Proceedings of the 2007 Industrial Engineering Research Conference
G. Bayraksan, W. Lin, Y. Son, and R. Wysk, eds.

Minimizing the Cost of Lean Production Control Transition

Sean M. Gahagan, Jeffrey W. Herrmann
Department of Mechanical Engineering
University of Maryland, College Park, MD 20742, USA

Abstract

Firms that implement lean principles commonly adopt pull production control techniques (especially kanbans). Changing a manufacturing system from push to pull production control, while ultimately beneficial, can be disruptive. This paper studies the transition process in a single-stage manufacturing system. It describes the three events that characterize this transition and the associated costs. Different combinations of cost mitigation techniques lead to three different scenarios, which require three different models. We compare these models and present tools to find an optimal transition policy. We derive some general lessons about the conditions that favor each mitigation technique.

Keywords
Lean, Production Control, Optimization

1 Introduction
Firms that implement lean principles commonly adopt pull production control techniques (especially kanbans). The fundamental element of lean production control transition is the conversion of a single processing stage from push to pull production control. During this transition, the stage experiences a surge of orders as it attempts to build a kanban inventory while it is also processing regular customer orders. The surge may overwhelm the capacity of the station, resulting in a backlog that would adversely affect customer lead times. To prevent this, we propose two temporary mitigating techniques – adding more resources or deferring some of the customer orders. The purpose of this paper is two-fold; to determine the best way to model this transition process and, using this model, to explore optimal mitigation policies to reduce the cost of lean transition.

1.1 Background
The lean literature unanimously advocates the transition of push production control to pull where possible [2, 4-7], but very little is written about the mechanics of the transition process or the behavior of systems in lean transition. Hopp and Spearman [2] discuss the mechanics of push and pull production control, their role in lean transition and even sketch out a lean transition scheme. However, they limit their analysis to the "before" and "after" steady state conditions. In fact, to our knowledge, there is no discussion of the transient effects of lean production control transition in the literature. Queueing literature though does discuss the effects of non-stationary arrival rates on system performance. Hall [1] discusses ways to model systems with non-stationary arrival rates. He explains that the size of changes in arrival rate relative to capacity dictate which modeling technique to use. For systems in which the arrival rate is always much lower than capacity, steady state approximations can be used. In systems where the arrival rate is much larger than capacity, a fluid flow approximation is more appropriate. For systems where the arrival rate is close to capacity, he suggests that only simulation can accurately model system performance.

1.2 Problem Setup
In this work we study a single stage of a production system as it transitions from push production control to pull. We characterize this transition in terms of three events. The first event is the arrival of the first kanban card, which occurs at time $t = t_0$. We assume that the number of kanban cards, n_k, is predetermined. Further, we assume that the arrival rate of the cards, λ_k, is constant. The last kanban card arrives at $t = t_1 = t_0 + (n_k - 1) / \lambda_k$. We assume $t_1 > t_0$. The stage continues to receive customer orders at a rate of λ_a. The total arrival rate for $t_0 < t < t_1$ is $\lambda_k + \lambda_a$. The stage has r resources to process the cards and orders. Each resource processes orders at a rate of λ_r. The total processing rate of the stage is $r\lambda_r$. If $\lambda_k + \lambda_a > r\lambda_r$ then a backlog of orders is accumulated during the arrival of the cards. The

final transition event is the completion of processing of the last kanban card at $t = t_2$. If $\lambda_k + \lambda_a << r\lambda_r$ then t_2 could be as little as $t_1 + 1/\lambda_r$. However, if $\lambda_k + \lambda_a > r\lambda_r$ and a backlog is created, t_2 could be as much as $n_b / (r\lambda_r - \lambda_a)$. When the last kanban card has been processed, the stage can be converted to pull production control because customer orders can now be satisfied from the now-full downstream buffer.

We are interested in how the system behaves during this transient phase and how temporary changes to the system affect the process. Two types of temporary changes are considered in this paper. We say that r_+ resources may be added to the system, increasing the total processing rate of the system to $(r + r_+)\lambda_r$ during the transition. We also consider deferring orders at a rate of λ_d such that the total arrival rate during the transition is $\lambda_k + \lambda_a - \lambda_d$.

