Stability of a 'Signal Kanban' Manufacturing System

Thomas I. Seidman

Department of Mathematics and Statistics

University of Maryland Baltimore County

Baltimore, MD 21250,

e-mail: ⟨seidman@math.umbc.edu⟩

Lawrence E. Holloway

Center for Robotics and Manufacturing Systems

University of Kentucky

Lexington, Kentucky 40506-0108

e-mail: ⟨holloway@engr.uky.edu⟩

February 4, 1997

Abstract

Signal kanban production control systems are often used in manufacturing systems with significant setup times where more conventional kanban systems are inappropriate. This paper analyzes signal kanban systems under demands with bounded fluctuation. It is shown that such systems are stable *in the sense that cumulative demand at any time trails cumulative demand by no more than a constant. We determine a maximum time for which a queue of backorders can be nonempty, and present results for setting the signal kanban's reorder* *point such that the backorder queue is guaranteed to remain empty under the bounded demand fluctuation. The results are then extended to consider the case where product orders are serviced in a fixed sequence, and we determine a minimum buffer fill level to ensure no backorders will occur.*

Submitted to 1997 ACC

1 Introduction

'Kanban' is an important production control technique used in "just-in-time" manufacturing or "lean manufacturing" [1]. In principle, a kanban is an authorization for a work station to produce a product based on the consumption of the product by a subsequent station. A kanban system is a classic example of a *pull* system in production control, where orders (authorizations) for production are based on consumption at downstream stations (hence, with demand arriving at the end of the line), in contrast to *push* systems, where orders arrive at the start of the line [4].

There has been recent interest in system-theoretic analysis of production control policies. Perkins and Kumar [5] considered manufacturing systems with multiple machines and multiple part flows with distributed control policies and introduced the notion of *stability* for such systems: a system is said to be stable if its cumulative production can trail cumulative demand by no more than a constant lag so all buffers are bounded. Certainly, the traditional capacity constraints are needed to have any possibility of stability. However, examples are known of layouts and systems with such simple control policies as clearing [3] or first-come first-served [6] which satisfy such constraints but are nevertheless unstable. Seidman and Humes [7] have observed examples of instability in the context of a production kanban system with orders based on consumption but serviced according to a clearing policy.

In this paper, we consider a single machine $\mathcal{M}$ using *signal kanbans* for control [4]. A signal kanban system differs from the more familiar *production kanban* system in that there is exactly one token(kanban) associated with each part type that the machine produces, instead of a separate kanban for each batch or each individual part [1].

The basic system we analyze consists of a buffer maintaining supply of several products (e.g., varieties of panels for an assembly line) and the machine (e.g., stamping press) which feeds it, controlled by a 'signal kanban' pull system. We begin by describing the setting and some variants; for further description, see [4].

Our investigation of signal kanban systems was motivated by the production control system formerly in use at the metal stamping line at Toyota Motor Manufacturing (Georgetown, KY). Signal kanban systems are used when a production process has a significant setup time (e.g., changing the die for stamping) or otherwise requires a minimum batch size. In this paper, we show that signal kanban systems are stable (subject only to a simple capacity condition) so the queue of backorders is bounded. Any backorders, however, can starve subsequent processes, so it is preferable that backorders be avoided altogether, although one also wishes to avoid unnecessary inventory costs. We will determine a minimal signal level which then ensures that the system will eventually settle into an operation with zero backorders, regardless of the initial state.

Section 2 further introduces our model of the signal kanban system. Section 3 presents the analytical results. In section 4, we extend the analysis to cyclic production policies.

2 Description of the control setting

The signal kanban control system is intended to govern the interface between a *demand stream*, involving requests for a variety ($k = 1, \ldots, K$) of products, and a shared resource $\mathcal{M}$ which is to provide each of these. It may alternatively be viewed as *production control* for the supplying machine or as *inventory control* for the set of intermediate buffers.

In the signal kanban system, the operation of $\mathcal{M}$ consists of production runs, each of a given part type, with output placed in the buffer designated for that part type. Each output buffer has a fixed reorder point r_k, such that when inventory in the buffer falls to the reorder point, a request to produce that part type again is issued in the form of a kanban (authorization). Typically, the kanban card or indicator token is physically attached to a part (or bin of parts) corresponding to that inventory signal level and is released when that particular part or bin of parts is consumed. Upon its release, the signal kanban enters a queue of other signal kanbans each of which is waiting to authorize $\mathcal{M}$ to produce a particular part type. The kanbans are then serviced in a first-come first-served manner; the machine never idles[1] when it has such authorization. In the standard version of this (as actually implemented: cf., e.g., [4]), there is a fixed batch size[2] $\hat{\beta}_k$ for the production run.

