Characterizing Token Delays of Timed Event Graphs for $K$-Cyclic Schedules

Tae-Eog Lee, Hyun-Jung Kim, Dong-Hyun Roh and Ramavarapu S. Sreenivas

Abstract—A timed discrete event system, which repeats identical work cycles, has task delays due to synchronization between work cycles. Real such systems tend to operate mostly in a $K$-cyclic timing regime, where a sequence of identical timing patterns is repeated for every $K$ cycles. Therefore, the task delays fluctuate and repeat a sequence of $K$ different values, and hence have higher risk of violating an upper limit. Task delays correspond to token delays at the system's timed event graph model. We therefore examine token delays in $K$-cyclic schedules of a timed event graph, an essential class of Petri nets. We first identify all possible $K$-cyclic schedules and define their initial phases. We then develop a closed-formula on the token delays on a path for a $K$-cyclic schedule, which can be computed by the longest path lengths between the nodes in an associated directed graph. We also present a formula for $1$-cyclic schedules. The formulae can be used for computing statistics on $K$-different token delays, maximizing or minimizing the token delays with regard to all possible initial phases, and verifying task delay constraints, if any.

Index Terms—token delay, task delay, timed event graph, Petri net, $k$-cyclic schedule, discrete event system.

I. INTRODUCTION

Timed event graphs (TEGs) are a class of timed Petri nets that have only one input and one output transitions for each place. They are used for modeling and analyzing discrete event systems, which have no decision for the next event choice and hence repeat identical work cycles for each resource. Examples are robotized cluster tools for semiconductor manufacturing [1]-[15], hoist-based manufacturing systems [16], [17], railway systems [18], asynchronous circuits [19], etc. Such a system is subject to task delays due to synchronization between the work cycles of the resources. Task delays, which cause waiting of service entities or resources, can be detrimental to system performance and quality. For instance, in a cluster tool for semiconductor manufacturing, a wafer processed at a chamber should wait within the chamber until it is unloaded by a robot. Such delays cause wafer quality degradation and variation or even quality failures due to residual gases and heat within the chamber. Such task delays correspond to token delays at a place or on a path between a pair of transitions in a TEG model for the system. The delay of a token at a place is the waiting time of the token until the token is removed from the place by firing of the place’s output transition after the required token holding time at the place has elapsed. Most real discrete event systems start a task or activity as soon as possible. Unless the timings are deliberately controlled, they tend to operate mostly in a $K$-cyclic timing regime, where a sequence of identical timing patterns is repeated for every $K$ cycles. Therefore, the task delays fluctuate around the average and repeat a sequence of $K$ different values. Such increased variability of the task delays increases the risk of violating an upper limit and causes process quality variation. We are therefore interested not only in such varying token delay values, the extreme values, and the variance, but also in how token delays occur and what affects or determines token delays.

To do this, we examine a TEG that fires transitions as soon as possible. The firing epochs of the transitions of a TEG are called a schedule. A $K$-cyclic schedule repeats an identical timing pattern every $K$ cycles. A firing schedule of a strongly connected TEG reaches a $K$-cyclic schedule after a finite number of cycles regardless of the initial sojourn times of tokens at the places called initial lags [2], [14], [20], [22], [29]. The cyclicity $K$ is the same as the least common multiple of the greatest common divisors of the number of tokens in each set of critical circuits that are connected by sharing one or more transitions (Section 3.7.1 of [22]). We may choose the initial lags carefully to make the asymptotic $K$-cyclic schedule $1$-cyclic. Otherwise, the asymptotic $K$-cyclic schedule repeats $K$ distinct timing patterns for each $K$ cycles. In fact, as exemplified for automated manufacturing systems in [20], [29], real discrete event systems operate mostly in a $K$-cyclic schedule unless the initial lags are deliberately chosen or controlled. The initial lags are unknown or not controllable due to initial transient or partial work cycles, or sporadic time disruptions.

There have been numerous works on computing task or token delays and existence of a feasible schedule against constraints on task delays, including [1], [2], [3], [4], [6], [7], [8], [9], [10], [11], [12], [14], [15], [16], [17], [24], [25], [26], [27]. However, most of them consider $1$-cyclic schedules. In order to regulate the task delays not to exceed a limit, a feedback controller postpones some tasks or events intentionally until their associated tasks or events occur and some additional time elapses. Such feedback controllers for TEGs have been developed using the max-plus algebra the-
ory [12], [15], [25], [26], [27]. Houssin et al. [25], [26] design a feedback controller to satisfy constraints on token delays on paths in a TEG. However, we yet wish to more directly compute token delays, especially for $K$-cyclic schedules, and understand why and how they occur.