1.3 Control Variables

We can explicitly identify the decision variables for managing the transition process. Since the number of kanban cards to be introduced is known a priori, only the rate of their introduction λ_k is an input, where $0 < \lambda_k < \infty$. As an alternative, one could specify the length of the transition, t_1-t_0, where $\lambda_k = n_k / (t_1 - t_0)$. The mitigation factors are also inputs. The number of additional resources r_+ is an input, where r_+ is an integer on $0 \le r_+ < \infty$. The deferral rate of customer orders λ_d is also an input, where $0 \le \lambda_d \le \lambda_a$.

1.4 Transition Cases

From this, we can identify three distinct conditions under which transition takes place based on the relationship between arrival rate and processing rate. This relationship directly affects how much of the transition time occurs before and after t_1.

- **CASE 1: Arrival Rate is Lower than Processing Rate: $\lambda_k + \lambda_a - \lambda_d << (r + r_+)\lambda_r$**
  In this case, the surge of orders does not completely consume the available capacity and no backlog is created during transition. The kanban cards are processed as they arrive and the transition is nearly complete at t_1, minimizing t_2.

- **CASE 2: Arrival Rate is Higher than Processing Rate: $\lambda_k + \lambda_a - \lambda_d >> (r + r_+)\lambda_r$**
  In this case, the surge of orders completely consumes the available capacity and a backlog of cards and orders is created during transition. Here t_1 may be minimized as more, possibly all, of the kanban cards are processed after the final arrival.

- **CASE 3: Arrival Rate is Equal to Processing Rate: $\lambda_k + \lambda_a - \lambda_d \approx (r + r_+)\lambda_r$**
  In this case, the surge of orders nearly equals the available capacity. A backlog of orders and cards may be created. When arrivals and capacity are nearly in balance, the formation of a backlog becomes more dependent on variation in processing times. For this condition, t_1 may be at any point between t_0 and t_2.

An interesting feature of this problem is the fact that there are decision variables in this identification scheme, meaning that the nature of the problem itself is a function of the inputs.

1.5 Transition Objective

Holding orders in inventory or backlog, adding resources and deferring orders all have a cost. Our goal is to find the optimal input values to minimize total cost. We define total cost C_{tot} as

$$C_{tot} = C_d + C_r + C_i + C_b \qquad (1)$$

where C_d is the cost of orders deferred, C_r is the cost of additional resources, C_i is the cost of holding inventory (kanban orders in downstream queue) and C_b is the cost of holding backlog (orders in queue during transition). We can further define these cost components in terms of the system variables we have already defined.

$$C_d = \lambda_d (t_2 - t_0)\, c_d \qquad (2)$$

$$C_r = r_+ (t_2 - t_0)\, c_r \qquad (3)$$

$$C_i = (n_k/2) (t_2 - t_0)\, c_i \qquad (4)$$

$$C_b = c_b \left(\int_{t_0}^{t_2} \mathbf{Q}(t) dt \right) \qquad (5)$$

where c_d is the cost per unit of orders deferred, c_r is the cost rate per unit of additional resources, c_i is the cost rate per order held in inventory, c_b is the cost rate per order held in backlog and $Q(t)$ is the number of orders held in backlog at time = t. We note that, as the deferral rate λ_d increases, the deferral cost increases, but the other costs decrease due to the smaller backlog and shorter transition time. Similarly, as r_+ increases, the resource cost increases, but the other costs decrease due to the smaller backlog and shorter transition time. Increasing the kanban introduction rate λ_k should reduce the transition time unless it is too large, in which case excessive server increases the backlog.

2 Modeling Approach

We consider three different techniques to model this process: steady state approximation, deterministic fluid flow approximation and discrete event simulation. Hall [1] proposed that these three modeling techniques are the best candidates for analyzing systems with non-stationary arrival rates. Hall's categorization of these systems corresponds with our case definitions discussed above. The following sections describe the models.

