Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

METAMODEL-BASED QUANTILE ESTIMATION FOR HEDGING CONTROL OF MANUFACTURING SYSTEMS

Giulia Pedrielli

School of CIDSE
Arizona State University
699 S Mill Avenue
Tempe, 85258, Arizona, USA

Russell R. Barton

Department of Supply Chain and Information Systems
The Pennsylvania State University
413 Business Building
University Park, PA 16802, USA

ABSTRACT

Hedging-based control policies release a job into the system so that the probability of a job completing by its deadline is acceptable; job release decisions are based on quantile estimates of the job lead times. In multistage systems, these quantiles cannot be calculated analytically. In such cases, simulation can provide useful estimates, but computing a simulation-based quantile at the time of a job release decision is impractical. We explore a metamodeling approach based on efficient experiment design that can allow, after an offline learning phase, a metamodel estimate for the state-dependent lead time quantile. This allows for real time control if the metamodel is accurate, and computationally fast. In preliminary testing of a three-stage production system we find high accuracy for quadratic and cubic regression metamodels. These preliminary findings suggest that there is potential for metamodel-based hedging policies for real time control of manufacturing systems.

1 INTRODUCTION

Real-time control has become increasingly difficult as manufacturing systems and their models become more complex, in terms of job variety, machine flexibility and machine reliability. We consider a *control policy* to be a set of rules governing when jobs are released to the manufacturing floor (and perhaps to which machine in a flexible system). The objective of a control policy is to maximize efficiency (machine utilization) while meeting job demand times.

Control policies that are static in nature may lead to ineffective operation in terms of tardiness and high levels of work in process. Recently, dynamic release policies have been proposed. These policies choose the release of particular jobs based on the state of the manufacturing system at the time of the release decision (Angius et al. 2018). The work was motivated by (Gershwin 2000), where a family of *Control Point Policies* (CPP) was proposed for the first time. The basic idea underlying CPP, was to extend the *Hedging Point Policy* (HPP), originally proposed in Kimemia and Gershwin (1983). HPP sought production rates balancing inventory costs and costs for failure to meet demand, while hedging against possible machine downtime. CPP had a different focus, on lead time, and proposed the concept of *hedging time*, i.e., conservative measures of the lead time to complete an individual job.

Token based policies have been analytically investigated and implemented in industrial settings, the most famous being kanban, conwip, and basestock. The interested reader can refer to the review in (Liberopoulos 2013). All these policies have the same objective as HPP, but control the work in progress in different portions of the system instead of the release process, and they are applied in single as well as multi-stage systems. The parameters of these policies are typically optimized in a static manner, i.e., they are not a function of the specific system state. On the converse, real time scheduling policies such as Earliest

Due Date, or Least Slack, have been proposed to schedule and load parts onto the manufacturing system considering the due dates of the different batches.

CPP policies are most closely related to our work. For a CPP policy, lead time is characterized as a random variable, and the hedging time a quantile based on its probability distribution. A job within the set available for release is *ready* if the slack between the job demand time and the current system time is below the hedging time. In (Angius et al. 2018), the distribution of the lead time was estimated using an analytical method. The hedging time values were computed as quantiles of the lead time distribution, dependent on the specific state of the system (number of parts and state of the machines). The accuracy of the lead time estimation depends on the assumptions underlying the analytical model used to approximate the dynamical behavior of the manufacturing system. They examine systems that allow discrete-time Markov Chain (DTMC) representation and characterize the distribution of $Y_{jk}(S,W)$ by considering time to the absorbing state, for a particular starting condition S,W. But this approach required equal and fixed processing times, and fixed lead time constraints. We examine a more realistic modeling approach based on simulation and metamodeling. Our dispatching rule takes similar form: based on the machine state and work in process at each station, release a job of type k if the estimated lead time quantile for that particular S,W does not exceed the lead time constraint for that job. This allows the control policy to be used in more general and realistic settings.

We propose an analysis method, based on offline simulation of a fraction of all possible states, that fits a metamodel to predict the state-based lead time quantile. The rapid calculation enabled by the metamodel allows real-time support of a dynamic strategy for job release. Our approach builds on related work in quantile metamodeling, design of simulation experiments, and simulation in production planning. Section 2 introduces the needed notation, reviews the related work, and distinguishes our setting and approach from that work. Following that, we propose an analysis method and identify options for particular analyses that are part of the method. This is followed by a simple example, and includes comparison of the effectiveness of some of the options. Finally, we talk about the implications for dynamic job release policies.

2 RELATED WORK

In this section, we propose the formulation for our problem and point at the vast relevant literature for the topic of interest related to real time control of complex manufacturing systems. It is important to highlight that this paper focuses on lead time, defined as the random time required to a job to complete process when it joins a production system in a specific state. This metric is different from cycle time, that is state independent, and used to characterized the steady state time for a job to complete processing.

2.1 Problem Description and Notation

We consider a production system composed of machines and buffers. In this general setting, we characterize the state of machine i as S_i and the total work in process (in the buffer or in process) for jobs of type k at machine i as W_{ik}. Note that the machine state characterizes the processing time distribution. The machine might operate at a set of discrete rates, or might be down for a discrete number of failure types, adding a random repair time (whose distribution is failure type dependent) to the next jobs processing time. Suppose that τ_{ik} represents the (random) processing time for a job of type k at machine i, with distribution $F_{\tau_{ik}}(t|S_i)$. We define the lead time for the j-th job of type k, as Y_{jk}, as the sum of its waiting times in each buffer plus its processing time at each machine. Besides job type, the lead time will depend on the state of *all* machines, represented by the vector $\mathbf{S} = (S_i)$ and the work in progress state at every machine, represented by the array $\mathbf{W}$. We may write this variable as Y_{jk} rather than $Y_{jk}(S,W)$, but we assume that the lead time distribution depends on the system state (S,W). The lead time observed is not simply the sum of the processing times (across all machines) of jobs at the first station (including job j) when job j enters. In fact, the sum of processing times at machine 1 correctly characterizes the lead time for job j at machine 1, but at later machines, the number of jobs waiting to be processed when job j arrives may be greater

or less than the number experienced at machine 1, depending on the specific realizations of the (random) processing times at machine 1. Long processing times at machine 1, particularly for job j, will result in fewer jobs at machine 2 when job j arrives, and the opposite would occur when machine 1 processing time realizations are relatively short.