In this paper, we develop a closed-form formula of token delays on a path between a pair of transitions in a TEG, where the schedule is $K$-cyclic. We characterize token delays for a given $K$-cyclic schedule. We also identify all possible $K$-cyclic schedules and define their initial phases. We then develop a closed-formula on the token delays on a path for a $K$-cyclic schedule with a given initial phase, which can be computed by the longest path lengths between the nodes in an associated directed graph. We also present a formula for 1-cyclic schedules. The formulas can be used for computing statistics on $K$-different token delays, maximizing or minimizing the token delays with regard to all possible initial phases, and verifying task delay constraints, if any. We can evaluate delay performance for a TEG with a feedback controller. We also can improve or design a TEG structure for better delay performance.

II. Token Delays in Timed Event Graphs

A. Timed Event Graph

A formal definition of a TEG follows.

**Definition 1 (Timed Event Graph):** A timed event graph is formally defined as a 6-tuple $\mathcal{N} = (P, T, I, O, M_0, H)$, where:

1. $T = \{t_i | i \in T\}$ is a finite nonempty set of transitions, where $T$ is a finite subset of natural numbers such that $i = 1, 2, \ldots, |T|$. $\mathcal{N}_r = \bigcup_{r=1,2,\ldots}^{\infty}\{t_i | i \in T\} \cup \{t_0\}$ and $\mathcal{N} = \{t_i | i \in T\} \cup \{t_0\}$.
2. $P = \{p_{ij} | (i, j) \in P\}$ is a finite nonempty set of places, each between a pair of transitions, where $P$ is a subset of ordered pairs of $(i, j), i, j \in T$.
3. Function $I: (P \times T) \rightarrow \{0, 1\}$ indicates the existence of a directed arc from a place to a transition.
4. Function $O: (P \times T) \rightarrow \{0, 1\}$ defines the existence of a directed arc from a transition to a place.
5. Function $M_0: P \rightarrow Z^+$ determines the initial number of tokens at the places in the circuit. The maximum circuit ratio among all circuits is called the critical circuit ratio and the circuit with the maximum ratio and the transitions in the circuit are called critical. It is also known that the minimum average cycle time among all feasible firing schedules of a TEG is the same as the critical circuit ratio of the TEG.
6. Function $H: P \rightarrow R^+$ defines the required token holding time at each place, $H(p_{ij}) = h_{ij}$, where $R^+$ is the set of nonnegative integers.
7. $h_{ij}$ is defined as the required token holding time at each place, $H(p_{ij}) = h_{ij}$, where $R^+$ is the set of nonnegative integers. A token that enters place $p_{ij}$ becomes available for enabling the output transition after $h_{ij}$ time units.
8. $\forall p_{ij} \in P$, there is only one pair of transitions $t_i, t_j \in T$ such that $I(p_{ij}, t_i) = 1$ and $O(p_{ij}, t_j) = 1$.

For a place $p_{ij}$, $t_i$ and $t_j$ are called its input and output transitions, respectively. A transition $t_j$ is enabled if each place $p_{ij} \in \{p_{ij} | I(p_{ij}, t_j) = 1\}$ has at least one available token. We assume that a transition has no firing delay and hence fires as soon as it is enabled. When an enabled transition fires, a token is removed from each preceding place and a token is added to each succeeding place. We do not intentionally delay the firing of a transition. In most of the literature this policy is called earliest firing policy and a schedule with such timing control is called an earliest schedule. We assume the earliest firing policy. Fig. 1 illustrates a TEG.

A TEG is called strongly connected if there exists a path from $t_i$ to $t_j$ for any ordered pair of transitions $t_i$ and $t_j$. A strongly connected TEG is live, that is, every transition can fire arbitrarily often, if and only if every circuit has a token.

In the sequel, we consider strongly connected and live TEGs. Under the earliest firing policy, a TEG reaches, after some transient period, a periodic regime of a firing schedule, during which all transitions fire the same number of times and the firing schedule repeats identical timing patterns. As the time goes, the length of such a periodic time divided by the number of times each transition fires converges to a finite value, called average cycle time. For a circuit, the circuit ratio is the ratio of the sum of the token holding times at the places to the number of tokens at the places in the circuit. The maximum circuit ratio among all circuits is called the critical circuit ratio and the circuit with the maximum ratio and the transitions in the circuit are called critical. It is also known that the minimum average cycle time among all feasible firing schedules of a TEG is the same as the critical circuit ratio of the TEG.

