Abstract - Cluster tools with a dual-arm wafer transport robot are common in semiconductor wafer manufacturing. However, existing modeling methods suffer either from computational complexity (e.g., Petri nets, detailed simulation) and the resulting dearth of insight or from simplicity and its attendant loss of accuracy and expression (e.g., Ax+B models). Looking toward application in tool configuration optimization, performance evaluation and fab-wide simulation models, we develop expressive and computationally tractable equations and recursions for the cycle time of such cluster tools. The models include transient periods and robot behavior to express nonlinearities in tool performance. The models incorporate the affinity of such tools toward lots with certain numbers of wafers. We conduct simulations of dual-arm cluster tools to assess the quality of our models and compare performance and computational complexity between various approximation methods.

Index Terms – Cluster tools, dual-arm robot, cycle time, fab-wide simulation, nonlinearity.

I. INTRODUCTION

With the proliferation of cluster tools in semiconductor wafer fabricators, an understanding of their behavior is essential. Expressive models for such tools can be used to understand the performance implications of changes to module process times, number of redundant modules, tool layout and other factors. If the models are computationally efficient, they can be used in fab-wide simulators to answer questions regarding fab capacity, cycle time, wafers per lot and wafer size, among others.

Existing modeling methods suffer either from computational complexity, inability to provide insight or inaccuracy. Petri net models and detailed simulation models can provide an accurate representation of the tools and have been studied for many years; see [1-3] for examples. However, these models often require an intractable duration of time for model development and performance computation. In addition, since such models are algorithmic in nature, they seldom provide insight into tool behavior. The simpler models in [4, 5] were developed to improve understanding and address computational complexity. They developed simple and intuitive models to evaluate the elapsed time from the loading to the unloading of a lot of a cluster tool; this duration is called the cycle time. These performance models treat cycle time as a linear function of lot size. However, in some cases, such as a parallel configuration or when the bottleneck consists of redundant process modules, the cycle time may be a nonlinear function of the number of wafers in a lot. In this paper, we address the nonlinearity issue. It is hoped that our models will find application with fab engineers seeking to understand key features of tool performance and as replacements for the ubiquitous Ax+B model used in full fab simulation. As our models are both very accurate (in the process bound region) and significantly less computationally expensive than detailed tool simulation (e.g. via Petri net model), they may well serve for the purposes outlined above.

One candidate model to address the nonlinearity is the flow line. As models of manufacturing systems, flow line models have been studied for many years; they may be called assembly lines or tandem queues (see [6-9]). Since they are appropriate for the modeling of certain tools in semiconductor wafer fabs, much work has been conducted to improve basic flow line models and analyze tool behaviors. In [10], efforts to incorporate realistic features such as wafer dependant process times, internal wafer buffers and setups dependent on wafer location were discussed. In [11], some insights and intuition about wafer advancement and transient behaviors were sought. However, flow line models do not include wafer transport robots. Although flow line models can give high fidelity predictions of tool behavior when the wafer handling robot does not limit tool performance, this need not be true when the robot dictates tool throughput.

In this paper, we develop expressive and computationally tractable models for dual-arm cluster tools. We extend the idea of the simpler models in [4, 5] to develop a model for dual-arm cluster tools. We also introduce flow line models that can explain nonlinearity. Finally, we propose a new approximation model to address nonlinearity with higher accuracy than flow line models. The nonlinearity accounts for the fact that such tools have an affinity for lots with a certain number of wafers (we call it wafer count affinity, it was termed wafer affinity in [12]).

The organization of this paper is as follows. Section II introduces the systems and some basic notation related to cluster tools. In section III, we develop three performance models to estimate lot cycle time. We develop linear models, called the Ax+B model, for dual-arm cluster tools by modifying the models from [4, 5]. The second class of models is based on flow lines and the results of [13]. We develop the third class of approximation models by extending the second class of models to incorporate robot behavior. In section IV,
two realistic and practical examples of the results are discussed. Concluding remarks are presented in Section V.