2.2 Related Work in Production Modeling and Scheduling

As mentioned above, the closest related control policy work is that of Angius et al. (2018). They examine systems that allow discrete-time Markov Chain (DTMC) representation and characterize the distribution of $Y_{jk}(S,W)$ by considering the time to the absorbing state, for a particular starting condition S,W. Unlike traditional production control methods that focus on target throughput while minimizing inventory, and treat system work in process as randomly varying, they develop policies that focus on the lead time distribution. Their policies control the job release based on a system state variable related to work in process state and machine state. But this approach required equal and fixed processing times, and fixed lead time constraints, which is avoided by computing quantiles via simulation.

Discrete-event simulation has been used for evaluating and developing production plans since the 1950s (Malcolm 1960). Ankenman et al. (2011) classify uses of simulation as the reference tool for the (i) evaluation of candidate production plans to select the best, (ii) to evaluate a production plan produced by mathematical programming either one-time or iteratively. The paper focuses on cycle time, steady state measure of performance for a manufacturing system, while this manuscript focuses on lead time, an state dependent measure. Discrete-event simulation has also been used for real-time production control (Son et al. 1999).

Simulation-based cycle time quantile estimation has been of interest for many years. Sivakumar and Chong (2001) simulate a backend process for semiconductor manufacturing, and extract from a 42-day run quantiles for cycle time. Simulations were repeated for different levels of material handling time, reduced machine failure rate, reduced setup time and other process changes. These were examined in a one factor at a time approach, without statistical assessment via confidence intervals or hypothesis tests. More recently, Xi Chen and coauthors explored metamodel-based estimation of cycle time quantiles as a function of system parameters (Chen and Kim 2013; Bekki et al. 2014; Batur et al. 2018). Other authors have found steady state distributions for the lead time for simpler Buzakott-type production systems using Markov models (Shi and Gershwin 2016; Angius et al. 2018).

2.3 Quantile Metamodeling

As we stated in Section 2.1, Y_{jk}, is the sum of buffer waiting times for the j-th job of type k plus its processing time at each machine. It will depend on the state of all machines, represented by the vector $\mathbf{S}$ and the work in progress state at every machine, represented by the array $\mathbf{W}$. We may write this variable as $Y_{jk}(S,W)$. For job release policies, we will need to know the value of a particular quantile for the lead time, say q_p, for $Y_{jk}(S,W)$ for every possible combination of values of S and W. We propose to use a quantile metamodeling approach for two reasons. First, we expect lead times for *nearby* states (S_1,W_1) and (S_2,W_2) to have similar quantile values, so that a low-order polynomial should approximate the change and allow reduced error for prediction, because all simulation replications across all states would be used in prediction, not just the set at the particular (S,W) setting. Second, we expect that it will be possible to fit a model of high fidelity based on running only a fraction of all possible (S,W) combinations. In the examples that follow, we consider three quantile metamodeling approaches: (1) ordinary regression with sample quantiles as the dependent variable, (2) ordinary regression using a variance-stabilizing transformation of the sample quantiles, (3) and quantile regression. In the computational example below, it is clear that the sample variance of the quantile varies with the expected value, and that a variance stabilizing transformation is appropriate. Quantile regression is popular in finance and other fields (Bassett and Koenker 2017) and has been employed in simulation metamodeling (Batur et al. 2018). Regression coefficients are estimated

by solving a mathematical programming problem:

$$\hat{\beta}(p) = \arg\min_{\beta} \left[\sum_{n} \rho_p \left(y_n - x_n' \beta \right) \right]$$

where, for our case, y_n is the observed lead time quantile for the n-th job (in the simulation experiment), x_n is the vector of regression terms, functions of the values of S and W for the n-th job, and

$$\rho_p(u) = (p - I(u < 0))\, u$$

Quantile regression, while more robust, does not try to assure that the regression predicted value at each (S,W) has a fraction p below it. It places the regression hyperplane in a way that, overall, leaves a fraction p, computed over all values of (S,W), below the hyperplane. The predictive error distribution has greater spread in the computational example below.

2.4 Metamodel Design of Experiments

Metamodels are fit to experimental data. The best choice of experimental runs depends on both the purpose for which the metamodel will be used, and the type of metamodel. These issues are discussed, for example, in (Barton 2015; Kleijnen 2007; Sanchez et al. 2018). There has been little written on the design of experiments for quantile regression, which historically has been applied in observational studies with randomly assigned experimental units (Bassett and Koenker 2017). Bekki et al. (2014) employed Latin hypercubes for two-factor quantile regression modeling, and Batur et al. (2018) employed full-factorial designs for studies involving two and three factors and second through fourth order polynomial regression models. Our objective is to limit the number of simulation conditions needed to fit the metamodel, and so we consider orthogonal arrays, which one can consider to be a fraction of a multilevel factorial design.

3 DYNAMIC STATE-BASED PRODUCTION CONTROL POLICY

In this section, we first provide the details of the family of state based loading policies of interest and subsequently provide the simulation driven method to estimate the quantities required for the implementation of the policy.

3.1 Overall Strategy

We seek to estimate the hedging time for a threshold policy that decides loading in a manufacturing line. We seek a job release policy that is dynamic, based on the state of the production system at any point in time. We take inspiration from Angius et al. (2018), where the idea is to:

- Discretize the time domain into time units sized based upon the, deterministic, process time of the sequence of identical machines in the system;
- At the beginning of each time unit, the policy allows for one of the following actions:
  - Load part of type k into the first machine and start processing;
  - Do nothing

The decision is made based on control point parameters, the hedging times Z_k, to be estimated for each part type. In particular, the policy acts only on ready parts. At each time unit t, a job j of type k is ready if:

$$D_j - t \leq Z_k(\mathbf{S}(t), \mathbf{W}(t)) \tag{1}$$

Where D_j is the due date for job j (input to the problem) and t is the time when the decision is being taken. In the traditional literature, the effort is related to the computation of the hedging times $Z_k(\mathbf{S}(t), \mathbf{W}(t))$.

In particular, these are related to the lead time of a part considering the current state $(\mathbf{S}(t), \mathbf{W}(t))$, that is the time interval necessary for the system to complete the processing of the parts that are currently loaded. The hedging time is defined as a quantile of the lead time distribution:

$$P(LT(\mathbf{S}(t), \mathbf{W}(t)) > Z_k(\mathbf{S}(t), \mathbf{W}(t))) \leq p_k \tag{2}$$

Where p_k is a probability of meeting lead time provided as input by the user. A clear problem is how to estimate the (state dependent) distribution of the lead time $LT(\cdot)$. In fact, once the distribution of the lead time is established, then the hedging time $Z_k(\cdot)$ is the associated quantile as shown in equation (2). For serial production lines and bernoulli machines, Angius et al. (2018) use an absorbing Markov model to estimate the distribution of the lead time and derive the quantile $Z_k(\cdot)$. However, such an approach will not work with different processing times, and even if robust to the distribution it will not be possible to extend it to: (i) manufacturing systems different from serial lines; (ii) cases where the loading decision is performed at multiple machines (the authors decide about the loading at the first machine of the line). While simulation driven approaches can be used to solve these problems, the current practice is to estimate the lead time distribution through its Empirical Cumulative Density Function (ECDF). While this gives the needed flexibility in terms of assumptions, the number of simulations required to have enough accuracy with an ECDF type of approach will result computationally expensive.