Notation: We let τ_k denote the *processing time* required to produce 'one unit' of product$_k$; our analysis is formulated to be relatively insensitive to the distinction between this discrete setting and a 'continuous flow' model. We let $u_k(t)$ denote the cumulative demand for product$_k$ (up to time t) and let $v_k(t)$ denote the cumulative production. Here $v_k(0)$ represents the initial inventory so, letting $x_k(t)$ denote the current buffer level, we have $v_k(0) = x_k(0)$. We denote by $\delta_k \geq 0$ the setup time for product$_k$.

Our assumption is that demand is filled immediately, utilizing prior production, when parts are available in the buffer. If parts are not available ($x_k(t) = 0$) for some product$_k$ when a request arrives for it, then this request is enqueued until it can be filled. Moreover, all subsequent demand (for all products) remains blocked in that *backlog queue* until this backordered product can be filled. We denote by $b_k(t)$ the current amount of backlogged demand. Note that if $x_k(t) > 0$ for every k, then $b_k(t) = 0$ for every k as desired. With this notation, we see that $[u_k(t) - b_k(t)]$ is the (cumulative) *fulfilled demand* and that, similarly, $[v_k(t) - x_k(t)]$ is the (cumulative) *utilized production*. These must match so we have the identity

$$\begin{aligned} &[u_k(t) - u_k(s)] - [b_k(t) - b_k(s)] = \\ &\qquad [v_k(t) - v_k(s)] - [x_k(t) - x_k(s)] \end{aligned} \tag{1}$$

for any $t \geq s \geq 0$.

Letting r_k denote the *signal level* (reserve) for product$_k$, we define

$$y_k(t) := \max\{0,\ x_k(t) - r_k\}, \qquad z_k(t) := \max\{0,\ r_k - x_k(t)\} \tag{2}$$

as the *reserve credit* and *reserve debit* (levels of inventory above and below the signal level, respectively) so $x_k(t) = r_k + y_k(t) - z_k(t)$. We assume that

[1] It would be possible to modify the argument and our hypotheses to take account of scheduled and unscheduled downtime as well as setup time, but we do not attempt this here.

[2] One may easily propose variants in which one might fill the buffer to a specified level instead. This would compensate for some variable draw down from the buffer while the kanban token is enqueued, although it slightly complicates the information flow in requiring some further 'signal' to terminate the run.

the buffer is never filled above its capacity φ_k so

$$0 \le y_k(t) \le \hat{\gamma}_k := \varphi_k - r_k. \tag{3}$$

If we have a fixed batch size $\hat{\beta}_k$) then the maximal buffer level (at the conclusion of the production run) can be at most $r_k + \hat{\beta}_k$ with a difference corresponding to the (possible) amount drawn down from the reserve during the period between t_* and the start of the run. This is because a production run occurs only if a signal kanban has been sent, necessarily meaning that $x_k(t_*) \le r_k$ at the moment t_* at which that signalling occurred. [3] Since the run is at most $r_k + \hat{\beta}_k$ for this fixed batch case, then we require that $\varphi_k := r_k + \hat{\beta}_k$ to get (3). If, on the other hand, we were to use a 'fill to capacity' strategy, then (3) is clear and we would equally obviously require a production run of length at least $\varphi_k - x_k(t_*)$ (again, the difference would correspond to an intermediate drawdown), which is bounded below by a 'minimum batch size' (production run for product$_k$) of $\hat{\beta}_k := \varphi_k - r_k$.

This completes the description of our signal kanban system.[4]

3 Stability

Clearly, $0 \le b_k(t)$ and $0 \le z_k(t) \le r_k$ (with $z_k(t) = r_k$ corresponding to an empty buffer). At this point we introduce the *effective backlog* $(b_k(t) + z_k(t))$.

[3]The continued consumption and the fixed batch size could leave $x_k(t) < r_k$ even upon completion of the run, so a token would be immediately re-issued, i.e., one could have two consecutive production runs for the same product. One might or might not then require an intervening setup time; our analysis is insensitive to this distinction.