B. K-cyclic schedules

The collection of the transition firing epochs, $\{x^r_i | i \in T, r = 1, 2, \ldots\}$, is called a schedule, where $x^r_i$ indicates the $r$-th firing epoch of transition $t_i$. For a $K$-cyclic schedule, $x^{r+K}_i = x^r_i + K\lambda \forall i \in T, r = 1, 2, \ldots$, where $\lambda$ is the average cycle time. $\lambda$ is known to be the critical circuit ratio of the TEG.

When a critical circuit $\pi$ has $\tau_\pi$ tokens, each transition in the circuit fires $\tau_\pi$ times during a cycle. It implies that $\tau_\pi$ copies of circuit $\pi$ are simultaneously in progress. An earliest schedule is determined by the first firing epochs for the critical circuits. We let $C = \{t^k_i | i \in T, r = 1, 2, \ldots\}$, where $t^k_i$ is the $k$-th firing instance of $t_i$.

Suppose that a schedule $\{x^r_i | i \in T, r = 1, 2, \ldots\}$ has entered a cyclic timing regime. For such a cyclic schedule, the initial phase is defined as $\delta \equiv \{\delta_k | \delta_k = x^r_i, t^k_i \in C\}$. As shown in a $K$-cyclic schedule in Fig. 2, a critical circuit $\pi$ generates $\tau_\pi$ distinct sequences of firing epochs in the firing schedule. The initial phase $\delta$ indicates the relative timing differences among such sequences that are generated by the critical circuits. We define a schedule graph and token delays in Definition 2 and Definition 3, respectively.

**Definition 2 (Schedule Graph $G=(\mathcal{N}, \mathcal{A})(\delta)$):** For a given TEG, we define an infinite graph $G=(\mathcal{N}, \mathcal{A})(\delta)$ such that $\mathcal{N} = \bigcup_{r=1,2,\ldots}^{\infty}\{t^r_i | i \in T\} \cup \{t_0\}$ and $\mathcal{A}$ =

![Fig. 1. A TEG.](image-url)
two critical paths. The schedule depends on the initial phases $\delta_i$. Such a graph, $G=(\bar{N}, \bar{A})(\delta)$, is called a schedule graph of the TEG.

**Definition 3 (Token Delay):** For a given firing schedule $\{x_i \mid i \in T, r = 1, 2, \ldots\}$,

1) the $r$-th token delay at a place $p_{ij}$ is $d_{ij}^r = (x_j^r + \tau_{ij}) - x_i^r$,

2) the token delays at a path $\psi(i,j)$ are $d_{\psi(i,j)}^r = (x_j^r + \tau_{ij}) - h_{\psi(i,j)}, r = 1, 2, \ldots$, where $h_{\psi(i,j)}$ and $\tau_{\psi(i,j)}$ are the sum of the token holding times of the places and the sum of the number of tokens of the places in the path, respectively, and

3) the token delay in a circuit is the sum of the token delays at the places in the circuit.

**Example 1 (K-Cyclic Schedule):** Fig. 2 demonstrates the schedule graph and a K-cyclic schedule for Fig. 1, where $K = 2$. $t_0$ is a dummy start node. The firing instances are marked at the firing epochs. The critical circuits correspond to the critical paths with bold arcs in the schedule graph. Since the critical circuit $(t_5 \rightarrow t_6 \rightarrow t_3)$ has two tokens, it generates two critical paths. The schedule depends on the initial phases of the critical paths. Once the initial phase $\delta$ is given, the longest path lengths from the start node $t_0$ to a node $t_i$ in the schedule graph determine a K-cyclic schedule. The solid head part and dotted tail part of each arc indicate the token holding time and the token delay at the corresponding place, respectively. The number near a dotted part is the value of the token delay.

**Lemma 1 (Feasible Initial Phases):** Any finite real-valued initial phase $\delta$ defines a K-cyclic schedule if the schedule graph starts each node as soon as possible while allowing no delay between the nodes of a critical circuit.

**Proof:** In a schedule graph, we arbitrarily shift a critical path as a solid object back or forth while keeping no delay on the path. Then, it is seen that any other critical path, which has a path from or to the critical path, also can shift accordingly while maintaining no delay on the path. A non-critical path also shifts accordingly and are aligned to the critical paths even though the delays on the path may change. Therefore, the resulting schedule maintains K-cyclic even if the starting times are shifted back or forth. The initial phases change according to the shifts. It implies that any finite real values of the initial phases $\delta_i$’s define a K-cyclic schedule.