Figure 1, represents the job flow and the information flow within the manufacturing system. The machines and buffers share the information that constitutes the system state $(\mathbf{S}, \mathbf{W})$, which is used to produce an estimate for the quantile of the lead time ($\widehat{Z}$, refer to Section 3.2) in equation (2), the information on the due date (D in Figure 1) is then used as in equation (1) in order to rank the jobs and decide which job should be loaded (Colledani and Gershwin 2017). In the next section, we specify the method to derive the estimate of the lead time quantile at the center of the CPP strategy.

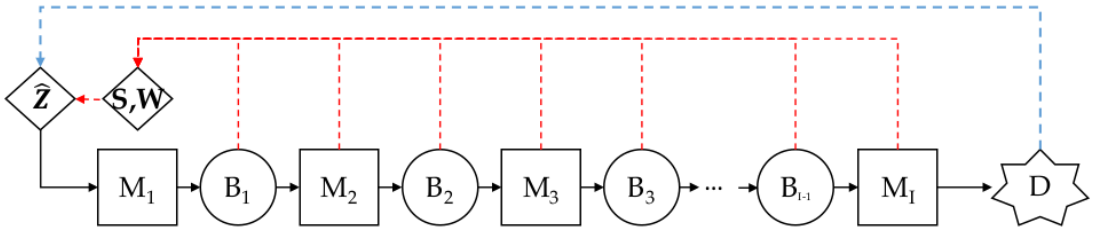


Figure 1: Job flow and information flow for a control point policy.

3.2 Specific Steps and Options

At the highest level, the metamodel-based approach we propose will: i) choose a metamodel type to predict a given quantile as a function of system state, ii) design a simulation experiment to generate quantile data (necessitating many replications for each job type and starting state), iii) fit and validate the fit of a metamodel, and finally, iv) use the metamodel to forecast quantiles for use in a real-time job release policy. In the current work, we want to explore the effectiveness of different types of metamodels, and follow the analysis process shown in Algorithm 1. In Steps 7-9, multiple models are estimated. In a real setting, the user might also wish to test alternative model fits during the offline fitting step, but the experiment design should be appropriate for the types of metamodels under consideration. Our options include the type or types of metamodels, and the experiment design to be used to fit the metamodel.

4 PRELIMINARY INVESTIGATION

In this section, we discuss a preliminary investigation of the proposed meta-model based approach for real time estimation of job lead time quantiles as a function of system state. First, we describe the system setup

Algorithm 1 Meta-model based lead time estimation for real time control of manufacturing systems

Input: Number of machines $i = 1,\dots,M$, number of part types $k = 1,\dots K$, p_k quantile probability for each part type; initial design, $\mathscr{D}$; n_0 number of replications for each experiment condition. Choose Model Type (MT): (i) Ordinary regression, (ii) Regression with variance stabilizing transformation, (iii) Quantile regression. Choose the order of the model (MO): (i) Linear, (ii) Quadratic, (iii) Cubic.

Output: $\hat{\beta}^*(\mathbf{S},\mathbf{W})$; estimation of model(s) coefficients as a function of the system state.

1:

2: **Run simulation experiments**: Run $N \equiv |\mathscr{D}|$ simulation conditions from the time 0 with the loading decision until the queue and the loaded job are cleared.

3: **Compute lead times**: Evaluate the lead time $LT_k(S_c,W_c), c = 1,\dots,N$, for each condition and replication. This will result in $N \times n_0$ lead times.

4: **Estimate the quantile**: For each condition $c = 1,\dots,N$ separate the n_0 observations into b independent batches of size $m = \frac{n_0}{b}$, resulting in the values $Z_{k,j}(c) = LT_{k\lceil p_k m\rceil}(c), k = 1,\dots,K; c = 1,\dots,N; j = 1,\dots b$. Derive the point estimate for the quantile $\widehat{Z}_k = \frac{1}{b} Z_{k,j}(c) k = 1,\dots,K; c = 1,\dots,N$. These will be the input to the model estimation step.

5: **Model estimation**: Use as independent variable the initial system state as defined by design $\mathscr{D}$. Set the dependent variables as responses estimated from the simulation $\widehat{Z}_k = \frac{1}{b} Z_{k,j}(c) k = 1,\dots,K; c = 1,\dots,N$.

6: **Variance stabilizing transformation**: for the VST model (described below), transform $\widehat{Z}_k \leftarrow \widehat{Z}_k^{0.3}$.

7: **For** All Model Types, All orders

8: Estimate the coefficients, fit and statistical significance for the regression models

9: **End**

in Section 4.1 and the models selected (Section 4.2). The obtained results are presented in Section 4.3, and subsequently discussed in Section 4.4.

4.1 System Description

We simplify the setting for this study, to consider a system with three machines, only two job types with different processing time distributions, no machine failures and only one machine speed (so the machine state space is a singleton). Processing times are exponentially distributed with rate 2 for job type 1 and rate 1 for job type 2. Of course a thorough testing would require examining other combinations of processing rates. We assume that no job release policy would consider a release with more than three jobs of either type at each station. That means, at the time of a decision, $W_{ik} \in \{0,1,2,3\}$ for $i = 1,2$ and $k = 1,2$. As a result, our work in process state space has $N = 256$ possible values. To implement a dynamic state-based control policy we also need to include the processing of the candidate job. Thus, we need 256 quantile estimates for each potential job type, $k = 1,2$, resulting into 512 quantile estimates in total.