II. SYSTEM DESCRIPTION

Cluster tools are commonly used in the fabrication of semiconductor wafers and consist of input/output equipment, such as cassette ports and load locks, wafer transport robots and process modules. Cluster tools can be characterized based on the configuration of their process modules as possessing a serial configuration, parallel configuration or mixed configuration. Examples are depicted in Fig. 1 (a), (b), and (c), respectively. In semiconductor manufacturing, it is common for the process modules to constrain the throughput of cluster tools as opposed to the wafer transport robot. Therefore, we focus on developing performance approximations for process-bound cluster tools. For serial configuration tools, all the process modules conduct a different process and every wafer must receive processing sequentially from each module. (In Fig. 1 (a), there are four processes denoted A, B, C and D.) In the parallel configuration, all the modules provide an identical process and each wafer need visit only one of the modules. (In Fig. 1 (b), four modules each provide process A.) In mixed configuration tools, there is a series of processes that each wafer must receive and each such process may be provided by several modules dedicated to that process. (In Fig. 1 (c), there are three processes denoted as A, B and C. Two modules are denoted to process B.) Such mixed configuration tools may be interconnected to form multi-cluster tools. A classic example of a multi-cluster tool is the clustered photolithography scanner; see Fig. 1 (d). We use $M$, $\tau_i$ and $N_i$ to denote the number of processes, deterministic process time of process $i$ and number of modules devoted to process $i$, respectively ($i = 1, \ldots, M$). For example, Fig. 1 (c) has $M = 3$, $N_1 = 1$, $N_2 = 2$ and $N_3 = 1$. The process time of all modules providing the same process is identical.

We study tools with a dual arm wafer transport robot. Dual arm robots are popular in practice and enable the swap operation (see [14]). The swap operation occurs when process $i$ has a completed wafer and the robot holds one wafer that is next destined for process $i$. The swap operation consists of the removal of the completed wafer from process $i$ (the “pick”), a rotation for the swap and placing the unprocessed wafer into the module for process $i$. Robot move time from one module to another module takes a constant $\delta$ units of time and the pick or place operations require a constant $\varepsilon$ units of time. Rotation time for the swap operation takes $\theta$ units of time. Thus, each swap operation consumes $2\varepsilon + \theta$ units of time. We use $b$ to denote the bottleneck process. In this paper, the bottleneck process is defined as the process achieving the maximum value for the sum of process time and swap operation time divided by the number of redundant modules devoted to that process. That is,

$$b = \arg \max_i \frac{\tau_i + 2\varepsilon + \theta}{N_i}.$$  

As is common in the literature, we do not consider input equipment as it can typically supply wafers into the tool faster than the rate of tool processing. In [5], the required number of load locks for steady state operation of a single arm serial configuration cluster tool is studied.

Note that cluster tools with a dual-arm robot mostly use the swap operation, especially in steady state. In [14], it has been shown that an optimal steady state robot operation policy for a dual-arm robot is the swap operation under certain reasonable conditions. Here, we assume that the cluster tool robot employs a schedule derived from the optimal steady state robot operation policy. That is, wafers will be advanced by swap operation except the load of the first wafer into the modules and the unload of the last wafer from the modules.

III. PERFORMANCE MODELS

In this section, we will introduce several performance models for the elapsed time from the arrival of a lot to an empty tool at the completion of the lot; this duration is called the lot cycle time (CT). For tools with a single robot arm, simple and intuitive performance models were introduced in [4, 5]. However, their discussion is limited to tools with a single arm robot in either a serial configuration or a parallel configuration. They do not consider the mixed configuration. In addition, their model does not address tool affinity toward lots with a certain number of wafers. They treat performance as a linear function of the wafer per lot, although parallel configuration and mixed configuration tools may have nonlinearity and wafer count affinity.

In subsection A, we extend the model of [5] to tools with a mixed configuration and a dual arm robot. In subsections B and C, we develop models to address nonlinearity and wafer count affinity. Note that transient behaviors are considered and incorporated in our estimates of lot cycle time. As is done in prior work, we assume throughout that $W$ wafers arrive to the empty tool at time 0 and it is our objective to calculate the cycle time required until wafer $W$ exits the tool.

A. $Ax+B$ Models

The first performance models treat lot cycle time as a linear function of lot size; we call it an $Ax+B$ model. The key
concept of this model is the same as that of [5]. This model consists of a fixed cycle time and an incremental cycle time.

We first calculate the incremental cycle time, which means the average increase in cycle time when the lot size increases by one wafer. Since the bottleneck process limits tool throughput and consists of \( N_b \) redundant modules, the cycle time increases by \( \tau_b + \theta + 2\epsilon \) to process an additional \( N_b \) wafers. Therefore, on the average, one additional wafer increases the cycle time by \( \frac{(\tau_b + \theta + 2\epsilon)}{N_b} \). Use \( A \) to denote this incremental cycle time.