2.1 CASE 1 - Steady State Model, Arrivals << Capacity

First, we consider the case where the surge of orders and kanban cards is much smaller than the capacity of the resource. That is, where $\lambda_k + \lambda_a - \lambda_d < (r + r_+)\lambda_r$. In this case we use a stochastic steady state (SSS) approximation. For $t < t_0$, we assume the system is in a steady state. For $t_0 < t < t_2$ we assume that the system switches to a second steady state. For $t > t_2$, the system reverts to a third steady state similar to the first. Since the surge never exceeds capacity, there is no backlog to deal with at the end of the transition and $t_1 = t_2$. We choose to approximate the system as a *G/G/m* server. Hopp and Spearman [2] provide an approximation for the cycle time, $CT_q(G/G/m)$. Using this, and assuming the coefficients of variation for the interarrival and processing time are both 1, we find t_2:

$$t_2 = t_1 + CT_q(G/G/m) + 1/(\lambda_r) = t_1 + \left(\frac{\left(\frac{\lambda_k + \lambda_a - \lambda_d}{\lambda_r (r + r_+)} \right)^{\sqrt{2(r+r_+ +1)}-1}}{(r + r_+)\left(1 - \frac{\lambda_k + \lambda_a - \lambda_d}{\lambda_r (r + r_+)} \right)} \right) \left(\frac{1}{\lambda_r} \right) + \left(\frac{1}{\lambda_r} \right) \quad (6)$$

We find the backlog by substituting the cycle time approximation into Little's Law:

$$\int_{t_0}^{t_2} \mathbf{Q}(t)dt = CT_q(G/G/m)(\lambda_a + \lambda_k - \lambda_d) = \left(\frac{\left(\frac{\lambda_k + \lambda_a - \lambda_d}{\lambda_r (r + r_+)} \right)^{\sqrt{2(r+r_+ +1)}-1}}{(r + r_+)\left(1 - \frac{\lambda_k + \lambda_a - \lambda_d}{\lambda_r (r + r_+)} \right)} \right) \left(\frac{1}{\lambda_r} \right) (\lambda_a + \lambda_k - \lambda_d) \quad (7)$$

Using these equations, it was possible to create a very simple, very fast spreadsheet model to evaluate the cost of Case 1 transitions.

2.2 CASE 2 - Fluid Flow Model, Arrivals >> Capacity

Next, we consider the case where the surge of orders and kanban cards is much greater than the capacity of the resource. That is, where $\lambda_k + \lambda_a - \lambda_d > (r + r_+)\lambda_r$. We choose to model this transition with a deterministic model called a fluid approximation model. A deterministic fluid flow (DFF) approximation model is one in which the flow of arrivals and departures are modeled as a continuous variables - flow rates. Fluid approximation models are easily illustrated. Figure 1 shows a fluid approximation model of our system in transition.

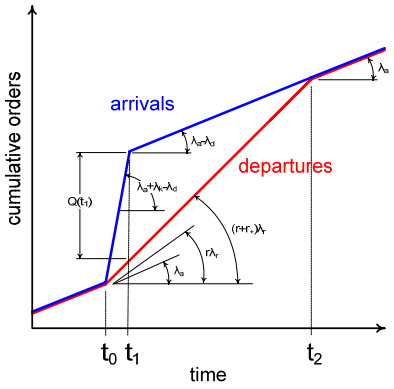

Figure 1: Deterministic fluid flow (DFF) approximation model

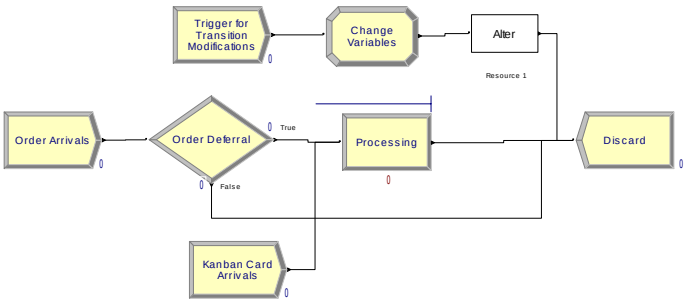

Figure 2: Simulation Model in Arena

In the figure, the blue line represents the cumulative flow of order and card arrivals during transition. The red line shows the flow of completed orders from the system. The slopes of these lines are equivalent to the flow rates of arrivals and departures from the system. Evaluation of this model is straightforward geometry. We address the transition in two phases; $t_0 < t \le t_1$ and $t_1 < t \le t_2$. In the former, the backlog is building, while in the latter it is being consumed. We first solve to find the backlog, $Q(t_1)$.

$$Q(t_1) = \left(\frac{(n_k - 1)}{\lambda_k} \right) \left(\lambda_a + \lambda_k - \lambda_d - \lambda_r (r + r_+) \right) \tag{8}$$

We can then use this result to find t_2:

$$t_2 = t_1 + \frac{\left(\frac{(n_k - 1)}{\lambda_k} \right) \left(\lambda_a + \lambda_k - \lambda_d - \lambda_r (r + r_+) \right)}{\lambda_a - \lambda_d - \lambda_r (r + r_+)} \tag{9}$$

Using these equations, we built a second spreadsheet model to calculate the cost of Case 2 transitions.