4.2 Metamodel Type and Experiment Design

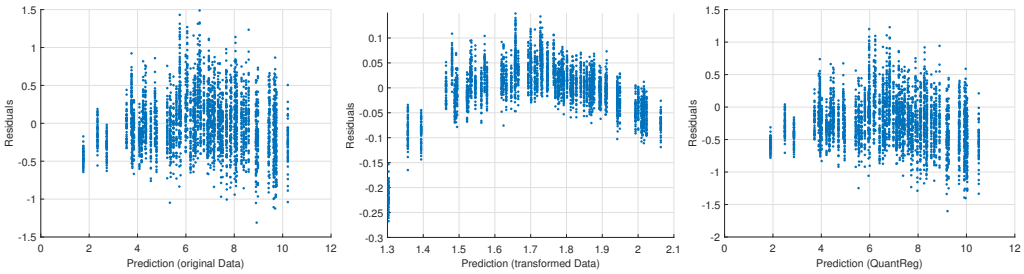
For this system, we expected large linear effects. If there were a single station with Gaussian processing times, the model would have a mean value and standard deviation linear in the jobs in system, and a linear function would be an appropriate quantile metamodel. But even in this simple case, if the processing time for jobs is non-normal then the distribution of the sum of prior-job processing times will change, tending to normal, and the nature of change in quantiles values will not generally be linear. So we also expected higher order terms to be important. Further, in a tandem system, the clearing time at the second station will depend on jobs at the first station interaction with jobs at the second station. Consider a job with long processing time at station 1 just entering the system. If there were one job with short processing time in process at station 1 and no jobs at station 2, then the station 1 job will clear station 2 before arrival of the new job, and will not contribute to station 2 clearing time for the new job. But if station 2 has a set of 3 long processing time jobs waiting/in process at station 2, then the station 1 job will be in the station 2 queue when the new job arrives at station 2, and will contribute to the new job lead time. This suggests that interaction terms will also be important. We did not expect homogeneous variance for the sample quantile values for each job type and each state condition, and so we explored a Box-Cox variance stabilizing transformation (VST) based on the observed relationship between the log mean and the log standard deviation of the quantiles across 512 conditions (see Chapter 3 in Montgomery (2017)). We observed a slope of .7, suggesting a transform of the quantile to the power 0.3.

Based on these considerations, we included linear regression with first-, second- and third-degree polynomials, using either the lead time quantile or the VST of the lead time quantile as the dependent variable. Because we are estimating a quantile, we also chose to include quantile regression as a metamodel type. Since our simulations are state-dependent, running one long simulation and collecting lead times for each job type as a function of work in process and state is problematic. First, the job type to release at each clearing of machine 1 is not clear. Second, the data would suffer from observational variation: it would not be the case that all combinations of work in process would occur with equal frequency. Furthermore, since we decided on an explicit experiment design running the simulation an equal number of times for each work in process combination, we did not explore the batching and sectioning approaches described in (Chen and Kim 2013). Because we were unsure of the proper degree, we selected a 68-run orthogonal array from the online collection of Neil Sloane (Sloane 2019). And Batur et al. (2018) suggest that such designs work well for quantile regression. We replicated this design for each job type. Because of our interest in determining quality of fit of a low-order polynomial, we designed the experiment to have relatively low uncertainty in the lead time quantiles. In particular, $5,000$ random sequences were generated for each condition, and the results were batched into groups of 100 observations for each condition, leading to 50 independent observations of the lead time quantile. One could instead use the quantile computed for the full set of $5,000$ sequences, but that would not allow us to explore variability in the quantile with the system state.

4.3 Results

Figures 2-4 show the residuals from the different modeling strategies as a function of the prediction over the 68 experimental conditions. We only report the results for the case where parts of type 1 need to be loaded in the system. Each vertical grouping of dots corresponds to lead time quantiles for one (or more) of the 68 state conditions used in the experiment. Also note that the scales of residuals are not directly comparable since the modeling approaches are different.

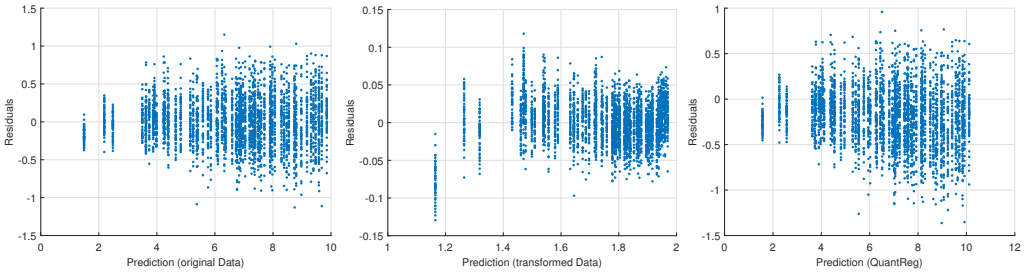
In Figure 2, we observe two phenomena in the residuals: in the first and third plots we observe heterogeneous variance, and perhaps some bias indicating the need for at least some quadratic terms. In the middle plot we see that the variance stabilizing transformation has been effective, but at a cost: a clear need for higher order terms. It is important to understand that the transformation affects not only the local variance, but it nonlinearly transforms the response surface as well. This constituted a strong motivation to increase the order of the polynomial. Quadratic terms were added to all three models, although the need was not as clear from the residual plots from the first and third regression metamodel types. This had no effect on heteroschedasticity for the first and third models, but much of the bias was removed from the transformed model. Still, bias at smaller quantile values was apparent, and so third-order polynomial terms were added.



(a) Original Data. (b) Transformed Data. (c) Quantile Regression.

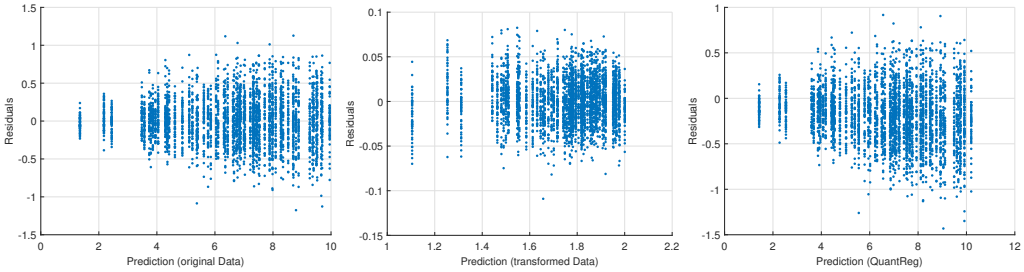
Figure 2: Residuals against predictions for the linear model case.

While the cubic model (as expected) does not correct the heterogeneous variance issue in original and quantile regressions, we can notice the improvement in the residuals resulting from the transformed data



(a) Original Data. (b) Transformed Data. (c) Quantile Regression.

Figure 3: Residuals against predictions for the quadratic model case.