[4]For this single machine setting there is a push system 'dual' to the pull system described. Consider a mixed product stream arriving to a set of supply buffers for a machine $\tilde{\mathcal{M}}$. If supply arrives for a buffer filled to its capacity φ_k, then it (and the subsequent supply stream) is enqueued as backlog. When a buffer level $\tilde{x}_k(t)$ reaches $\tilde{r}_k$, a signal is sent to authorize a run processing product$_k$ — either to clear the buffer or with a fixed batch size. (This would differ from the clearing systems considered in [5] in that (a) $\tilde{\mathcal{M}}$ idles if no buffer has reached its signal level and (b) the selection of runs is FCFS by signals.) One sees that this mirrors the behavior of $\mathcal{M}$: taking $\tilde{r}_k := \varphi_k - \tilde{r}_k$ and $\tilde{x}_k(t_0) = \varphi_k - x_k(t_0)$ initially and if the supply stream to $\tilde{\mathcal{M}}$ matches the demand stream for $\mathcal{M}$, then $\mathcal{M}$ and $\tilde{\mathcal{M}}$ behave identically so one has $\tilde{x}_k(t) = \varphi_k - x_k(t)$ afterward and stability for $\mathcal{M}$ and for $\tilde{\mathcal{M}}$ are equivalent.

The change in this effective backlog is given by rearranging (1) to get

$$
\begin{aligned}
&[(b_k(t) + z_k(t)) - (b_k(s) + z_k(s))] \\
&\quad = [u_k(t) - u_k(s)] - [v_k(t) - v_k(s)] + [y_k(t) - y_k(s)]\,.
\end{aligned} \tag{4}
$$

We also introduce the *adjusted processing time* $\hat{\tau}_k$, given by

$$
\hat{\tau}_k := \tau_k + \delta_k/\hat{\beta}_k. \tag{5}
$$

This is the maximum time per part required for producing a batch, with the setup amortized over the minimum batch size $\hat{\beta}_k$. We will use this as a normalization (weighting) for amounts of the various products, so when we consider a time interval $[s, s+T]$ which corresponds to a *complete production run* of any product$_k$ (including setup time, so $T = \delta_k + \tau_k\beta$) then the normalized production must be at least T — i.e., the run produces β_* units of product (necessarily with $\beta_* \geq \hat{\beta}_k$), then (5) implies that the normalized production $\hat{\tau}_k\beta_*$ is at least $\delta_k + \tau_k\beta = T$. It follows (by summing over such run intervals) that if $[s,t]$ $(t > s \geq 0)$ is any interval which contains no incomplete production runs for product$_k$, then

$$
\hat{\tau}_k\,[v_k(t) - v_k(s)] \geq \alpha_k[t-s] \tag{6}
$$

where α_k is the fraction of the total interval attributed to runs of product$_k$. The restriction on $[s,t]$ regarding non-interruption of runs can be omitted if we weaken (6) to

$$
\hat{\tau}_k\,[v_k(t) - v_k(s)] \geq \alpha_k[t-s-\delta_k]. \tag{7}
$$

It is convenient to use this normalization in combining the cumulative totals of demand and production for all products: summing over k,

$$
\hat{U}(t) := \sum_k \hat{\tau}_k u_k(t), \qquad V(t) := \sum_k \hat{\tau}_k v_k(t). \tag{8}
$$

Roughly, $[\hat{U}(t) - \hat{U}(s)]$ estimates the time which would be needed (allowing for setup times) for the machine $\mathcal{M}$ to process the demand arriving during the time interval $[s,t]$, but we note that this will not *exactly* correspond to such a processing time unless the production happens to break into a sequence of runs which each involve precisely the minimum batch size; a similar comment applies to $V(\cdot)$. This normalization does permit us to express conveniently our assumption on the total demand rate: we assume that there are constants $d_*, \xi \geq 0$ such that (for any time interval $[s,t]$ with $t \geq s \geq 0$) one has

$$\hat{U}(t) - \hat{U}(s) \leq d_*(t-s) + \xi. \tag{9}$$

Thus, d_* rate of (normalized) total demand while ξ provides some tolerance for demand fluctuations (burstiness). The *capacity constraint* we will need is then

$$d_* < 1. \tag{10}$$

Over relevant time intervals, this will ensure that, with some margin (sufficient to handle fluctuations and to eliminate any initial backlog), there will be enough time available for the demanded production.

