MANAGING PREVENTIVE MAINTENANCE ON A DISASSEMBLY LINE USING MULTI-KANBAN MECHANISM

K. Nakashima¹, S.M. Gupta²

¹Department of Industrial Engineering and Management, Kanagawa University, Yokohama, 221-8686, Japan
²Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA

Abstract
Your This paper investigates the effect on the performance of two production control mechanisms, viz. push and pull, on a disassembly line where interruptions are caused by preventive maintenance. In the push system, blocking and starvation are compensated by the large amount of inventories stored in the buffers. Thus, the system must disassemble components prior to the arrival of actual demand. Alternatively, the multi-kanban pull system offsets the blocking and starvation by manipulating the kanban routing. Hence, it directs kanban to the proper workstation based on system’s current status. Compared to the push system, the pull system tends to keep smaller amounts of inventories while attempting to service the actual demand. Through a case example, we present advantages and disadvantages of both systems.

Keywords:
Traditional Push system (TPS), Flexible Kanban system (FKS), preventive maintenance, processing time variations, demand variation.

1 INTRODUCTION
There has been a lot of talk about sustainable manufacturing as of late [1]. One of the pillars of sustainability has to do with environmentally conscious manufacturing (ECM). ECM has been a subject of interest during the recent past. Of the many important aspects of ECM, product recovery plays a major part in separating reusable components and materials from end-of-life products and transporting them to remanufacturing and recycling industries. Selective disassembly helps in increasing the recovery rate of valuable components and materials that are highly demanded and in minimizing costs of removing unwanted components. Among several disassembly settings that are employed in selective disassembly, disassembly line is one of the most commonly used settings. It is also considered one of the most suitable and effective ways to disassemble products in large quantity. A disassembly line faces numerous uncertainties and difficulties that are not present in an assembly line. These include arrival pattern of EOL products and demand for components, and unconventional fluctuation in inventory due to the disparity in yields and demands.

Disassembly line faces two major obstacles. One obstacle occurs due to the uncertainty in the arrivals of demand and supply. EOL products usually arrive in unpredictable conditions. They may also comprise of unknown number of components. These can create unpredictable flows of components in the disassembly line. Demands can also create further complications because they may arrive at varying levels along the line. This causes irregular variations in the number of components being supplied and demanded. Another obstacle occurs due to the nature of the disassembly process. The disassembly process often delivers two categories of components, viz. target component and residual component. The target component is usually in high demand while the residual component is in low demand or not demanded.

Similar to an assembly line setting, there are two types of control mechanisms in a disassembly line setting, viz., the push system and the pull system. A push system is easy to implement but is generally not efficient in the disassembly environment as it tends to generate large amounts of inventories. However, this may be beneficial during preventive maintenance. A pull system, on the other hand, operates efficiently and generates less amounts of inventory. Yet, with the occurrence of interruptions, this can be problematic. Most production control tools that implement pull mechanisms in the assembly line settings are not practical for the disassembly line settings. Moreover, some disassembled residuals have short shelf-lives. Some incur high storage cost especially if the amount of disassembled residuals is large.

In this paper, we demonstrate how a pull system, such as a Kanban mechanism, could be modified and implemented efficiently in disassembly line settings. We study the effects of interruption caused by preventive maintenance on the performance of a disassembly line. To that end we present a methodology to implement a kanban system in a disassembly line setting with the occurrence of preventive maintenance. We consider a case example to illustrate the implementation of the methodology and obtain results using simulation. We compare the performance of the pull system with that of the push system.

2 LITERATURE REVIEW
Several studies have recently emerged that address various aspects of product recovery. Comprehensive reviews on the issues in environmentally conscious manufacturing and product recovery were provided by Gungor and Gupta [2] and Ilgin and Gupta [3]. Disassembly is an important issue within the area of product recovery. Researchers have studied various aspects of disassembly including sequencing [4, 5], scheduling [6, 7, 8], disassembly line [9, 10], disassembly line balancing [11, 12], disassembly to order systems [13, 14], and automated disassembly [15, 16]. The interested reader is referred to recent books by Lambert and Gupta [17] and McGovern and Gupta [18] for further information on different aspects of disassembly.