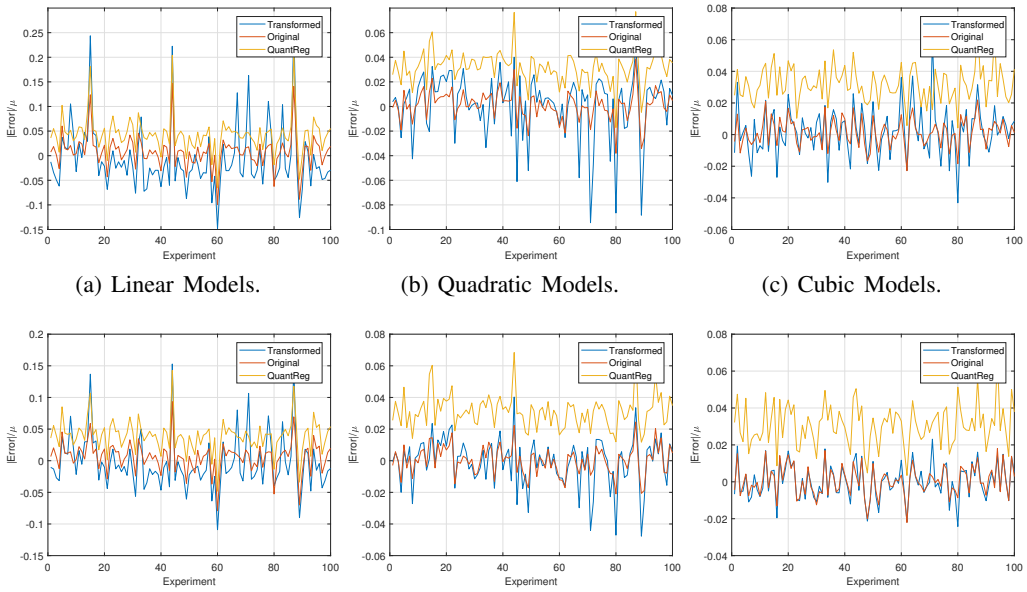


(a) Original Data. (b) Transformed Data. (c) Quantile Regression.

Figure 4: Residuals against predictions for the cubic model case.

(Figure 4b). While some higher order terms are significant as shown in the analysis of variance table in the Appendix, Tables 1-2, the higher order terms do not appear to improve the un-transformed or quantile regression model fits, since the residual spread remains the same. The third-order polynomial regression removes the large bias from the VST regression, however. A normal probability plot for the residuals for the model using transformed quantiles (not included here), gave a good fit, lending confidence to the appropriateness of this model for forecasts of quantile and quantile uncertainty. All but one of the terms for the third-degree VST regression were statistically significant. Probability plots for the residuals of the other models were not satisfactory, so there is less reliability in the P-values for those models.

Figure 5 report the relative error computed as $\frac{\hat{q}_k(x)-q_k(x)}{q_k(x)}$. Here, $\hat{q}_k(x)$ is the average quantile predicted using the model for condition x, while $q_k(x)$ is the quantile estimated running $5,000$ simulation replications of the test conditions. Concerning the testing, we randomly generated 100 conditions for the number of parts in the system when the target job is loaded. We show the results both for the case where the lead time quantile is computed for jobs of type 1 (Figure 5a-5c), and job of type 2 (Figure 5d-5f). While the relative error ranges between $(-0.15, 0.25)$ in the linear case, the range is reduced to $(-0.09, 0.07)$ across all strategies when a quadratic model is used (Figure 5b). The cubic models for regular and transformed regression see a further reduced range $(-0.04, 0.06)$ (Figure 5c). The same behavior is observed for jobs of type 2 (Figure 5d-5f). No such improvement is seen with quantile regression. We note that while the ordinary regression models work to minimize the sum of squared deviations at every state condition, the quantile regression works to find the best hyperplane (in model term space) separating that quantile overall, not individually for each state condition.



(a) Linear Models. (b) Quadratic Models. (c) Cubic Models.

(d) Linear Models. (e) Quadratic Models. (f) Cubic Models.

Figure 5: Relative error for part type 1 (Figure 5a-5c) and part type 2 (Figure 5d-5f).

4.4 Discussion

Simple quadratic and cubic regression based on original sample quantiles, with a variance-stabilized (VST), response produce accurate prediction for lead time quantile as a function of system state. Interestingly, we did not find superior performance for quantile regression. Also, in spite of clear variance heterogeneity, regression on VST of y, i.e., $Z_{kj}(c)$ does not exhibit superior performance over ordinary regression. This may be explained by the very strong linear components in the response, which are distorted in a non-polynomial way by the VST, with the cubic model the VST performance improves. Further, the VST regression produces residuals that appear both homoskedastic and Gaussian. We found no need to go to higher order polynomial response for improved prediction. Unlike prior work by Batur et al. (2018), we find no need for LASSO to remove terms from model, perhaps due to the direct connection of the estimator to the lead time. Also, we have effectively $5,000$ samples for each condition, whereas Batur et al. (2018) only use 250.

5 CONCLUSIONS & FUTURE WORK

Based on inspiration from the growing literature in Control Point Policies, we investigated the use of offline simulations to generate metamodels of state-based lead time quantiles for real-time control purposes. We found several interesting aspects: i) *all* models show a very low prediction error when tested out-of-sample; ii) quantile regression appears not to provide advantages in this context, iii) the cubic model with VST data appears to be the best compromise between model variance homogeneity and quality of the prediction. This low error of prediction demonstrates the effectiveness of this simple cubic regression metamodels for real-time prediction of lead times for this particular example, and the result encourage us to explore the performance over a broader class of systems. With these limited tests, we cannot confirm the general strength of our approach, but the results are certainly promising. Several directions merit further investigation: i) to explicitly model more complex set-ups where the state of the machine (i.e., idle, busy, failed, under repair) is considered; ii) to consider a varied set of processing time distributions, in particular that result in different bottleneck conditions; and iii) to model larger systems with more than two machines and two

job types. These generalizations will result in state spaces of greatly increased size, so metamodels that are robust to sparse data will be required.

APPENDIX

In the following, we report the results in terms of model coefficients and Analysis of Variance for the third-degree polynomial for each metamodel type. All the models are computed with a standardized and centered input. In fact, starting from $W_{ik} \in [0,3]$ we derive the design variables $x_h \in [-2,2]$, with the standardization $x_h = \frac{W_{ik}}{3} \cdot 4 - 2$. We have that the following holds: $x_1 = \frac{W_{11}}{3} \cdot 4 - 2, x_2 = \frac{W_{12}}{3} \cdot 4 - 2, x_3 = \frac{W_{21}}{3} \cdot 4 - 2, x_4 = \frac{W_{22}}{3} \cdot 4 - 2$, while b_0 represents the intercept.

Table 1: Cubic models ANOVA (job type 1).