To calculate the fixed cycle time, consider the cycle time for single wafer lot. Then, the cycle time is the time to complete that single wafer. It requires \((M+1)\) robot moves between modules, \((M+1)\) pick operations, \((M+1)\) place operations, and requires service from all of the \( M \) processes. Therefore, \( CT = (M + 1)\delta + 2(M + 1)\epsilon + \sum_{i=1}^{M} \tau_i \). As the incremental cycle time \( \Delta \) is equal to \( \tau_b + \theta + 2\epsilon \), the fixed cycle time \( CT - \Delta = (M + 1)\delta + 2(M + 1)\epsilon + \sum_{i=1}^{M} \tau_i - A \). For a lot with more than one wafer, the first wafer is moved by the swap operation. Thus, we add \( M \) swap operations to the fixed cycle time. The final fixed cycle time is \( B = (M + 1)\delta + 2(M + 1)\epsilon + M\theta + \sum_{i=1}^{M} \tau_i - A \). Since the linear model only allows us to use a single value for \( B \), we have selected the value including swap time. If one is willing to allow \( B \) to take a slightly more general form, both cases can be readily included.

The following summarizes the previous discussion.

Ax+B Approximation: Consider a cluster tool with a single dual-arm robot that is initially empty. It may possess a serial, parallel or mixed configuration. A collection of \( W \) wafers arrive for processing and are served until the tool is empty. We can approximate the cycle time of the tools as

\[
CT \approx AW + B
\]

where

\[
A = \frac{\tau_b + \theta + 2\epsilon}{N_b},
\]

\[
B = (M + 1)\delta + 2(M + 1)\epsilon + M\theta + \sum_{i=1}^{M} \tau_i - A.\]

Once tool configuration, process time and robot action durations are determined, the incremental cycle time \( A \) and fixed cycle time \( B \) can be readily calculated for this model. With these fixed values, one easily obtains an approximation for the lot cycle time as a function of the wafers per lot.

However, this simple Ax+B model does not explain the nonlinearity of cycle time. When tools possess a parallel configuration or mixed configuration, wafers may not complete service at constant time intervals. This property has been discussed in [4]. While they were aware of this behavior, they neglected it to obtain simple models. We strive to address the nonlinearity since it may improve the accuracy of approximation and be a decisive factor for decisions on lot sizing or tool configuration. The following models address this behavior.

B. Deterministic Flow Line Models

Flow line models have been used as models of manufacturing systems and have been shown to well model clustered photolithography tools. In flow line model, a wafer enters the next module as soon as the module is ready; there is no consideration of wafer transport robots. For the purpose of model tractability, we next neglect wafer transport robots and abstract to flow line models. Such a flow line system is depicted in Fig. 2. Despite such abstraction, these models are fairly accurate to estimate performance and thus have been used to model cluster tools that are process bound. For example, the work of [10] and [11] discusses flow line models with the goal of incorporating diverse features of clustered photolithography scanners.

The system behavior of deterministic flow line models are described by the elementary evolution equations (EEEs). Let \( \tau_i \) denote the deterministic process time for process \( i \). \( N_i \) is the number of modules devoted to process \( i \). Let \( X_i(w) \) denote the entry time of wafer \( w \) into process \( i \). That is, \( C_i(w) = X_i(w) + \tau_i \). Then,

\[
X_i(w) = X_0(w-N_i),
\]

\[
X_i(w) = \max \{ X_i(w) + \tau_i, X_{i-1}(w-N_{i-1}) \},
\]

\[
X_i(w) = \max \{ X_{i-1}(w) + \tau_{i-1}, X_{i}(w-N_i) + \tau_i \},
\]

for \( w > 0, i = 1, ..., M-1 \) and initial conditions \( X_i(w) = -\infty \) for \( w \leq 0 \) and \( i = 1, ..., M \).

