	5				W_0
2	20-22	Move to PM_1	0				W_0
3	22-34	Load W_1 to PM_1	2				W_0
4	34-36	Move to PM_2	0			W_1	W_0
5	36-110	Wait at PM_2		103	112	W_1	W_0
6	110-121	Unload W_0 from PM_2	1			W_1	
7	121-124	Move to loadlock	1			W_1	
8	124-135	Load W_0 to loadlock	1			W_1	
9	135-137	Move to PM_1	0	103	103	W_1	
10	137-149	Unload W_1 from PM_1	2				
11	149-151	Move to PM_2	0				
12	151-162	Load W_1 to PM_2	1				
13	162-164	Move to loadlock	0				W_1
14	164-182	Unload W_2 from loadlock	3				W_1
15	182-185	Move to PM_1	1				W_1

16	185-195	Load W_2 to PM_1	0				W_1
17	195-198	Move to PM_2	1			W_2	W_1
18	198-271	Wait at PM_2		101	109	W_2	W_1
19	271-283	Unload W_1 from PM_2	2			W_2	
20	283-286	Move to loadlock	1			W_2	
21	286-297	Load W_1 to loadlock	1			W_2	
22	297-299	Move to PM_1	0	101	104	W_2	
23	299-309	Unload W_2 from PM_1	0				
24	309-311	Move to PM_2	0				
25	311-321	Load W_2 to PM_2	0				
26	321-324	Move to loadlock	1				W_2
27	324-343	Unload W_3 from loadlock	4				W_2
28	343-345	Move to PM_1	0				W_2
29	345-355	Load W_3 to PM_1	0				W_2
30	355-358	Move to PM_2	1			W_3	W_2
31	358-431	Wait at PM_2		104	110	W_3	W_2
32	431-443	Unload W_2 from PM_2	2			W_3	
33	443-445	Move to loadlock	0			W_3	
34	445-457	Load W_2 to loadlock	2			W_3	
35	457-459	Move to PM_1	0	103	104	W_3	
36	459-469	Unload W_3 from PM_1	0				
37	469-471	Move to PM_2	0				
38	471-481	Load W_3 to PM_2	0				
39	481-483	Move to loadlock	0				W_3
40	483-503	Unload W_4 from loadlock	5				W_3
41	503-505	Move to PM_1	0				W_3
42	505-516	Load W_4 to PM_1	1				W_3
43	516-518	Move to PM_2	0			W_4	W_3
44	518-592	Wait at PM_2		104	111	W_4	W_3
45	592-602	Unload W_3 from PM_2	0			W_4	
46	602-605	Move to loadlock	1			W_4	
47	605-617	Load W_3 to loadlock	2			W_4	
48	617-619	Move to PM_1	0			W_4	
49	619-620	Wait at PM_1		104	104	W_4	
50	620-632	Unload W_4 from PM_1	2				
51	632-634	Move to PM_2	0				
52	634-644	Load W_4 to PM_2	0				
53	644-647	Move to loadlock	1				W_4
54	647-662	Unload W_5 from loadlock	0				W_4
55	662-665	Move to PM_1	1				W_4
56	665-676	Load W_5 to PM_1	1				W_4
57	676-678	Move to PM_2	0			W_5	W_4
58	678-752	Wait at PM_2		102	108	W_5	W_4

59	752-762	Unload W_4 from PM_2	0			W_5	
60	762-764	Move to loadlock	0			W_5	
61	764-775	Load W_4 to loadlock	1			W_5	
62	775-778	Move to PM_1	1	100	102	W_5	
63	778-789	Unload W_5 from PM_1	1				
64	789-791	Move to PM_2	0				
65	791-803	Load W_5 to PM_2	2				
66	803-805	Move to loadlock	0				W_5

Example 2: The flow pattern is (1, 1). Under the normal condition, $c_0 = 15$, $c = 10$, $\alpha = 2$, $a_1 = 100$, $a_2 = 80$, and $\delta_1 = \delta_2 = 15$. The activity times are subject to random variations, and we have $d_0 = 20$, $d = 12$, $\beta = 3$, and $b_1 = 105$, $b_2 = 85$. For this example, we have $\psi_1 = 77$. Further, we have $\vartheta_{1L} = \vartheta_{Lmax} = 151$, $\vartheta_{2L} = 126$, $\vartheta_{1U} = 166$, $\vartheta_{2U} = 141$. Clearly, $[\vartheta_{1L}, \vartheta_{1U}] \cap [\vartheta_{2L}, \vartheta_{2U}] = \emptyset$. Thus, an off-line schedule can be obtained by setting $\omega_{11} = 0$, $\omega_{21} = 0$, $\omega_{02} = 0$, $\omega_{01} = 0$, $\omega_{12} = 10$, and $\omega_{22} = 64$ based on [Wu et al., 2008]. Although this schedule is feasible under the normal condition, it is shown infeasible by simulation when the system is subject to activity time variation.