(a) Original Data

Term	Coeff	SE	tStat	p
b_0	6.501	0.019	349.627	0.00
x_1	0.441	0.020	22.281	0.00
x_2	1.012	0.017	58.853	0.00
x_3	0.271	0.019	14.381	0.00
x_4	0.695	0.019	37.171	0.00
$x_1 \cdot x2$	-0.010	0.003	-3.019	0.00
$x_1 \cdot x3$	-0.013	0.003	-3.875	0.00
$x_1 \cdot x4$	-0.017	0.004	-4.777	0.00
$x_2 \cdot x3$	-0.028	0.003	-11.097	0.00
$x_2 \cdot x4$	-0.066	0.004	-17.695	0.00
$x_3 \cdot x4$	0.050	0.003	16.779	0.00
$x_1 \cdot x2 \cdot x3$	0.003	0.002	1.409	0.16
$x_1 \cdot x2 \cdot x4$	0.004	0.003	1.188	0.23
$x_1 \cdot x3 \cdot x4$	0.008	0.003	3.063	0.00
$x_2 \cdot x3 \cdot x_4$	0.006	0.002	2.939	0.00
$x_1^2 \cdot x2$	0.002	0.003	0.555	0.58
$x_1^2 \cdot x3$	0.005	0.003	1.797	0.07
$x_1^2 \cdot x4$	0.000	0.003	0.170	0.87
$x_1 \cdot x2^2$	-0.001	0.002	-0.572	0.57
$x_2^2 \cdot x3$	0.006	0.002	2.886	0.00
$x_2^2 \cdot x4$	0.008	0.002	3.286	0.00
$x_1 \cdot x3^2$	-0.001	0.002	-0.207	0.84
$x_2 \cdot x3^2$	0.003	0.002	1.228	0.22
$x_3^2 \cdot x4$	-0.010	0.002	-3.982	0.00
$x_1 \cdot x4^2$	0.001	0.003	0.471	0.64
$x_2 \cdot x4^2$	0.006	0.003	1.898	0.06
$x_3 \cdot x4^2$	-0.010	0.003	-3.256	0.00
x_1^2	0.001	0.004	0.337	0.74
x_2^2	-0.015	0.004	-4.159	0.00
x_3^2	0.015	0.003	4.474	0.00
x_4^2	0.053	0.004	13.507	0.00
x_1^3	0.002	0.004	0.555	0.58
x_2^3	0.003	0.004	0.770	0.44
x_3^3	0.003	0.004	0.783	0.43
x_4^3	-0.014	0.004	-3.438	0.00

(b) Transformed Data

Term	Coeff	SE	tStat	p
b_0	1.106	0.003	351.953	0.00
x_1	0.169	0.006	29.920	0.00
x_2	0.333	0.006	58.470	0.00
x_3	0.090	0.005	16.631	0.00
x_4	0.139	0.006	21.890	0.00
$x_1 \cdot x_2$	-0.037	0.002	-15.456	0.00
$x_1 \cdot x_3$	-0.028	0.003	-10.878	0.00
$x_1 \cdot x_4$	-0.035	0.002	-14.500	0.00
$x_2 \cdot x_3$	-0.035	0.002	-17.171	0.00
$x_2 \cdot x_4$	-0.067	0.003	-26.483	0.00
$x_3 \cdot x_4$	0.002	0.002	0.743	0.46
$x_1 \cdot x_2 \cdot x3$	0.003	0.000	7.212	0.00
$x_1 \cdot x_2 \cdot x4$	0.005	0.001	8.784	0.00
$x_1 \cdot x_3 \cdot x4$	0.004	0.000	8.746	0.00
$x_2 \cdot x_3 \cdot x4$	0.003	0.000	9.131	0.00
$x_1^2 \cdot x_2$	0.002	0.001	4.170	0.00
$x_1^2 \cdot x_3$	0.003	0.000	5.684	0.00
$x_1^2 \cdot x_4$	0.002	0.001	3.588	0.00
$x_1 \cdot x_2^2$	0.003	0.000	6.107	0.00
$x_2^2 \cdot x_3$	0.004	0.000	8.951	0.00
$x_2^2 \cdot x_4$	0.008	0.000	16.192	0.00
$x_1 \cdot x_3^2$	0.001	0.000	2.199	0.03
$x_2 \cdot x_3^2$	0.001	0.000	3.377	0.00
$x_3^2 \cdot x_4$	-0.002	0.000	-4.234	0.00
$x_1 \cdot x_4^2$	0.002	0.001	4.121	0.00
$x_2 \cdot x_4^2$	0.003	0.001	5.795	0.00
$x_3 \cdot x_4^2$	-0.001	0.001	-2.293	0.02
x_1^2	-0.023	0.004	-5.889	0.00
x_2^2	-0.052	0.003	-14.970	0.00
x_3^2	-0.011	0.003	-3.168	0.00
x_4^2	0.016	0.004	3.997	0.00
x_1^3	0.002	0.001	2.805	0.01
x_2^3	0.004	0.001	5.915	0.00
x_3^3	0.003	0.001	3.316	0.00
x_4^3	-0.004	0.001	-4.999	0.00

(c) Quantile regression

Term	Coeff	SE	tStat	p
b_0	6.712	0.025	265.796	0.00
$x1$	0.453	0.026	17.619	0.00
$x2$	1.068	0.024	44.584	0.00
$x3$	0.276	0.023	12.135	0.00
$x4$	0.694	0.026	26.345	0.00
$x1 \cdot x2$	-0.015	0.004	-3.627	0.00
$x1 \cdot x3$	-0.015	0.004	-3.444	0.00
$x1 \cdot x4$	-0.017	0.005	-3.611	0.00
$x2 \cdot x3$	-0.030	0.004	-8.555	0.00
$x2 \cdot x4$	-0.078	0.005	-15.212	0.00
$x3 \cdot x4$	0.051	0.004	11.943	0.00
$x1 \cdot x2 \cdot x3$	0.002	0.002	0.658	0.51
$x1 \cdot x2 \cdot x4$	-0.001	0.004	-0.357	0.72
$x1 \cdot x3 \cdot x4$	0.007	0.003	2.102	0.04
$x2 \cdot x3 \cdot x4$	0.006	0.003	2.217	0.03
$x1^2 \cdot x2$	0.002	0.004	0.546	0.58
$x1^2 \cdot x3$	0.007	0.003	2.523	0.01
$x1^2 \cdot x4$	0.005	0.004	1.230	0.22
$x1 \cdot x2^2$	0.000	0.003	0.045	0.96
$x2^2 \cdot x3$	0.007	0.003	2.364	0.02
$x2^2 \cdot x4$	0.007	0.003	2.105	0.04
$x1 \cdot x3^2$	0.006	0.003	1.902	0.06
$x2 \cdot x3^2$	0.003	0.003	0.979	0.33
$x3^2 \cdot x4$	-0.011	0.003	-3.361	0.00
$x1 \cdot x4^2$	0.004	0.004	0.958	0.34
$x2 \cdot x4^2$	-0.001	0.004	-0.255	0.80
$x3 \cdot x4^2$	-0.016	0.004	-3.913	0.00
$x1^2$	0.000	0.006	-0.034	0.97
$x2^2$	-0.019	0.005	-3.718	0.00
$x3^2$	0.011	0.004	2.439	0.01
$x4^2$	0.055	0.005	10.360	0.00
$x1^3$	-0.005	0.006	-0.805	0.42
$x2^3$	-0.003	0.005	-0.687	0.49
$x3^3$	0.004	0.005	0.928	0.35
$x4^3$	-0.010	0.005	-1.758	0.08

Table 2: Cubic models ANOVA (job type 2).