Here, we use the theoretical upper bound on lot cycle time from [13] instead of the EEEs as the starting point for our models. We incorporate robot behavior by using a modified process time \( \tau_i' \). As we assume that the swap operation is used except for the advancement of the first wafer, the swap operation time is added to the module process time. We define the modified process time as

\[
\tau_i' = \tau_i + 2\epsilon + \theta.
\]

Let \( E(w) \) denote the exit time of wafer \( w \) from the tool and use \( W \) as the number of wafers in the lot. We assume that a lot is ready at time 0. From the theorem of [13], we obtain a recursion for the upper bound on the completion time of each wafer in a deterministic flow line as follows

\[
E(w) \leq \max \left\{ \sum_{i=1}^{M} \tau_i', \max_{1 \leq k \leq M} \left\{ E(w-N_k) + \tau_i' \right\} \right\},
\]

with initial conditions \( E(w) = -\infty \) for \( w \leq 0 \).
We compute the completion time of all the wafers recursively. Lot cycle time is equal to completion time of the last wafer of the lot, that is \( CT = E(W) \).

This calculation can be simplified with the following notation. Let \( N \) be the subset of the integers \( Z \) for which \( N_i \) takes that value for some \( i \). That is, \( N = \{ xeZ : x = N_i, 1 \leq i \leq M \} \). Let \( \tau_{\max}(i) \) be the maximum process time for all processes with \( i \) redundant modules, that is,

\[
\tau_{\max}(i) = \max_{\{j/N_i=j\}} \tau_{j}.
\]

For example, consider a tool with \( N_1 = 1, N_2 = 2, N_3 = 1, \tau_1 = 10, \tau_2 = 15, \) and \( \tau_3 = 20 \). Then, \( N = \{1, 2\} \). \( \tau_{\max}(1) = 20 \) and \( \tau_{\max}(2) = 15 \). With this notation, we obtain a modified and equivalent, but slightly simpler, performance model as follows.

**Flow Line Approximation:** Consider a cluster tool with a single dual-arm robot that is initially empty. It may possess a serial, parallel or mixed configuration. A collection of \( W \) wafers arrive for processing and are served until the tool is empty. Based on a flow line model, the exit time of these wafers is approximated by the recursion:

\[
E(w) \approx \max \left\{ \sum_{i=1}^{M} \tau_i', \max_{i \in N} \{E(w-i) + \tau_{\max}(i)\} \right\},
\]

with initial conditions \( E(w) = -\infty \) for \( w \leq 0 \).

C. Approximation Models

The flow line models exhibit wafer count affinity. However, the flow line models have two restrictions. First, the process times are deterministic and we set \( \tau_i \) assuming the swap operation. However for the first wafer, there are robot move times not included in \( \tau_i \). Second, as flow line models neglect the wafer transport robot, they do not incorporate a key robot behavior. Consider a flow line model for a tool with a parallel configuration of four identical modules. Every modified process time, four wafers simultaneously exit the tool. This demonstrates that the tool has an affinity for lots consisting of a multiple of four wafers. Such a tool is most efficient when the lot size is an integral multiple of four. However, since only one robot exists, it is not possible to complete more than one wafer at the same time. We now develop an approximation model without these restrictions.

With an assumption of advancement by swap operation, the first wafer can exit at \( \sum_{i=1}^{M} \tau_i + (M \plus 1)(\delta + 2\epsilon) + M\theta \). Let \( C = (M \plus 1)(\delta + 2\epsilon) + M\theta \). Then, \( E(1) = \sum_{i=1}^{M} \tau_i + C \). Since \( E(w) \geq E(1) \) for \( w \geq 1 \), we obtain

\[
E(w) \geq \sum_{i=1}^{M} \tau_i + C.
\]

Since a tool has only one wafer transport robot, the fastest possible completion time of wafer \( w \) is attained when the robot returns to the last process immediately after unloading the previous wafer \( (w-1) \) from the tool. The robot then picks wafer \( w \) (which we assume is complete to obtain the fastest possible completion time), moves and then places it to the output port. Thus,

\[
E(w) \geq \sum_{i=1}^{M} \tau_i + C + \delta + 2\epsilon + \theta.
\]

When the process constrains the throughput of the tool, the bottleneck process time determines the throughput time. For example, consider a parallel configuration tool with \( 4 \) redundant modules. Wafer \( w \) can enter the tool when wafer \( (w-4) \) exits the tool. Thus, the earliest possible completion time of wafer \( w \) is the sum of the completion time of wafer \( (w-4) \), the process time and the robot handling time. The time spent for the swap operation is \( D = \theta + 2\epsilon \). Let \( \tau_{\max}(i) \) denote the maximum process time for all processes with \( i \) redundant modules, that is,

\[
\tau_{\max}(i) = \max_{\{j/N_i=j\}} \tau_{j}.
\]

We thus obtain the following approximation when the process of the tool limits the throughput:

\[
E(w) \approx \max_{i \in N} \{E(w-i) + \tau_{\max}(i)\} + D.
\]

Combining these concepts, we obtain the following approximation.