By setting the robot waiting times as $\omega_{11} = 0$, $\omega_{21} = 0$, $\omega_{02} = 0$, $\omega_{01} = 3$, $\omega_{12} = 22$, and $\omega_{22} = 49$, another off-line schedule is obtained. It is easy to check that it is feasible under the normal condition. With this schedule, by using the results presented in this paper, we have $B_1 = 9$ and $B_2 = 9$ such that $B_1 + (\Lambda_1 - a_1) = 9 + (151 - 151) = 9 < \delta_1$ and $B_2 + (\Lambda_2 - a_2) = 9 + (80 - 80) = 9 < \delta_2$. This means that the off-line schedule together with the real-time control policy forms a feasible real-time schedule when the system is subject to activity time variation.

V. CONCLUSIONS

With residency time constraints, it is difficult to operate single-arm cluster tools, especially, when the activity times are subject to random variation. In this paper, to solve this problem, a generic PN model is developed for the system by extending the PN model proposed in [Wu et al., 2008]. With the model, a real-time control policy is proposed by analyzing the properties of the system. With the policy, the activity time variation can successfully be offset as much as possible. Based on it, wafer sojourn time delay caused by the activity time variation is analyzed and upper bound is analytically presented. Thus, such bound can hence be efficiently calculated. It can be easily used to check if a given off-line schedule obtained under deterministic activity times is feasible subject to activity time variation under the proposed Control Policy. For a single-arm cluster tool with wafer residency time constraint and activity time variation, schedulability is a vitally important issue. Further, if schedulable, how the system should be scheduled is also a problem. All of them are open and they are our future work by using the results presented in this paper.

Reference:

[1] M. Bader, R. Hall, and G. Strasser, Integrated processing equipment, *Solid State Technology.*, vol. 33, pp. 149–154, 1990.

[2] P. Burggraaf, Coping with the high cost of wafer fabs, *Semiconduct. Int.*, vol. 38, pp. 45–50, 1995.

[3] W. K. V. Chan, J. Yi, and S. Ding, Optimal scheduling of multicluster tools with constant robot moving times, part I: two-cluster analysis, *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, 5-16, 2011.

[4] A. Che and C. Chu, "Multi-degree cyclic scheduling of two robots in a no-wait flowshop", *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 2, 173-183, 2005.

[5] A. Che and C. Chu, "Cyclic hoist scheduling in large real-life electroplating lines", *OR Spectrum*, vol. 29, no. 3, 445-470, 2007.

[6] H. X. Chen, C. B. Chu and J.-M. Proth, Cyclic scheduling of a hoist with time window constraints, *IEEE Transactions on Robotics and Automation*, vol. 14, 291-318, 1998.

[7] S. Ding, J. Yi, and M. Zhang, Scheduling multi-cluster tools: An integrated event graph and network model approach, *IEEE Transactions Semiconductor Manufacturing*, vol. 19, no. 3, pp. 339-351, Aug. 2006.

[8] D. Jevtic, Method and apparatus for priority based scheduling of wafer processing within a multiple chamber semiconductor wafer processing tool, *US patent* 5,928,389, 1999.

[9] V. Kats, E. Levner, and L. Mayzin, Multiple-part cyclic hoist scheduling using a sieve method, *IEEE Transactions on Robotics and Automation*, vol. 15, 704-713, 1999.

[10] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, Scheduling analysis of timed-constrained dual-armed cluster tools, *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, 521-534, 2003.

[11] J.-H. Kim and T.-E. Lee, Schedule stabilization and robust timing control for timed-constrained cluster tools, in *Proceedings of 2003 IEEE International Conference on Robotics and Automation*, 1039-1044, September, 2003.

[12] J.-H. Kim and T.-E. Lee, Schedulability analysis of timed-constrained cluster tools with bounded time variation by an extended Petri net, *IEEE Trans. on Automation Science and Engineering*, vol. 5, no. 3, 490-503, 2008.

[13] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, Workload balancing and scheduling of a single-armed cluster tool, in *Proceedings of the 5th APIEMS Conference*, Gold Coast, Australia, 1–15, 2004.

[14] T.-E. Lee and S.-H. Park, "An extended event graph with negative places and tokens for timed window constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 4, 319-332, 2005.

[15] D-Y. Liao, M. D. Jeng, and M. C. Zhou, "Petri net modeling and Lagrangian relaxation approach to vehicle scheduling in 300 mm semiconductor manufacturing," in *Proc. 2004 IEEE Conf. Robot. Autom.*, New Orleans, LA, 2004, pp. 5301 – 5306.

[16] M.-J. Lopez and S.-C. Wood, Systems of multiple cluster tools - configuration, reliability, and performance, *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 2, 170-178, 2003.

[17] T. K. McNab, Cluster tools, part 1: emerging processes, Semiconductor Int., vol. 13, no. 9, 58-63, 1990.

[18] T. Murata, Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-580, 1989.