(a) Original data

Term	Coeff	SE	tStat	p
b_0	7.293	0.020	360.022	0.00
x_1	0.436	0.022	20.233	0.00
x_2	0.985	0.019	52.556	0.00
x_3	0.237	0.021	11.550	0.00
x_4	0.636	0.020	31.245	0.00
$x_1 \cdot x_2$	-0.007	0.003	-2.077	0.04
$x_1 \cdot x_3$	-0.011	0.004	-2.923	0.00
$x_1 \cdot x_4$	-0.013	0.004	-3.277	0.00
$x_2 \cdot x_3$	-0.021	0.003	-7.363	0.00
$x_2 \cdot x_4$	-0.051	0.004	-12.441	0.00
$x_3 \cdot x_4$	0.055	0.003	16.919	0.00
$x_1 \cdot x_2 \cdot x_3$	0.000	0.002	-0.179	0.86
$x_1 \cdot x_2 \cdot x_4$	0.005	0.003	1.619	0.11
$x_1 \cdot x_3 \cdot x_4$	0.003	0.003	1.083	0.28
$x_2 \cdot x_3 \cdot x_4$	0.001	0.002	0.331	0.74
$x_1^2 \cdot x_2$	0.002	0.003	0.639	0.52
$x_1^2 \cdot x_3$	0.001	0.003	0.475	0.63
$x_1^2 \cdot x_4$	-0.001	0.003	-0.180	0.86
$x_1 \cdot x_2^2$	-0.002	0.003	-0.678	0.50
$x_2^2 \cdot x_3$	0.004	0.002	1.651	0.10
$x_2^2 \cdot x_4$	0.005	0.003	1.719	0.09
$x_1 \cdot x_3^2$	-0.005	0.003	-1.797	0.07
$x_2 \cdot x_3^2$	0.002	0.002	0.666	0.51
$x_3^2 \cdot x_4$	-0.008	0.003	-3.078	0.00
$x_1 \cdot x_4^2$	0.001	0.003	0.200	0.84
$x_2 \cdot x_4^2$	0.005	0.003	1.619	0.11
$x_3 \cdot x_4^2$	-0.006	0.003	-1.949	0.05
x_1^2	0.005	0.005	1.199	0.23
x_2^2	-0.016	0.004	-4.241	0.00
x_3^2	0.013	0.004	3.586	0.00
x_4^2	0.050	0.004	11.700	0.00
x_1^3	0.004	0.005	0.931	0.35
x_2^3	0.006	0.004	1.504	0.13
x_3^3	0.005	0.004	1.211	0.23
x_4^3	-0.014	0.004	-3.037	0.00

(b) Transformed Data

Term	Coeff	SE	tStat	p
b_0	1.988	0.003	687.094	0.00
x_1	0.110	0.006	17.929	0.00
x_2	0.259	0.005	48.433	0.00
x_3	0.054	0.006	9.191	0.00
x_4	0.172	0.006	29.637	0.00
$x_1 \cdot x_2$	-0.040	0.002	-20.510	0.00
$x_1 \cdot x_3$	-0.017	0.002	-7.827	0.00
$x_1 \cdot x_4$	-0.028	0.002	-12.583	0.00
$x_2 \cdot x_3$	-0.033	0.002	-20.789	0.00
$x_2 \cdot x_4$	-0.079	0.002	-33.980	0.00
$x_3 \cdot x_4$	0.017	0.002	9.261	0.00
$x_1 \cdot x_2 \cdot x_3$	0.009	0.002	3.842	0.00
$x_1 \cdot x_2 \cdot x_4$	0.027	0.004	7.280	0.00
$x_1 \cdot x_3 \cdot x_4$	0.010	0.003	3.248	0.00
$x_2 \cdot x_3 \cdot x_4$	0.008	0.002	3.508	0.00
$x_1^2 \cdot x_2$	0.010	0.003	2.850	0.00
$x_1^2 \cdot x_3$	0.006	0.003	1.925	0.05
$x_1^2 \cdot x_4$	0.005	0.003	1.575	0.12
$x_1 \cdot x_2^2$	0.016	0.003	5.082	0.00
$x_2^2 \cdot x_3$	0.018	0.003	6.616	0.00
$x_2^2 \cdot x_4$	0.036	0.003	12.044	0.00
$x_1 \cdot x_3^2$	-0.002	0.003	-0.784	0.43
$x_2 \cdot x_3^2$	0.004	0.003	1.386	0.17
$x_3^2 \cdot x_4$	-0.011	0.003	-3.687	0.00
$x_1 \cdot x_4^2$	0.008	0.004	2.149	0.03
$x_2 \cdot x_4^2$	0.016	0.004	4.301	0.00
$x_3 \cdot x_4^2$	-0.007	0.004	-1.998	0.05
x_1^2	-0.006	0.003	-2.443	0.01
x_2^2	-0.055	0.002	-25.006	0.00
x_3^2	0.005	0.002	2.367	0.02
x_4^2	0.014	0.002	5.799	0.00
x_1^3	0.008	0.005	1.598	0.11
x_2^3	0.025	0.004	5.533	0.00
x_3^3	0.012	0.005	2.492	0.01
x_4^3	-0.019	0.005	-3.640	0.00