Kanban is a tool that is often used to control the flow in an assembly line. However, traditional kanban system cannot be used directly in a disassembly line. It is due to the sudden or large variations in demands in disassembly situation that greatly reduce the applicability of traditional kanban system. Several modified kanban methodologies were developed to deal with demand variability while preserving the advantages of traditional kanban system. Gupta and Al-Turki [19, 20, 21] and Gupta et al. [22] proposed the concept of the flexible kanban system (FKS)
in various environments involving uncertainties and interruptions. They demonstrated that in such environments, FKS outperforms the traditional kanban system. Other adaptive and reactive techniques to adjust the number of kanbans were proposed by Tardif and Maaseidvaag [23], Takahashi and Nakamura [24, 25, 26] and Takahashi et al. [27]. Kizilkaya and Gupta [28], Udomsawat and Gupta [29, 30], Nakashima, Kojima and Gupta [31]. A comprehensive overview of studies on modified kanban systems was provided by Lage Jr. and Godinho-Filho [32].

Maintenance is defined as a series of actions taken during the operation of a line where it is necessary to interrupt the flow of production. Several studies have considered maintenance models. Shafiee and Chukova [33] proposed a taxonomy scheme to classify various maintenance models.

3 DISASSEMBLY LINE
Disassembly line consists of a series of workstations operating in a sequence to disassemble the end-of-life (EOL) products into subassemblies and/or components. The line usually faces numerous problems. The most important of all is the disorderly fluctuation in inventory levels. The fluctuation is a result of two unique characteristics of a disassembly line, viz., multi-level arrivals of EOL products and multi-level arrivals of demands. Figure 1 depicts a typical structure of a disassembly line.

3.1 Product Arrival Pattern
Based on the type of EOL product, it may enter the disassembly line at any of the workstation, not just the first workstation. The arriving products may consist of different combinations of components from a given set of components. From a set of \( N \) components, the total number of possible combinations of components, \( Q(N) \), is given by

\[
Q(N) = 2^N - N - 1
\]  

For example, a set of 4 components (A, B, C, D) can produce up to 11 possible product combinations (viz., AB, ABC, ABCD, ABD, AC, ACD, AD, BC, BCD, BD, CD). By adding one more component to the set, the number of possible combinations increases to 26. It is therefore clear that the number of combinations increases exponentially with the increase in the number of components. The workstation where a product enters the disassembly line depends on the combination of the components in the product. Thus, for example, consider a disassembly line with three workstations. If component A is disassembled at workstation 1, component B is disassembled at workstation 2 and components C and D are disassembled at workstation 3, then a product arriving at the disassembly line consisting of components B, C, and D does not have to go to workstation 1 at all. It could enter the disassembly line directly at workstation 2. Considering the same example, if an arriving product consists of components A, C, and D, it would have to enter workstation 1. However, after getting processed at station 1, it could skip workstation 2 entirely. Both these situations destabilize the disassembly line by causing an overflow of materials at one workstation while starving some other workstations leading to undesirable fluctuations of inventory in the system. It is therefore crucial to balance the line and manage the materials flow of the line.

3.2 Demand Arrival Pattern
The arrival pattern of demand in a disassembly line is much more complicated than in a typical assembly line. In addition of having multi-level arrivals of EOL products, depending on what is demanded, the demand could also occur at any workstation, not just the last workstation. In most assembly lines, demand arrives only at the last workstation. However, in that case, even if multilevel arrival of demand were considered, its effect would be benign because the product does not go forward from there on as it is taken off the line to fulfill the demand. In a disassembly line setting, however, the arrivals of external demand at workstations other than the last one creates a disparity between the number of demanded components and the number of partially disassembled products. Thus, if the system responds to every request for components, it would end up with a significant amount of extra inventory of components that are in low demand.