We also define

$$B(t) := \sum_k \hat{\tau}_k \left[b_k(t) + z_k(t) \right], \qquad \Gamma := \sum_k \hat{\tau}_k \hat{\gamma}_k.$$

Here, $B(t)$ is the total 'generalized backlog', combining the actual enqueued backlog (if any) at time t with the 'reserve debit', and Γ is the total buffer capacity above the reserve levels — in each case normalized as a rough time estimate for the corresponding production with setup times.

Consider any time interval $[s, s+T]$ made up of a sequence of (complete) production runs so the machine $\mathcal{M}$ is continuously active (either in production or in setup). We multiply (4) by $\hat{\tau}_k$ and sum over k to get

$$B(s+T) \leq B(s) + [\hat{U}(s+T) - \hat{U}(s)] - [V(s+T) - V(s)] + \Gamma$$

in view of (3). Our assumption that $\mathcal{M}$ is continuously active during the interval ensures that (6) applies for each product with $\sum_k \alpha_k = 1$ so we have $V(s+T) - V(s) \geq T$. We use this and (9) above to obtain

$$B(s+T) \leq B(s) + (d_* - 1)T + [\xi + \Gamma]. \tag{11}$$

Since we have assumed (10), this bounds the length T of such an interval.

If, instead, one were to use (7) for a more general time interval, then, noting that the termination time t can interrupt at most one of the production runs involved, (11) becomes

$$B(t) \leq B(s) - (1-d_*)(t-s) + [\hat{\delta} + \xi + \Gamma] \leq B(s) + [\hat{\delta} + \xi + \Gamma] \tag{12}$$

where $\hat{\delta} := \max_k \delta_k$.

We summarize these results in the following theorem.

THEOREM 1: *Consider a machine $\mathcal{M}$ producing products according to any of the signal kanban production control system variants described above. Assume the arriving demand is bounded as in* (9) *with the capacity condition $d_* < 1$. Then the following statements are true:*

1. *Given the buffer levels and backorders at time t_0, the machine cannot remain continuously active (without idle time) for longer than*

$$T_0 := \frac{B(t_0) + [\xi + \Gamma]}{1 - d_*}\,. \tag{13}$$

2. *The backorder queue cannot be nonempty for longer than this T_0.*

3. *Given the initial state at time* 0*, the backorder queue remains bounded in time, so the system is stable.*

Statements 2 and 3 in the above theorem follow by noting that there can be a demand backlog only if some buffer$_k$ is empty ($x_k(t) = 0 < r_k$), in which case a signal kanban has necessarily been sent (with this order not yet filled) so $\mathcal{M}$ must be active. During any such interval, (12) bounds $B(t)$ as, at the initiation of this interval — unless we were to begin with initial backorders as part of $B(0)$ — we had $b_k(s-) = 0$ for each k so $B(s-) = \sum_k \hat{\tau}_k z_k(s) \le \sum_k \hat{\tau}_k r_k$; similarly, (13) bounds the duration of any interval with nonvanishing backorders.

In a quite similar spirit, we remark that $\mathcal{M}$ must be active whenever $B(t) \ne 0$ (as we must then have at least one $x_k(t) < r_k$) so, assuming we have $B(0) = 0$, (12) will apply to give $B(t) \le [\hat{\delta} + \xi + \Gamma]$ for all $t > 0$. Thus, if we were to have large enough reserves for each k, we would always have $z_k(t) \le B(t)/\hat{\tau}_k < r_k$ so $x_k(t) > 0$ and no demand backorders will ever occur: all requests arriving at the buffer can be filled immediately. This is summarized in the following theorem.

THEOREM 2: *Under the hypotheses of Theorem 1, suppose*

$$r_k > [\hat{\delta} + \xi + \Gamma]/\hat{\tau}_k \qquad \textit{for each } k. \tag{14}$$

Then, any initial backorder queue will empty as in (13) *and will remain empty thereafter.*

The required size of each reserve supply r_k seems necessary in the context of (9), where in our analysis we have asserted no control over arriving demand for individual products and permit the 'mix' to vary quite arbitrarily with time so each buffer$_k$ must maintain a reserve sufficient to insure against the possibility that the demand would be entirely for that product. We do not consider here the determination of a tighter set of maintained reserves (signal levels) when one would have such more detailed information available.