**New Approximation (APPX):** Consider a cluster tool with a single dual-arm robot that is initially empty. It may possess a serial, parallel or mixed configuration. A collection of \( W \) wafers arrive for processing and are served until the tool is empty. The exit time of these wafers is approximated by the recursion.

\[
E(w) \approx \max \left\{ \sum_{i=1}^{M} \tau_i + C, E(w-1) + 2(\delta + \epsilon), \max_{i \in N} \{E(w-i) + \tau_{\max}(i)\} + D \right\}
\]

where

\[
C = (M + 1)(\delta + 2\epsilon) + M\theta.
\]

\[
D = \theta + 2\epsilon.
\]

with initial condition \( E(w) = -\infty \) for \( w \leq 0 \).

IV. NUMERICAL EXPERIMENTS

A. Simulation Overview and Assumptions

We now evaluate and compare the performance of our models using simulation. We consider three simple cluster tools with a dual-arm robot in the serial configuration, parallel configuration and mixed configuration. All of the tools are in the process bound region. A lot is ready at time 0. The robot move time between modules is a constant \( \delta = 4.15 \) seconds. It takes 0.5 seconds for both pick/place time and rotation time; that is, \( \epsilon = 0.5 \) and \( \theta = 0.5 \). The swap operation consists of pick, rotation and place operations; it requires \( 2\epsilon + \theta = 1.5 \) seconds.

We first obtain an exact performance baseline to evaluate our performance models. In an effort to obtain the minimal (optimal) cycle time value, we simulate the first tool and the third tool by hand. Since the robot operation policy is determined by trial and error, there is the possibility of a smaller cycle time. Thus, we call the cycle time obtained via simulation an upper bound on the optimal cycle time (UB on...
Opt.). For the second tool, we can easily obtain the optimal robot operation policy due to the simple configuration.

With the baseline upper bound on the optimal cycle time obtained from simulation, we next verify our models. The results of numerical experiments are presented in Fig. 3 and Table 1.

One of the key strengths of approximation is reduced computational complexity. We employ the C# programming language on an Intel Core 2 Quad CPU at 2.40 GHz with 2.00 GB of RAM. For all the tools, we conduct simulation and measure the computation required. As the results of the first and third tools are similar to those of the second tools, we only discuss computation for the second tool.

### B. Example 1: Serial Configuration Tool

The first example is a serial configuration tool from [15], and was designed and developed by FSI International, Inc. The tool consists of 10 processes with \( \tau_1 = 37.50, \tau_2 = 70.13, \tau_3 = 37.24, \tau_4 = 39.50, \tau_5 = 100.13, \tau_6 = 42.24, \tau_7 = 13.00, \tau_8 = 100.13, \tau_9 = 42.24 \) and \( \tau_{10} = 65.00 \). The time unit is seconds. This tool has a serial configuration and thus \( N_i = 1 \) for \( i = 1, \ldots, 10 \).

For the serial configuration tool, both the \( Ax + B \) model (1) and APPX model (3) estimate lot cycle time very accurately. The error is 0.11% for both models. The error percent is calculated by averaging (over all \( W \) values considered) the percent differences between cycle times from our models and the simulated upper bound on optimal cycle time. As seen in Table 1, all three performance models provide generally good performance estimation. In Fig. 3 (a), the serial configuration yields linearity of lot cycle time with respect to the number of wafers.

### C. Example 2: Parallel Configuration Tool

The second tool has a parallel configuration with four redundant modules. This tool is simple but allows us to demonstrate the wafer count affinity property. The process time is 80 seconds (\( \tau_1 = 80.00 \)) and \( N_1 = 4 \). The second tool has a parallel configuration with four redundant modules. The results are depicted in Fig. 3 (b).