3.3 Control Mechanism
Since service level is important and it is necessary to maximize it, becomes necessary to develop a good methodology to control the system and find a way to manage the extra inventory produced. For decades, a push system has been the dominant controlling mechanism for assembly lines. The system generates schedule for raw materials to arrive based on the forecasted demand. Materials and finished goods are stocked in order to fulfill the future demand. As mentioned earlier, disassembly line tends to generate residual inventory. These residuals are of concern because disassembly process, storing and moving of materials are labor intensive. It costs money to disassemble unwanted components that are not likely to sell. Furthermore, the line serves demands at multiple levels. In such an environment, an operation using a push system can be problematic and costly. On the other hand, a pull system offers the benefits of carrying minimal inventory and reduction in cycle time variation. The system initiates production only when an actual demand arrives. In this study, we modify a kanban system to perform in a disassembly environment so that the benefits of a pull system could be seized in disassembly environment.

3.4 Disassembly Line and Interruptions
A pull system employing multi-kanban mechanism [29] offers a viable solution that is capable of keeping up with the demand rate in such situations without carrying large amount of inventory. However, disassembly of EOL products in large quantity can be challenging when scheduled or sudden interruption occurs. The multi-kanban mechanism could help in managing disorderly fluctuation of inventory level when interruption occurs in disassembly line. Thus, when dealing with preventive maintenance, the system can also take advantage of these characteristics. Besides, knowing when and where system will be inoperative in advance can help in manipulating kanban routing.
4 MULTI-KANBAN MECHANISM
As mention earlier, pull mechanism is more likely to perform better than push mechanism in a disassembly line. The mechanism only produces when and where there is a need. The actual demand triggers the process and causes a flow of materials throughout the system. The pull mechanism is designed to control the inventory levels. It relies on the consistency of raw materials supplies and agility of the server. Among several tools to implement pull mechanism, kanban is one of the most commonly used. However, once implemented in a disassembly line setting, it is fraught with numerous uncertainties. A modification of the mechanism is therefore needed to improve its performance by reducing these difficulties and allowing the system to operate at its best.

4.1 Material Types
There are two basic types of materials in the system, viz., components and subassemblies. A component is a single item that cannot be further disassembled. It is placed in the component buffer waiting to be retrieved via a customer demand. On the other hand, a subassembly is something that can still to be disassembled. Subassembly is composed of at least two components. Both types of materials can be further distinguished as regular or overflow items. Regular items are what customers or downstream workstations demand. In order to fulfill the demand, a server must disassemble the demanded component or subassembly. The residual item from this disassembly process that does not fulfill any request is called overflow item. Because the disassembly process is initiated by a single kanban, the overflow item will not have a kanban attached to it. However, the overflow item is routed in the same way as the regular item. The only difference between them is that the overflow item is given priority of being retrieved after it arrives at its buffer. It should be noted that, as long as there is an overflow item in the buffer, its demand would not initiate any further disassembly process. This will help the system eliminate any extra inventory first that is caused due to unbalanced demands.

4.2 Kanban Types
Corresponding to material types, there are two basic types of kanbans in the system, viz., component kanbans and subassembly kanbans. A component kanban is attached to a disassembled component that is placed in the component buffer of the workstation where it is disassembled. Similarly, a disassembly kanban is attached to a residual subassembly that is placed in the subassembly buffer of the workstation where it was separated from the component. A component placed in a component buffer can be retrieved by an external demand. When authorized, a subassembly placed in the subassembly buffer is routed for disassembly to the next workstation based on its disassembly sequence.