**III. TOKEN DELAYS IN K-CYCLIC SCHEDULES**

We now characterize the token delays in K-cyclic schedules.

**Theorem 1 (Token Delays in a K-Cyclic Schedule):** Consider a strongly connected and live TEG with critical circuit ratio $\lambda$. A K-cyclic schedule of the TEG, $\{x_i \mid x_i + K = x_i + K \forall i \in T, \forall r = 1, 2, \ldots\}$, has the following properties.

1) The token delays on a path $\psi(i,j)$ between a pair of transitions $t_i$ and $t_j$ repeat $K$ values of

$$d_{\psi(i,j)}^r = \left[ \frac{k + r + \tau_{\psi(i,j)}}{K} - \frac{K\lambda - h_{\psi(i,j)}}{K} \right] K - x_i^r \quad \forall k = 1, 2, \ldots, K.$$  (1)

2) The average token delay on a path $\psi(i,j)$ between a pair of transitions $t_i$ and $t_j$ is the same as

$$\bar{d}_{\psi(i,j)} = \tau_{\psi(i,j)}\lambda - h_{\psi(i,j)} + \frac{\sum_{k=1}^{K} (x_i^k - x_i^k)}{K}.$$  (2)

3) The average token delay for a cycle along a circuit $\pi$ is always the same as

$$\bar{d}_{\pi} = \tau_{\pi}\lambda - h_{\pi}.$$  (3)

4) The token delay for a cycle along a critical circuit $\pi$ is always zero.

**Proof:** For any firing schedule $\{x_i \mid i \in T, r = 1, 2, \ldots\}$, the token delays on a path $\psi(i,j)$ are $d_{\psi(i,j)}^r = x_i^r - x_j^r$.

For a K-cyclic schedule such that $x_i^r + K = x_i^r + K \forall i \in T, \forall r = 1, 2, \ldots, (x_j^r + K - x_i^r - \tau_{\psi(i,j)}) = (x_j^r - x_i^r - \tau_{\psi(i,j)})$ $\forall i,j \in T, \forall r = 1, 2, \ldots$. Therefore, $d_{\psi(i,j)}^K = d_{\psi(i,j)}^r$ $\forall i,j \in T, \forall r = 1, 2, \ldots$. Consequently, we have $d_{\psi(i,j)}^r = x_i^r + K - \tau_{\psi(i,j)} - x_i^r$ $\forall k = 1, 2, \ldots, K$. Since a K-cyclic schedule repeats an identical timing pattern for each $K$ cycles, $x_j^r$ $\tau_{\psi(i,j)}$ is $x_i^r$ $\tau_{\psi(i,j)}$ $[\frac{K r^r - \tau_{\psi(i,j)}}{K}] K + \left[ \frac{K r^r - \tau_{\psi(i,j)}}{K} - 1 \right] K\lambda$. By subtituting this to the delay formula, we obtain the token delays of equation (1).

By summing up the $K$ consecutive token delays and dividing the sum by $K$, we have $\sum_{k=1}^{K} d_{\psi(i,j)}^r = \frac{\sum_{k=1}^{K} (x_i^r + K - x_i^r - \tau_{\psi(i,j)})}{K} - h_{\psi(i,j)} \forall i \in T, \forall r = 1, 2, \ldots$. This can also be written as

$$\sum_{k=1}^{K} (x_i^r + K - x_i^r - \tau_{\psi(i,j)}) = \sum_{k=1}^{K} (x_i^r + K - x_i^r - \tau_{\psi(i,j)}) + \sum_{k=1}^{K} (x_i^r + K - x_i^r - \tau_{\psi(i,j)}) = \tau_{\psi(i,j)}\lambda + \frac{K r^r - \tau_{\psi(i,j)}}{K}. \lambda$$

The last equality is used because the schedule repeats an identical timing pattern every $K$ cycles.

$$\sum_{k=1}^{K} (x_i^r + K - x_i^r - \tau_{\psi(i,j)}) = \tau_{\psi(i,j)}\lambda + \frac{K r^r - \tau_{\psi(i,j)}}{K}.$$
When \( t_i \) and \( t_j \) are in a circuit \( \pi \), it is seen that 
\[
\sum_{k=1}^{K} (x_i^k - x_j^k) = 0 \quad \text{and hence we have equation (3). If}
\]
the circuit \( \pi \) is critical, 
\[
x_i^{k+\tau_{\psi(i,j)}} - \frac{k+\tau_{\psi(i,j)}-1}{\lambda} K - x_i^k = x_i^{k+\tau_{\psi(i,j)}} K - x_i^k = 0, \tau_\pi \lambda - \pi_\pi = 0, \text{and hence } d_{\psi(i,j)}^k = 0.
\]