4 A Cyclic Policy

We next analyze a cyclic service (round robin) policy — which may be viewed, somewhat artificially, as a signal kanban policy in which each signal level is exactly at the buffer capacity φ_k so the kanban is immediately re-issued at the termination of a production run. The time period of a cycle is not fixed and there is no minimum batch size[5] here so (5) gives $\hat{\tau}_k = \infty$ and the normalization of the previous section is unavailable: we could not possible have (9), (10). Nevertheless, we will see that the conditions

$$\begin{aligned} u_k(t) - u_k(s) &\leq d_k(t-s) + \xi_k \qquad (t \geq s \geq 0) \\ \textstyle\sum_k \tau_k d_k &=: d_* < 1 \end{aligned} \tag{15}$$

will be sufficient to ensure stability. Note that (15) bounds the demand of each product$_k$ during $[s,t]$, whereas the less restrictive (9) controlled only the (normalized) *total* demand. Since $\tau_k d_k$ is the time necessary to process the nominal demand rate, then requiring $d_* < 1$ ensures sufficient capacity for this production with some margin. What we will show is that in this cyclic

[5] We admit the possibility that a run is initiated, but the buffer has remained at its capacity (no intervening demand) so there is no actual production — although the kanban will once again be immediately re-issued at the end of the setup time.

service policy, this margin enables the batch sizes to adjust automatically, even in the presence of allowable burstiness, so the setup times are just amortized with $\mathcal{M}$ continuously active.

We let s^ν denote the time of initiation of the ν-th cycle (starting, somewhat arbitrarily, with product$_1$) and let $\{t_k^\nu\}$ denote the termination times of the individual production runs. Thus, the length (time) of the run (including setup time) for product$_k$ is $\sigma_k^\nu = \left[t_k^\nu - t_{k-1}^\nu\right]$ and we have

$$t_k^\nu = s^\nu + \sum_{j\le k} \sigma_j^\nu \qquad s^{\nu+1} = t_K^\nu = s^\nu + \sum_k \sigma_k^\nu. \tag{16}$$

It is convenient to introduce the *reserve debits* z_k^ν at the cycle start s^ν, normalized by the actual processing time τ_k, so $z_k^\nu := \tau_k\left[\varphi_k - x_k(s^\nu)\right]$ and also set

$$\begin{aligned} \eta_k^\nu &:= \tau_k\left[u_k\left(t_k^\nu\right) - u_k\left(s^\nu\right) - d_k\left(t_k^\nu - s^\nu\right)\right], \\ \zeta_k^\nu &:= \tau_k\left[u_k\left(s^{\nu+1}\right) - u_k\left(t_k^\nu\right) - d_k\left(s^{\nu+1} - t_k^\nu\right)\right]. \end{aligned}$$

Note that

$$\begin{aligned} \zeta_k^\nu + \eta_k^{\nu+1} &= \tau_k\left[u_k\left(t_k^{\nu+1}\right) - u_k\left(t_k^\nu\right) - d_k\left(t_k^{\nu+1} - t_k^\nu\right)\right] \\ &\le \tau_k \xi_k \end{aligned}$$

by (15).

Since we are assuming 'fill to capacity' in the form $x_k(t_k^\nu) = \varphi_k$, we get

$$\begin{aligned} \sigma_k^\nu &= \delta_k + z_k^\nu + \tau_k\left[u_k(t_k^\nu) - u_k(s^\nu)\right] \\ &= \delta_k + \eta_k^\nu + z_k^\nu + \tau_k d_k \sum_{j\le k} \sigma_j^\nu, \\ z_k^{\nu+1} &= \tau_k\left[\varphi_k - x_k(s^{\nu+1})\right] \\ &= \tau_k\left[u_k(s^{\nu+1}) - u_k(t_k^\nu)\right] \\ &= \zeta_k^\nu + \tau_k d_k \sum_{j>k} \sigma_j^\nu. \end{aligned} \tag{17}$$

Note that this also assumes that all arriving demand is immediately filled, i.e., that the reserve levels φ_k are 'sufficient' — a point to which we shall return as the principal concern of our analysis.