At the first workstation, products arrive only from outside sources. However, at any other workstation \( i \), where \( 1 < i \leq N-1 \), there are two possible types of arrivals. The first type is a subassembly that arrives from an upstream workstation, called internal subassembly. There is always a subassembly kanban attached to an internal subassembly. The second type is a product (or subassembly) that arrives from outside sources, called external subassembly. There is no kanban attached to an external subassembly. This is also true of the products arriving from external sources to the first workstation. As long as there is an external product or subassembly available at an input buffer, the system will process it first before processing any available internal subassembly. This will avoid unnecessary pulling of an internal subassembly from an upstream workstation. Thus, the number of kanbans attached to internal subassemblies will remain constant throughout the process. Figure 2 illustrates the kanbans and materials flow in a disassembly line.

4.3 Kanban Routing Mechanism
Consider workstation \( j \), where \( 1 < j \leq N-1 \). When a demand for component \( j \) arrives at the component buffer of workstation \( j \), one unit of component \( j \) is retrieved and the component kanban attached to it is routed to the most desirable workstation. The procedure for determining the most desirable workstation to route component kanban \( j \) is given below. (Note that this procedure is not applicable to component kanbans \( N-1 \) and \( N \). In both cases the kanbans are routed to the input buffer of the last workstation).

A component kanban originating from workstation \( j \) will be routed to a workstation \( i \), where \( 1 < i < j \), or workstation \( j \) depending on the availability and the desirability of the subassembly that contains component \( j \). Routing component kanban \( j \) to workstation \( i \), where \( 1 < i \leq (j-1) \), will result in an immediate separation of component \( j \) from component \( i \). Thus, the only subassembly located at the input buffer of workstation \( i \) that would be useful is a subassembly that contains only components \( i \) and \( j \). If this type of subassembly exists in the input buffer of workstation \( j \), then workstation \( j \) is qualified. Similarly, if there is at least one subassembly in the input buffer of workstation \( j \), then workstation \( j \) is qualified.

Next, we need to select the most desirable workstation to route component kanban \( j \) to, among the qualified ones, such that, if chosen, will cause the least amount of extra inventory in the system. Choosing workstation \( i \) will increase the inventory level of component \( i \) by an additional unit. Thus, the best workstation \( j \) is the one that is most starving for its component. By checking the backorder level for demand \( i \), we could determine the most starving workstation. If there is a tie, select the most downstream workstation. Choosing workstation \( j \) will create a residual subassembly that will be further disassembled at downstream workstations. If workstation \( j \) is chosen, then a proper subassembly must be chosen to disassemble. For example, if a backorder exists at the component buffer of workstation \( k \), where \( j < k \leq (N-1) \), then, if available, we might try to disassemble a subassembly that contains only components \( i \) and \( k \). If more than one workstation qualify as starving workstations, then the one that is most starving among them is chosen. If there is a tie, then the most downstream workstation is selected.

We can now compare the starving levels of workstations \( i \) and \( j \). If the highest starving level of workstation \( i \) is greater than or equal to the starving level of workstation \( j \) then we will route the component kanban \( j \) to workstation \( i \), otherwise, we will route it to workstation \( j \). Note that whenever an external subassembly is available, it will
always be chosen first. Internal subassemblies will only be used when no external subassembly of the desired kind is available. Subassembly kanbans are routed in a fashion similar to component kanbans. Figure 3 shows a concept of the Multi-Kanban Mechanism.

4.4 Selection of Products

Because we allow multiple combinations of products, the worker may have several options when selecting the product for disassembly. If the authorization of disassembly is initiated by the subassembly kanban \((j_x)\), which can occur only at workstation \(i\), where \(1 \leq i < j\), the workers will have no option but to select the subassembly that results in immediate separation of subassembly \((j_x)\), viz., subassembly \((j_y)\). If the authorization of disassembly is initiated by component kanban \(j\) at workstation \(i\), where \(1 \leq i < j\), the worker will have to remove subassembly \((j_y)\) from the product buffer with no other options because the only subassembly that results in immediate separation of component \(j\) is the subassembly \((j_y)\). However, if the component kanban \(j\) arrives at workstation \(j\), there are multiple options because every subassembly located in the product buffer contains component \(j\) and always results in immediate separation of component \(j\). In this case, we determine whether or not the residual that is created by the disassembly will result in overflow of inventory. We choose the subassembly \((j_y)\) where \(x\) is the most desirable residual ranking based on the request of subassembly kanban \(x\) at workstation \(j\) (existing kanban \(x\) at the workstation \(j\)) or current inventory level of subassembly (component) \(x\), respectively.