Because of K-cyclicity, it suffices to consider the schedule 
for only K cycles, \( \{x_i^k \mid i \in \mathcal{T}, k = 1, 2, \ldots, K \} \). From 
the proof of Theorem 1, we have the following precedence relations:
\[
x_j^{k+\tau_{\psi(i,j)}-\frac{k+\tau_{\psi(i,j)}-1}{\lambda} K} - x_i^k \geq h_{ij} - \frac{k+\tau_{\psi(i,j)}-1}{\lambda} K \lambda
\]
\forall (i, j) \in \mathcal{P}, \forall k = 1, 2, \ldots, K. \quad (4)

Now, we will show that a K-cyclic schedule and its token 
delays can be directly computed from the following compact 
directed graph.

Definition 4 (K-Cyclic Schedule Graph): For a TEG with 
the critical circuit ratio \( \lambda \) and the cyclicity \( K \), we define a 
directed graph \( G^K = (N^K, A^K)(\delta) \) from precedence relations (4) 
such that the node set is \( N^K = \{t_i^k \mid i \in \mathcal{N}, k = 1, 2, \ldots, K \} \) \cup 
\( \{0\} \) and the arc set is \( A^K = \{(t_i^k, t_j^l) \mid k+\tau_{\psi(i,j)}-\frac{k+\tau_{\psi(i,j)}-1}{\lambda} K \} \) has 
length \( h_{ij} - \frac{k+\tau_{\psi(i,j)}-1}{\lambda} K \lambda \). \( t_0 \) is a dummy start node. The initial 
phase \( \delta_0 \) is given to the nodes in \( C \). \( D \) is the set of dummy 
archs with length \( \delta_0 \) from \( t_0 \) to a node \( t_i^k \in C \).

Example 2 (K-Cyclic Schedule Graph): Consider the TEG 
in Fig. 1. The two critical circuits \( \tau_{\pi_1} = (t_3 \rightarrow t_4 \rightarrow t_3) \) 
and \( \tau_{\pi_2} = (t_5 \rightarrow t_6 \rightarrow t_5) \) do not share any transition. \( \tau_{\pi_1} \) and 
\( \tau_{\pi_2} \) are 2. Therefore, \( K = 2, C = \{t_4^1, t_5^1, t_2^2 \} \). Fig. 3 illustrates 
the K-cyclic schedule graph.

Fig. 3. K-cyclic schedule graph \( G^K = (N^K, A^K)(\delta) \).

Let \( \gamma_{0_{ij}}(\delta) \) denote the longest path length from \( t_0 \) to a 
node \( t_i^k \in N^K \). \( \gamma_{0_{ij}}(\delta) = \max_{q \in C}(\delta_{ij} + \gamma_{q_{ij}^{k+1}}) \)'s in 
the delay formula can be computed by a longest path algorithm, 
Bellman-Ford algorithm in \( O(|\mathcal{T}| |\mathcal{P}|) \).

We discuss how all possible K-cyclic schedules are 
defined and they are related to initial phasors. The max-plus 
algebra has the two algebraic operations (\( \oplus, \otimes \)) such that 
\( a \oplus b = \max(a, b) \) and \( a \otimes b = a + b \), respectively [22]. 
The operations also can be defined for matrices and vectors. 
For notational convenience, we indicate \( k \)'s by single symbols 
such as \( p, q, e \), etc., if necessary. We let matrix \( B \) be the 
incidence matrix of a K-cyclic schedule graph, excluding the 
start node \( t_0 \) and its connected arcs, such that an element of 
\( B, B_{pq} \), is the length of the arc from node \( q \) to node \( p \) in 
the K-cyclic schedule graph if there is such an arc, and \( e \) (that is, 
\( -\infty \)) otherwise. We let \( B^+ \equiv B \oplus B^\oplus \oplus \cdots B^\oplus \) and \( B^\oplus_2 \) 
be the column vector of \( B^+ \) corresponding to node \( q \).

Theorem 2 (All Possible K-Cyclic Schedules): For any finite 
initial phase \( \delta \), vector \( x = \max_{t_i \in C}(\delta_{ij} + B^+_{ij}) \) defines a 
K-cyclic schedule.