[19] B. Newboe, Cluster tools: a process solution, *Semiconductor Int.*, vol. 13, no. 8, 82-88, 1990.

[20] T. L. Perkinson, P. K. MacLarty, R. S. Gyurcsik, and R. K. Cavin, III, Single-wafer cluster tool performance: An analysis of throughput, *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, 369-373, 1994.

[21] T. L. Perkinson, R. S. Gyurcsik, and P. K. MacLarty, Single-wafer cluster tool performance: An analysis of the effects of redundant chambers and revisitations sequences on throughput, *IEEE Transactions on*

Semiconductor Manufacturing, vol. 9, 384-400, 1996.

[22] S. Rostami, B. Hamidzadeh, and D. Camporese, .An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints, *IEEE Transactions on Robotics and Automation*, vol. 17, 609-618, 2001.

[23] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, Modeling and implementing a real-time scheduler for dual-armed cluster tools, *Computers In Industry*, vol. 45, no. 1, 13-27, 2001.

[24] P. Singer, The driving forces in cluster tool development, *Semiconductor Int*., vol. 18, no. 8, 113-118, 1995.

[25] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, A steady state throughput analysis of cluster tools: Dual-blade versus single-blade robots, *IEEE Trans. Semiconduct. Manuf*., vol. 10, pp. 418–424, 1997.

[26] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 713-723, 1990.

[27] N. Q. Wu, “Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, vol. 29, no. 2, pp. 192-204, 1999.

[28] Wu, N. Q. and M. C. Zhou, “Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems”, *IEEE Transactions on Robotics and Automation*, vol. 17, no.5, 657-668, 2001.

[29] N. Q. Wu and M. C. Zhou, Deadlock Avoidance in Semiconductor Track Systems, in *Proceedings of 2002 IEEE International Conference on Robotics and Automation*, 193-198, Washington D.C., May 2002.

[30] N. Q. Wu and M. C. Zhou, Modeling and deadlock control of automated guided vehicle systems, *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 1, 50-57, 2004.

[31] N. Q. Wu and M. C. Zhou, Modeling and Deadlock Avoidance of Automated Manufacturing Systems with Multiple Automated Guided Vehicles, *IEEE Transactions on Systems, Man, & Cybernetics, Part B*, vol. 35, no. 6, 1193-1202, 2005.

[32] N. Q. Wu and M. C. Zhou, Real-time deadlock-free scheduling for semiconductor track systems based on colored timed Petri nets, *OR Spectrum*, vol. 29, no. 3, 421-443, 2007.

[33] N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, A petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints, *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, No. 2, 224-237, 2008.

[34] N. Q. Wu, M. C. Zhou, S. S. Peng, F. Chu, and C. B. Chu, Petri net modeling and real-time control of dual-arm cluster tools with residency time constraint and activity time variations, in *Proceedings of the 4th IEEE Conference on Automation Science and Engineering*, 109-114, Washington DC, USA, August 23-26, 2008.

[35] N. Q. Wu and M. C. Zhou, A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis, *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, 303-315, 2010a.

[36] N. Q. Wu and M. C. Zhou, Colored timed Petri nets for modeling and analysis of cluster tools, *Asian Journal of Control*, vol. 12, no. 3, 253-266, 2010b.

[37] N. Q. Wu and M. C. Zhou, Analysis of wafer sojourn time in dual-arm cluster tools with residency time constraint and activity time variation, *IEEE Transactions on Semiconductor Manufacturing*, vol. 23, no. 1, 53-64, 2010c.

[38] N. Q. Wu and M. C. Zhou, Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation, *IEEE Transactions on Automation Science and Engineering*, in press, 2011.

[39] J. Yi, S. Ding, and M. Zhang, Steady-state throughput and scheduling analysis of multi-cluster tools: A

decomposition approach, *IEEE Trans. Automation Sci. Eng*., vol. 5, no. 2, pp. 321-336, Apr. 2008.

[40] H. J. Yoon and D. Y. Lee, On-line scheduling of integrated single-wafer processing tools with temporal constraints, *IEEE Transactions on Semiconductor Manufacturing*, vol. 18, no. 3, 390-398, 2005.

[41] M. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 4, pp. 515-527, 1991.

[42] M. Zhou, F. DiCesare, and A. Desrochers, "A hybrid methodology for synthesis of Petri nets for manufacturing systems," *IEEE Trans. Robotics and Automation*, vol. 8, 350-361, 1992.

[43] M. C. Zhou and M. D. Jeng, Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: a Petri net approach, *IEEE Transactions on Semiconductor Manufacturin*g, vol. 11, no. 3, 333-357, 1998.

[44] M. C. Zhou and K. Venkatesh, *Modeling, simulation and control of flexible manufacturing systems: Petri net approach*, World Scientific, Singapore, 1998.

[45] W. M. Zuberek, Timed Petri nets in modeling and analysis of cluster tools, *IEEE Trans. Robot. Automat*., vol. 17, no. 5, pp. 562-575, Oct. 2001.