2.3 CASE 3 - Simulation Model, Arrivals ≈ Capacity

Finally, we address the case where the arrival rate is approximately equal to capacity, or where $\lambda_k + \lambda_a - \lambda_d \approx (r + r_+)\lambda_r$. In this condition, Hall [1] recommends the use of simulation to model the system. Simulation is a very powerful, but computationally expensive modeling technique. We built a simulation model of our system using Arena [3]. Figure 2 shows our simple system as modeled.

The simulation model itself is fairly straightforward, but collecting good performance data from a potentially short, transient period requires careful setup of the replication parameters. Our model uses a warm-up period of 10,000 times the order processing time, which it repeats for each replication. It maintains a count of how many kanban cards are in the system and it stops the replication when the last card exits. We use 100 replications. To evaluate performance we used the default reports which provide statistics on number of orders in backlog and replication length.

3 Comparison

In order to compare the models, we used them to estimate the performance of a system over a range of transition arrival rates centered about the capacity of the system. For this comparison, the system was configured as follows:

Table 1: Comparison Setup Input Variables

Input	Value
Mean order interarrival time, initial	0.013 (75 orders per unit time)
Mean processing time per order	0.1 (10 orders per unit time per resource)
Number of resources, initial	10
Number of kanban cards to be introduced	240
Number of resources, during transition	20

With these inputs fixed, we varied the kanban introduction rate λ_k to manipulate the ratio of transition load to capacity, $\lambda_k + \lambda_a - \lambda_d / (r + r_+)\lambda_r$, from 0.08 to 1.2. We looked at the average number of orders in backlog indicated by each model. Figure 3 shows a plot of each model output.

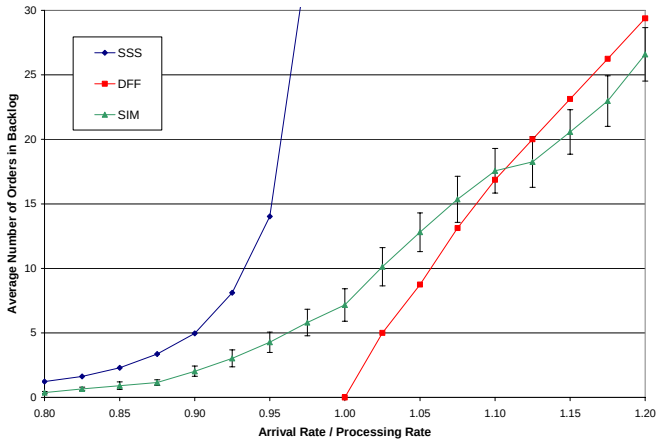

Figure 3: Average Number of Orders in Backlog versus Arrival Rate / Capacity

We expected the simulation model to most closely approximate the behavior of the system, which we predicted would be a smooth, monotonically increasing curve as the arrival rate slowly overcame the processing capacity of the system. As expected, the SSS approximation followed the simulation results initially, but increased asymptotically to infinity as the arrivals approached capacity from the left. The DFF reported backlogs well below the simulation model but caught up to and followed closely with it at higher arrival rates. If we assume that the simulation model is the closest approximation to the behavior of the system then the comparison is just as Hall predicted with the SSS approximation most useful when the arrival rate is well below capacity (Case 1), the DFF approximation most useful at rates well above capacity (Case 2) and the simulation model best at rates near capacity (Case 3). One marked difference between the models though is not reflected in the output - the processing time. The results of the SSS and DFF approximations were available from their respective spreadsheet models in a split second. The simulation model took nearly 2 minutes to process each data point. The flexibility of the simulation model has a high computational price that affects its usefulness for optimization of the transition.