Proof: By Lemma 1, any finite initial phase \( \delta \) defines a 
K-cyclic schedule in the schedule graph with initial phase \( \delta \). 
The first K cycles of such a K-cyclic schedule are computed by 
\( x_i^k = \gamma_{0_{ij}}(\delta) = \max_{t_i \in C}(\delta_{ij} + \gamma_{ij}^{k+1}) \). An eigenvector \( x \) of 
matrix \( B \) such that \( B \otimes x = \lambda \otimes x \) defines a K-cyclic schedule. 
Since the maximum circuit ratio of \( B \) is 0, from Theorem 3.100 of [22], 
yany linear combination of the column vectors of \( B^+, \max_{t_i \in C}(\delta_{ij} + B^+_{ij}) \), 
is an eigenvector \( B \). Consequently, 
\( B^+_{ij} = \gamma_{ij}^{k+1} \), vector \( x = \max_{t_i \in C}(\delta_{ij} + B^+_{ij}) \) defines a 
K-cyclic schedule.

We may wish to know whether a TEG with given initial 
lags (that is, initial token sojourn times) can reach such a 
K-cyclic schedule and whether any initial phase can be reached 
from a feasible initial lags. However, they are beyond our 
scope. We now show that for a given initial phase, a K-cyclic 
schedule and its token delays can be computed from the 
K-cyclic schedule graph.

Theorem 3 (Token Delays in K-Cyclic Schedules): 
Consider a strongly connected and live TEG with critical 
circuit ratio \( \lambda \). A K-cyclic schedule of the TEG, 
\( \{x_i^k \mid i \in \mathcal{T}, r = 1, 2, \ldots \} \), has the following properties.

1) \( \{x_i^k \mid i \in \mathcal{T}, k = 1, 2, \ldots, K \} \) defines an 
earliest K-cyclic schedule with cycle time \( \lambda \), 
\( x_i^{k+(r-1)K} = x_i^k + (r-1)K \lambda \forall i \in \mathcal{T}, k = 1, 2, \ldots, K, \forall r = 1, 2, \ldots \) 
for any real-valued initial phase \( \delta \).

2) For a K-cyclic earliest schedule with an initial phase \( \delta \), 
the token delays on a path \( \psi(i, j) \) in the TEG have \( K \) 
different values:
\[
d_{\psi(i, j)}^k(\delta) = \gamma_{ij}^{k+(r-1)}(\delta) + \gamma_{ij}^{k+\tau_{\psi(i,j)}-1}(\delta) - \gamma_{0_{ij}}(\delta)
\]
\[
\frac{k+\tau_{\psi(i,j)}-1}{K} K \lambda \psi(\psi(i, j)) \quad \forall k = 1, 2, \ldots, \delta.
\] \quad (5)

Proof: In a K-cyclic schedule \( \{x_i^{k+(r-1)K} \mid x_i^{k+(r-1)K} = x_i^k + (r-1)K \lambda \forall i \in \mathcal{T}, k = 1, 2, \ldots, K, \forall r = 1, 2, \ldots \} \), 
the precedence relations between the firing epochs, 
\( x_i^{k+\tau_{ij}} - x_i^k \geq h_{ij} \forall i, j \in \mathcal{T}, \forall r, \) are equivalent to 
\[
x_j^{k+\tau_{ij}} - x_i^k \geq h_{ij} - \frac{k+\tau_{ij}-1}{K} K \lambda
\]
\forall (i, j) \in \mathcal{P}, \forall k = 1, 2, \ldots, K. \quad (6)

By summing up the constraints for arcs \( (t_i, t_j) \) along a critical 
circuit \( \pi \), we have \( \sum_{(t_i, t_j) \in \pi}(h_{ij} - \tau_{\pi_\pi}) = \sum_{(t_i, t_j) \in \pi} h_{ij} - \sum_{(t_i, t_j) \in \pi} \tau_{\pi_\pi} = 0 \). Therefore, \( G^K = (N^K, A^K)(\delta) \) has no 
circuit of positive length and hence the longest path length 
between a pair of nodes is finite. In a K-cyclic schedule graph 
\( G^K = (N^K, A^K)(\delta) \), the longest path length \( \gamma_{ij}^{k+1}(\delta) \) is the 
earliest possible time of node \( t_i^k \) that satisfies the precedence
relations (6). Consider a schedule \( \{ x_t^k \}_{t \in \mathcal{T}, k = 1, 2, \ldots, K} \) in which

\[
x_t^k = \left( \gamma_{0t}^k (\delta) + (r - 1) K \right) \lambda \quad \forall t \in \mathcal{T}, k = 1, 2, \ldots, K, \forall r \in 1, 2, \ldots.
\]

Then, the schedule satisfies all precedence constraints \( x_t^k \geq h_{ij} \) \( \forall i, j \in \mathcal{T}, \forall r \). It is because once they are plugged in by \( x_t^{k+1} = \gamma_{0t}^k (\delta) + (r - 1) K \lambda \), they are reduced to (6). Therefore, the schedule is feasible.