Now define the matrices $\mathbf{A}$, $\mathbf{B}$ with entries

$$A_{j,k} := \begin{cases} \tau_j d_j & \text{if } j \le k \\ 0 & \text{else} \end{cases} \qquad B_{j,k} := \begin{cases} \tau_j d_j & \text{if } j > k \\ 0 & \text{else} \end{cases}.$$

Since (15) just ensures that $[\mathbf{I}-(\mathbf{A}+\mathbf{B})]$ (and also $[\mathbf{I}-\mathbf{A}]$) will be (column-wise) strictly diagonally dominant:

$$\sum_{j\neq k} |-A_{j,k}| < 1 - A_{k,k}$$

so $\bar{\lambda}(\mathbf{A}+\mathbf{B}) < 1$ (where $\bar{\lambda}(\cdot)$ denotes 'spectral radius' = largest | eigenvalue |) then $\mathbf{W} := \mathbf{B}(\mathbf{I}-\mathbf{A})^{-1} = \sum_n \mathbf{B}\mathbf{A}^n \geq \mathbf{0}$ is well-defined and we may apply a lemma[6] appearing in [2] to see that $\mathbf{W}^\nu \to \mathbf{0}$ and $(\mathbf{I}-\mathbf{W})^{-1} \geq \mathbf{0}$.

We now introduce a variety of vectors: $\vec{\boldsymbol{\delta}}$, $\vec{\boldsymbol{\xi}}$, $\vec{\boldsymbol{\sigma}}^\nu$, $\vec{\mathbf{z}}^\nu$, $\vec{\boldsymbol{\eta}}^\nu$, $\vec{\boldsymbol{\zeta}}^\nu$ with entries δ_k, $\tau_k\xi_k$, σ_k^ν, z_k^ν, η_k^ν, ζ_k^ν, respectively. Note that $\vec{\boldsymbol{\zeta}}^\nu + \vec{\boldsymbol{\eta}}^{\nu+1} \leq \vec{\boldsymbol{\xi}}$ (interpreting such inequalities entrywise) and that (17) becomes

$$\begin{aligned} \vec{\boldsymbol{\sigma}}^\nu &= \vec{\boldsymbol{\delta}} + \vec{\boldsymbol{\eta}}^\nu + \vec{\mathbf{z}}^\nu + \mathbf{A}\vec{\boldsymbol{\sigma}}^\nu = (\mathbf{I}-\mathbf{A})^{-1}\left[\vec{\boldsymbol{\delta}} + \vec{\boldsymbol{\eta}}^\nu + \vec{\mathbf{z}}^\nu\right], \\ \vec{\mathbf{z}}^{\nu+1} &= \vec{\boldsymbol{\zeta}}^\nu + \mathbf{B}\vec{\boldsymbol{\sigma}}^\nu = \vec{\boldsymbol{\zeta}}^\nu + \mathbf{W}\left[\vec{\boldsymbol{\delta}} + \vec{\boldsymbol{\eta}}^\nu\right] + \vec{\mathbf{z}}^\nu. \end{aligned} \tag{18}$$

A simple induction argument then gives

$$\vec{\mathbf{z}}^\nu \leq \vec{\mathbf{z}}^* + \mathbf{W}^\nu(\vec{\mathbf{z}}^0 - \vec{\mathbf{z}}^*) + \left[\mathbf{W}^\nu\vec{\boldsymbol{\eta}}^0 - \vec{\boldsymbol{\eta}}^\nu\right]$$

$$\text{with } \vec{\mathbf{z}}^* := [\mathbf{I}-\mathbf{W}]^{-1}\left[\vec{\boldsymbol{\xi}} + \mathbf{W}\vec{\boldsymbol{\delta}}\right]$$

so $\vec{\mathbf{z}}^\nu \leq \vec{\mathbf{z}}^* + \mathbf{W}^\nu\left[\vec{\boldsymbol{\xi}} + \vec{\mathbf{z}}^0 - \vec{\mathbf{z}}^*\right]$ for each $\nu = 0, 1, \ldots$. Assuming we start with an initial state (initial buffer levels) such that $\vec{\mathbf{z}}^0 \leq \mathbf{W}\vec{\boldsymbol{\delta}}$, we then will have $\vec{\mathbf{z}}^\nu \leq \vec{\mathbf{z}}^*$, i.e., $\vec{\mathbf{z}}^*$ is a bound on the processing time for the worst backorder which could exist at the start of a cycle. Using this in (18) then gives