4 Optimization

To ease the computational burden of an exclusively simulation-based optimization of the transition parameters, we propose a multi-model optimization scheme that utilizes all three models. Since the SSS and DFF models require so much less computing power than the simulation model, we can use them to reduce the size of the domain space before using the simulation model. For a given scenario, we suggest finding the minimum cost transition parameters using the SSS and DFF models, constraining the arrival rate to capacity ratio to, say <0.8 and >1.2, respectively. Spreadsheet applications like Excel have powerful solver utilities that can be used to find the optimal parameters very quickly. Then we can use the simulation model, which also comes packaged with an optimization engine, to explore the remaining domain space for superior solutions. The optimal solution can then be identified as the lowest result from the three separate optimizations. It should be noted that these arrival rate to capacity ratio thresholds are

strictly guesses, based largely on the results of the comparison case study above. Proper threshold selection for such a scheme is a subject for future study.

5 General Lessons

The objective of our modeling is to understand transition cost. Using the comparison case study above, we collected cost data for each trial with the cost rates all set to 100. Figure 4, demonstrates the effect of each cost component on overall transition cost with respect to arrival rate / capacity. It also shows how transition time is affected.

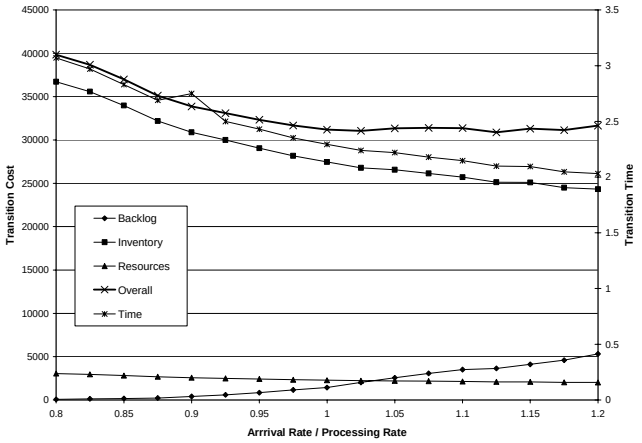

Figure 4: Transition Cost and Time versus Arrival Rate / Capacity

From this, we can see that in an optimum transition, the arrival rate is exactly balanced with the processing rate of the system. If the rate is too low, the processing resources are underutilized. Too high, and the backlog cost starts to catch up with inventory cost. We see here that the effect of backlog and resource costs are small compared to the contribution of inventory, but real-world cost rates could be quite different and drastically change these relationships. In general, balancing capacity with arrival rate is the answer to a low cost lean transition.

6 Conclusion

Conversion from push production control to pull is an important, but poorly understood part of lean manufacturing. In order to understand the cost of transition, we developed a cost model for transition and described three distinct types of transition. We developed three models, a stochastic steady state approximation, a deterministic fluid flow approximation and a simulation model, that all approximate the behavior of a single stage undergoing lean production control transition. We illustrated the differences in the models by applying them to a test case and we proposed a multi-model optimization approach that leverages the strengths of each model to quickly find the optimal transition parameters. We used our case study to demonstrate that an optimal transition balances the arrival rate with capacity. In the future we will expand our models to include multiple stages undergoing lean transition and endeavor to better understand how to optimize the many more decision variables such a model would present.

References

1. Hall, R., 1991, Queueing Methods for Services and Manufacturing, Prentice Hall, New Jersey.
2. Hopp, W. J., and M. L. Spearman, 1996, Factory Physics, Irwin/McGraw-Hill, Boston, Massachusetts.
3. Kelton, W., R. Sadowski, and D. Sturrock, 2004, Simulation with Arena, 3rd edition, McGraw-Hill, Boston, Massachusetts.
4. Liker, J., 2004, The Toyota Way, McGraw-Hill, New York.
5. Shingo, S., 1989, A Study of the Toyota Production System from an Industrial Engineering Viewpoint, Productivity Press, Cambridge, Massachusetts.
6. Slack, N. (Ed.). 1997. The Blackwell Encyclopedia Dictionary of Operations Management, Blackwell Publishers Ltd., Oxford, United Kingdom.
7. Womack, J., Jones, D. and Roos, D., The Machine That Changed the World: The Story of Lean Production, 1991, Harper Perenial, New York.