The graph shows the nonlinearity of lot cycle time with respect to the number of wafers. For example, the cycle time increment when the lot size increases from 17 to 18 is 10.80 seconds. When the lot size increases by one, lot cycle time also increases by 10.80 seconds until the lot size is 20. However, when the lot size becomes 21, the increment of lot cycle time is 49.10 seconds. In a sense, there is a loss of efficiency when the lot size is increased to 21 wafers. Likewise, we obtain the highest efficiency when the lot size is equal to an integral multiple of four wafers. This thus demonstrates the wafer count affinity.

This preference for lots with certain wafer counts is manifested in tool throughput. As depicted in the graphs, an \( Ax+B \) model (1) does not address this behavior. Although a flow line model (2) incorporates the property, a strict flow line can be inaccurate; here there is 10.10% error. APPX model (3) we propose has 0.60% error and expresses the wafer count

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Line</td>
<td>APPX</td>
</tr>
<tr>
<td>Tool 1 serial</td>
<td>3.11</td>
</tr>
<tr>
<td>Tool 2 parallel</td>
<td>10.10</td>
</tr>
<tr>
<td>Tool 3 mixed</td>
<td>2.24</td>
</tr>
</tbody>
</table>

Table 1. Accuracy of performance models.

<table>
<thead>
<tr>
<th>Tool 2</th>
<th>UB of Opt.</th>
<th>Flow Line</th>
<th>APPX</th>
<th>Ax+B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simul. Time (ms)</td>
<td>150009</td>
<td>20001</td>
<td>20001</td>
<td>10001</td>
</tr>
<tr>
<td>Scaled Time</td>
<td>15.00</td>
<td>2.00</td>
<td>2.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 2. Computation time for tool 2.
affinity property. The simulation time and computation time for the second tool are given in Table 2. Our models require 7.5 times less simulation time compared to detailed simulation. The computation time for the upper bound on the optimal cycle time is based only on simulation running time; the robot schedule has been calculated by hand in advance. To develop the simulator and determine the robot schedule takes considerably more time.

D. Example 3: Mixed Configuration Tool

The third tool has a mixed configuration with three processes. The process times are \( \tau_1 = 30.00 \), \( \tau_2 = 160.00 \) and \( \tau_3 = 20.00 \). The first and third processes are served by one module and two identical modules are devoted to process 2. Thus \( N_1 = 1 \), \( N_2 = 2 \) and \( N_3 = 1 \). Note that process 2, which is catered to by two redundant modules, is the bottleneck process of the tool.

For the mixed configuration tool, there is also wafer count affinity. Further, the number of bottleneck process modules plays a role in determining the preference for certain wafer counts. For the third tool, two process modules are devoted to the bottleneck process, thus it has highest efficiency when the lot size is an even number. The nonlinearity for the third tool is depicted in Fig. 3 (c).

E. Simulation Summary

Ax+B models (1) are very accurate only for the serial configuration tools. Although flow line models (2) address nonlinearity and wafer count affinity, it provides an inaccurate estimation for the parallel configuration tools. On the other hand, APPX models (3) provide the most accurate estimation among the three approximations for all the configurations. It estimates lot cycle time to within 0.60% of the true value. In addition, it reduces the computational complexity as well. The simulation time is reduced to 1/7.5 of the time required for detailed simulation.

V. CONCLUDING REMARKS

In this paper, we strive to develop expressive and computationally tractable models for lot cycle time in cluster tools that include nonlinearity and wafer count affinity. Since these properties are directly related with efficiency and operation cost at the fab level, it is important to address these properties in a manner which can be useful for fab simulation. In an effort to increase accuracy, we incorporate robot behaviors and transient periods. We develop three performance models to estimate lot cycle time: Ax+B models, flow line models and the APPX models. All of the models give accurate cycle time predictions for serial configuration tools. However, the Ax+B and flow line models are not as appropriate for parallel configuration tools. Ax+B models do not address the nonlinearity and wafer count affinity properties of parallel tools. While the flow line model incorporates those properties, nevertheless, its accuracy is less that desirable. The final APPX model addresses nonlinearity of lot cycle time without sacrificing the accuracy of estimation. From a computational complexity point of view, our models require less than 1/7.5 the simulation time compared to detailed simulation. Further, for certain classes of cluster tools, the APPX model has 10 times less error than the Ax+B model at only 2 times the computation. We expect that these models can be used for tool configuration optimization, performance evaluation and fab-wide simulation models.

Future work will include developing a model for throughput with cascading of lots, different lot types, setups and module failures.

REFERENCES