$$\begin{aligned} \vec{\boldsymbol{\sigma}}^\nu \leq\ & \vec{\boldsymbol{\sigma}}^* := (\mathbf{I}-\mathbf{A})^{-1}[\vec{\boldsymbol{\delta}} + \vec{\boldsymbol{\xi}} + \vec{\mathbf{z}}^*] \\ & = (\mathbf{I}-\mathbf{A})^{-1}[\vec{\boldsymbol{\delta}} + \vec{\boldsymbol{\xi}}] + [\mathbf{I}-(\mathbf{A}+\mathbf{B})]^{-1}(\vec{\boldsymbol{\xi}} + \mathbf{W}\vec{\boldsymbol{\delta}}) \end{aligned} \tag{19}$$

for each ν so σ_k^* is an upper bound on the processing time for product$_k$ during any cycle and $T_c := \sum_k \sigma_k^*$ is a bound on the cycle length. We can simplify (19) by noting that it corresponds to a worst case, taking the first condition

---

[6] **LEMMA (Seidman):** *Suppose* $\mathbf{A}, \mathbf{B} \geq \mathbf{0}$ *(entrywise inequality) with* $\bar{\lambda}(\mathbf{A}+\mathbf{B}) = \alpha < 1$. *Then we can set* $\mathbf{W} := \mathbf{B}(\mathbf{I}-\mathbf{A})^{-1}$ *with* $\bar{\lambda}(\mathbf{W}) \leq \alpha$ *so* $\sum_n \mathbf{W}^n$ *converges.*

of (15) to be an equality over the cycle period and looking at the resulting periodic solution. Since, now using the stability we have shown, production matches arriving demand for this, we have $\sigma_k^* = \delta_k + \tau_k[d_k T_c + \xi_k]$ so, summing and solving for T_c,

$$T_c = \frac{\sum_k(\delta_k + \tau_k \xi_k)}{1 - \sum_k \tau_k d_k}. \tag{20}$$

Now observe that, since the production run for product$_k$ fills the buffer, the reserve debit can never have exceeded the amount of production (batch size β_k^ν) which gives

$$\varphi_k - x_k(t) \le \beta_k^\nu = \frac{\sigma_k^\nu - \delta_k}{\tau_k} \le \frac{\sigma_k^* - \delta_k}{\tau_k} = d_k T_c + \xi_k$$

for $t_k^{\nu-1} \le t \le t_k^\nu$. Thus, we have justified our assumption that we proceed without interruption (all demand filled immediately) if the buffer capacities satisfy

$$\varphi_k > \varphi_k^* := d_k T_c + \xi_k \tag{21}$$

for each k with T_c given by (20).

We can summarize our results in the following theorem.

THEOREM 3: *Consider a machine $\mathcal{M}$ processing an ordered set of K product types. Suppose these are serviced according to a fixed cycle and, when servicing product$_k$, the machine fills the buffer to level φ_k. Assume the demands satisfy* (15) *and that initially $\vec{\mathbf{z}}^0 \le \mathbf{W}\vec{\boldsymbol{\delta}}$; assume the buffer capacity φ_k for each product$_k$ satisfies* (21). *Then the following statements are true for all subsequent time:*

1. *The maximum time between completion of production runs for any product$_k$ is T_c given by (20).*

2. *Demand is always filled immediately and backorders never occur.*

References

[1] J. Browne and J. Harhen and J. Shivnan, *Production Management Systems*, Addison Wesley, New York, 1988.

[2] A.F.P.C. Humes and C. Humes Jr., *A clearing round-robin-based stabilization mechanism*, in Proc. Allerton–94, Allerton (IL), 1994.

[3] P.R. Kumar and T.I. Seidman, *Dynamic instabilities and stabilization methods in distributed real time scheduling of manufacturing systems,* IEEE Trans. Autom. Control **AC-35**, pp. 289–298 (1990)

[4] Yasuhiro Monden *'Toyota Production System'* (2^{nd} edition), Inst. Industrial Engineers Press, Norcross (GA), 1993.

[5] J.R. Perkins and P.R. Kumar, *Stable distributed real-time scheduling of flexible manufacturing/ assembly/ disassembly systems*,
IEEE Trans Autom. Control **AC-34**, pp. 139–148 (1989).

[6] T.I. Seidman, *'First Come, First Served' can be unstable!*, IEEE Trans. Autom. Control **AC-39**, pp. 2166–2171 (1994).

[7] T.I. Seidman and C. Humes, Jr., *Some kanban–controlled manufacturing systems: a first stability analysis,* IEEE–TAC, **41**, pp. 1013–1018 (1996).