(c) Quantile regression

Term	Coeff	SE	tStat	p
b_0	7.487	0.022	335.613	0.00
x_1	0.433	0.024	18.052	0.00
x_2	1.005	0.020	49.651	0.00
x_3	0.196	0.022	8.918	0.00
x_4	0.642	0.022	28.799	0.00
$x_1 \cdot x_2$	-0.013	0.004	-3.109	0.00
$x_1 \cdot x_3$	-0.010	0.005	-2.161	0.03
$x_1 \cdot x_4$	-0.012	0.005	-2.447	0.01
$x_2 \cdot x_3$	-0.023	0.004	-6.189	0.00
$x_2 \cdot x_4$	-0.052	0.005	-9.956	0.00
$x_3 \cdot x_4$	0.062	0.004	16.728	0.00
$x_1 \cdot x_2 \cdot x_3$	0.000	0.003	-0.062	0.95
$x_1 \cdot x_2 \cdot x_4$	0.012	0.004	2.879	0.00
$x_1 \cdot x_3 \cdot x_4$	0.002	0.003	0.631	0.53
$x_2 \cdot x3 \cdot x_4$	0.002	0.003	0.872	0.38
$x_1^2 \cdot x_2$	0.001	0.004	0.235	0.81
$x_1^2 \cdot x_3$	0.005	0.003	1.506	0.13
$x_1^2 \cdot x_4$	-0.001	0.003	-0.284	0.78
$x_1 \cdot x_2^2$	-0.001	0.003	-0.304	0.76
$x_2^2 \cdot x_3$	0.003	0.003	1.117	0.26
$x_2^2 \cdot x_4$	0.002	0.003	0.892	0.37
$x_1 \cdot x_3^2$	-0.005	0.003	-1.389	0.16
$x_2 \cdot x_3^2$	0.002	0.003	0.645	0.52
$x_3^2 \cdot x_4$	-0.012	0.003	-4.126	0.00
$x_1 \cdot x_4^2$	0.002	0.004	0.513	0.61
$x_2 \cdot x_4^2$	0.005	0.003	1.332	0.18
$x_3 \cdot x_4^2$	-0.006	0.004	-1.443	0.15
x_1^2	0.014	0.005	2.628	0.01
x_2^2	-0.015	0.004	-3.477	0.00
x_3^2	0.014	0.004	3.200	0.00
x_4^2	0.054	0.005	10.492	0.00
x_1^3	0.005	0.005	1.004	0.32
x_2^3	0.006	0.004	1.424	0.15
x_3^3	0.016	0.005	3.423	0.00
x_4^3	-0.010	0.005	-1.903	0.06

REFERENCES

Angius, A., M. Colledani, and A. Horvath. 2018. "Lead-Time-Oriented Production Control Policies in Two-Machine Production Lines". *IISE Transactions* 50:178–190.

Ankenman, B. E., J. M. Bekki, J. Fowler, G. T. Mackulak, B. L. Nelson, and F. Yang. 2011. "Simulation in Production Planning: an Overview with Emphasis on Recent Developments in Cycle Time Estimation (Chapter 16)". In *Planning Production and Inventories in the Extended Enterprise*, edited by K. G. Kempf, P. Keskinocak, and R. Uzsoy, Volume 1. Boston: Springer.

Barton, R. R. 2015. "Tutorial: Simulation Metamodeling". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 1765–1779. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bassett, G. W., and R. Koenker. 2017. "A Quantile Regression Memoir". In *Handbook of Quantile Regression*, edited by R. Koenker, V. Chernozhukov, X. He, and L. Peng, Volume 1. Boca Raton: Chapman and Hall/CRC.

Batur, D., J. M. Bekki, and X. Chen. 2018. "Quantile Regression Metamodeling: Toward Improved Responsiveness in the High-Tech Electronics Manufacturing Industry". *European Journal of Operational Research* 264:212–224.

Bekki, J. M., X. Chen, and D. Batur. 2014. "Steady-state Quantile Parameter Estimation: an Empirical Comparison of Stochastic Krigin and Quantile Regression". In *Proceedings of the 2014 Winter Simulation Conference*, edited by S. Jain, R. Creasey, J. Himmelspach, K. P. White, and M. C. Fu, 3880–3891. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chen, X., and K.-K. Kim. 2013. "Building Metamodels for Quantile-Based Measures Using Sectioning". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 521–532. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Colledani, M., and S. B. Gershwin. 2017. "Dynamic Lead Time Based Control Point Policy for Multi-Stage Manufacturing Systems". In *Proceedings of the 11th Conference on Sotchastic Models of Manufacturing and Service Operations SMMSO 2017*, 19–26. Milan, Italy: Institute of Industrial Technology and Automation.

Gershwin, S. B. 2000. "Design and Operation of Manufacturing Systems: the Control-Point Policy". *IIE Transactions* 32(10):891–906.

Kimemia, J., and S. B. Gershwin. 1983. "An Algorithm for the Computer Control of a Flexible Manufacturing System". *AIIE Transactions* 15(4):353–362.

Kleijnen, J. P. C. 2007. *Design and Analysis of Simulation Experiments*. New York: Springer.

Liberopoulos, G. 2013. "Production Release Control: Paced, WIP-Based or Demand-Driven? Revisiting the Push/Pull and Make-to-Order/Make-to-Stock Distinctions". In *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, edited by J. M.-G. Smith and T. Baris, Volume 192. New York: Springer.

Malcolm, D. G. 1960. "Bibliography on the Use of Simulation in Management Analysis". *Operations Research* 8(2):169–177.

Montgomery, D. C. 2017. *Design and Analysis of Experiments*. Hoboken, New Jersey: John Wiley & Sons.

Sanchez, S. M., P. J. Sanchez, and H. Wan. 2018. "Work Smarter, not Harder: a Tutorial on Designing and Conducting Simulation Experiments". In *Proceedings of the Winter Simulation Conference*, edited by M. Rabe, A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 237–251. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Shi, C., and S. B. Gershwin. 2016. *Lead Time Distribution of Three-Machine Two-Buffer Lines with Unreliable Machines and Finite Buffers*. https://mit.atmire.com/bitstream/handle/1721.1/103963/OR%20398-16.pdf?sequence=1&isAllowed=y, 15th January 2019.

Sivakumar, A. I., and C. S. Chong. 2001. "A Simulation Based Analysis of Cycle Time Distribution, and Throughput in Semiconductor Backend Manufacturing". *Computers in Industry* 45(1):59–78.

Sloane, N. J. A. 2019. *A Library of Orthogonal Arrays*. http://neilsloane.com/oadir/index.html, 15th April 2019.

Son, Y. J., H. Rodríguez-Rivera, and R. A. Wysk. 1999. "A Multi-Pass Simulation-Based, Real-Time Scheduling and Shop Floor Control System". *Transactions of the Society for Computer Simulation* 16(4):159–172.

AUTHOR BIOGRAPHIES

GIULIA PEDRIELLI is an Assistant Professor for the School of Computing, Informatics and Decision Sciences Engineering at Arizona State University. Her research is in learning for simulation and simulation optimization. Her email address is: giulia.pedrielli@asu.edu.

RUSSELL R. BARTON is Distinguished Professor of Supply Chain and Information Systems and Professor of Industrial Engineering at the Pennsylvania State University. His research interests include applications of statistical and simulation methods to system design and to product design, manufacturing and delivery. His email address is rbarton@psu.edu.