We now show that the schedule is earliest. Suppose that the schedule is not earliest. Then, for some \( t_j^k \), the longest path length from \( t_0 \) to \( t_j^k \) in the schedule graph has length shorter than \( \gamma_{0j}^k (\delta) + (r - 1) K \lambda \). We contradict this claim by showing that there exists a path from \( t_0 \) to \( t_j^k \) with length no less than \( \gamma_{0j}^k (\delta) + (r - 1) K \lambda \). Consider a node \( t_j^k \). Let \( \pi \) be a critical circuit in the TEG that includes transition \( t_j \). Since the TEG is strongly connected, so \( G = (N^K, A^K) \) is any path in the TEG has the corresponding path in \( G = (N^K, A^K) \) and vice versa, where the two paths have equivalent sequences of transition firings. Therefore, for a path \( \psi(i, j) \) in the TEG, we can have a corresponding path in \( G = (\mathcal{N}, \mathcal{A}) \). We observe that for a critical circuit \( \psi(i, j) \) in the TEG, \( \psi(i, j) \) is an equality along the longest path, and hence we have

\[
x_j^{k+\tau_\psi(i,j)-1} + (r - 1) - K \lambda \leq x_j^k = h_{\psi(i,j)} - K \lambda.
\]

There exists a path \( \psi(0, j, k+(r-1)K-\tau_\psi(i,j)) \) in the schedule graph \( G = (\mathcal{N}, \mathcal{A}) \) that circulates around critical circuit \( \pi \) and has length \( \gamma_{0j}^k \). It is because in a \( K \)-cyclic schedule, the initial phase \( \delta \) is accommodated to ensure \( x_j^{k+\tau_\psi(i,j)-1} - x_j^k = (r - 1) K \lambda \) and there is no slack time between the transition firings along path \( G = (\mathcal{N}, \mathcal{A}) \), which circulates around critical circuit \( \pi \). The schedule graph also has a path \( \psi(0, j, k+(r-1)K-\tau_\psi(i,j)) \) with length \( h_{\psi(i,j)} \), which is equivalent to path \( \psi(i, j) \) in the TEG. By combining the two paths, we have path \( \psi(0, j, k+(r-1)K) \) with length \( \gamma_{0j}^k + (k + (r - 1) K - \tau_\psi(i,j)) \). Consequently, the length of path \( \psi(0, j, k+(r-1)K) \) is the same as \( \gamma_{0j}^k + (k + (r - 1) K - \tau_\psi(i,j)) \). Therefore, we reach a contradiction and hence the schedule is earliest. Then, the token delays \( d_k^{\psi(i,j)} \)’s come from Theorem 1.

We now explain an important application to identify the worst-case wafer delays within a processing chamber of a robotized cluster tool [1], [2], [3], [4], [6], [7], [12], [14], [15].

**Example 3 (Maximum Token Delays):** Consider \( K \)-cyclic schedule graph \( G = (N^K, A^K) \) in Fig. 3. We see \( C = \{ t_1^1, t_1^r \} \). We compute \( B_{15}^+ = -7, B_{34}^+ = 3, B_{54}^+ = -4, B_{43}^+ = 2, B_{24}^+ = 1 \). Consequently, an eigenvector \( x \) is computed as \( x_{y_1} = \max (y_1, y_1 - 7, y_{y_2} - 4), x_{y_2} = \max (y_1 + 3, y_{y_2} + 1), \) and \( x = \max (y_1 + 2, y_1 - 3, y_{y_2}) \). Then, the feasible ranges of \( \delta_{y_2} \) and \( \delta_{y_1} \) are enumerates to be [3, 7] and [2, 4], respectively, where \( \delta_{y_2} \leq 1 + \delta_{y_2} \leq \delta_{y_1} + 3 \). From the \( K \)-cyclic schedule graph, we compute \( \gamma_{23}^1 = 4, \gamma_{21}^3 = 9, \gamma_{51}^1 = 0, \gamma_{51}^3 = 3, \gamma_{41}^3 = 2, \gamma_{12}^3 = 7, \gamma_{41}^1 = 4, \gamma_{41}^2 = 1, \gamma_{51}^2 = -1, \gamma_{52}^2 = 4, \gamma_{52}^1 = -7, \gamma_{52}^1 = 0 \). By fixing \( \delta_{y_2} = 5 \) and searching all possible values of \( \delta_{y_1} \) and \( \delta_{y_2} \), we find the maximum token delay 3 on path \( \psi(3, 5) = (t_3 \rightarrow t_1 \rightarrow t_2 \rightarrow t_5) \) of \( \delta_{y_1} = 0 \) and \( \delta_{y_2} = 5 \). For 1-cyclic schedules, we also can derive token delays similarly, where \( \gamma_{0j}^{\psi(i,j)} \) is the longest path length defined for an 1-cyclic schedule graph \( G = (N^1, A^1) \).

**Corollary 1 (Token Delays for 1-Cyclic Schedules):** Consider a strongly connected and live TEG with critical circuit ratio \( \lambda \). An 1-cyclic schedule of the TEG, \( \{ x_t^i \mid t \in \mathcal{T}, r = 1, 2, \ldots \} \), has the following properties:

1. \( \{ \gamma_{0j}^{\psi(i,j)} \mid t \in \mathcal{T} \} \) defines a 1-cyclic earliest schedule with cycle time \( \lambda \) and feasible initial phase \( \delta \), \( \forall t \in \mathcal{T} \).

2. For a 1-cyclic earliest schedule with cycle time \( \lambda \) and initial phase \( \delta \), the token delay on any path \( \psi(i, j) \) is always the same as

\[
d^{\psi(i,j)}_\psi(\delta) = \tau_\psi(i,j) - h_{\psi(i,j)} + \gamma_{0j}^{\psi(i,j)} - \gamma_{0i}^{\psi(i,j)}\]

**IV. COMPARISON WITH FEEDBACK CONTROL**

Feedback control methods based on the max-plus algebra are effective for controlling token delays not to exceed a specified limit, on a place [12], [15], [27] or a path [25], [26]. However, the conditions required by a feedback controller may not always be met. Many real systems operate without feedback control or their controllers are not easily modified for a feedback controller. Our proposed method can compute the delay values, their average, the extremes, and the variance for a system or TEG without and even with a feedback controller. We present an example to evaluate token delay performance of a feedback controller.

**Example 4 (Feedback Control):** For the TEG in Fig. 1, we wish to limit the token delays at place \( p_{25} \) not to exceed 1. Fig. 4 shows the TEG controlled by a feedback controller designed by the method of [12], [15]. We observe that for the same initial phase \( \delta \), the feedback controller reduces the token delays at place \( p_{25} \) from \( d_{25}^1 (\delta) \) to \( d_{25}^2 (\delta) = 1 \) while it does not change the token delays on path \( \psi(1, 5) \). By maximizing the token delay formula (3) with regard to all possible initial phases \( \delta \)'s, we observe that the worst-case token delay at place \( p_{25} \) is also reduced from 3 to 1 by the feedback controller while the worst-case delay.
token delay on path \( \psi_{1}(3, 5) \) is not changed by the controller. We also would examine delay performance of a feedback controller on a path like \( \psi_{1}(3, 5) \) designed by the method of [25], [26].

V. Conclusion

For \( K \)-cyclic schedules of a TEG, we have characterized the delays and delay mechanism themselves on associated directed graphs. We have developed closed-formulae on token delays on a path, which is a function of the initial phase. We identified feasible initial phases for \( K \)-cyclic schedules. Thus, the delays can be computed from the longest path lengths between a pair of nodes in an associated directed graph. The formulae suggest how the delays are caused and how the delays should be controlled. The formulae can be used for predicting task delays and computing delay statistics, verifying delay constraints, maximizing or minimizing the delays, and improving the system design and control for better delay performance.

We yet need to examine how we can associate the initial phases with the initial lags of a TEG and how long it takes to reach a \( K \)-cyclic schedule. Further works also include efficient optimization algorithms for maximizing or minimizing token delays with regard to all possible initial phases. Mathematical analysis and systematic procedures for improving the system design or control so as to reduce or regulate token delays also should be examined. We may consider even a feedback controller design problem of determining proper feedback arcs and places between a set of transitions and intentional token sojourn times on the added places by using the token delay formulae. We can apply the results to task delay control in automated systems such as wafer delay control in cluster tools.

REFERENCES


Preprint submitted to IEEE Transactions on Automatic Control. Received: April 28, 2016 19:28:52 PST