4.5 Determining the Kanban Level

The kanban level plays an important role in the multi-kanban mechanism as it maintains a proper flow of components and subassemblies at a desired level throughout the system. It can be determined by considering product arrival rate, demand arrival rate and disassembly time. The number of kanbans for both the component kanban, \(k_i\) and the subassembly kanban, \(k_j\), can be computed, at any point in the disassembly line, using the following general expressions:

\[
K_i = \max(1, R_i / F_i)
\]

\[
K_j^* = \max(1, R_j^* / F_j^*)
\]

Where \(R\) is the request rate of component \(i\), \(F_i\) is the furnish rate of component \(i\), \(R_j^*\) is the request rate of subassembly \(j\), and of \(F_j^*\) is the furnish rate subassembly \(j\). These request rates and furnish rates can be calculated as follows:

\[
R_i = D_i, \quad \text{for } 1 \leq i \leq N
\]

\[
F_i = \sum_{w=1}^{i} S_{(i,w)}, \quad \text{for } 1 \leq i \leq N
\]

\[
R_j^* = S_j, \quad i \text{ is the next component to be disassembled in the sequence}
\]

\[
F_j^* = a_j^* + \sum_{w=1}^{j} S_{(j,w)}, \quad i \text{ is the latest component disassembled in the sequence}
\]

where \(d\) is the demand arrival rate of component \(i\), \(S_{(i,w)}\) is the disassembly rate of component \(i\) at workstation \(w\), \(s_j\) is the disassembly rate of subassembly \(j\), \(a_j^*\) is the arrival rate of subassembly \(j\) (from external source), \(m\) is the current workstation index, \(N\) is the maximum number of component, and \(N-1\) is the maximum number of workstation. For the case of component kanban, which is requested only from a single source, request rate is equal to the customer demand arrival rate. However, because the component kanban arrives from several sources in the system, the furnish rate is the summation of arrival rates from all possible sources. For the case of subassembly kanban, the furnish rate is influenced by both the disassembly rate and the external subassembly arrival rate. Thus, we take all external and internal arrival rates of subassemblies at the buffer into account. Similarly, the two requesting sources, viz., the demand for target component and the demand for residual subassembly affect the request rate. The number of kanbans is determined at the beginning of the disassembly process. It is clear that demand, supplies, disassembly time, and product structure, all affect the computation of the number of kanbans.

4.6 Coping with Preventive Maintenance of a Server

When there is a scheduled maintenance of a server, the mechanism copes with such situation in two stages. At the first stage, the mechanism calculates amount of extra components will be demanded at the starving workstation during the maintenance. The mechanism takes into consideration of disassembly rate, demanding rate, and current inventory level of that component. Also at this step, the mechanism calculates amount of extra subassemblies will be disassembled and routed to the blocking workstation. Again, it takes into account of disassembly rate, demanding rate, and current inventory level of that subassembly. Then, at a required amount of time prior to the maintenance, the mechanism enters

![Figure 4. Coping with the Preventive Maintenance on a Server: Stage 1.](image)

The second stage involves routing the kanban from the starving buffer to other candidate workstations according to rate of demand and supply assign by the first stage. The comparison criteria remain unchanged. However, the workstation that is on maintenance is excluded and does not qualify as a candidate. That is, the mechanism takes advantage of the availability of multiple candidates and
requests the part from other workstations that are still in operating. A workstation that is blocked by the maintenance is acceptable as a candidate. However, in a situation where the needed subassemblies are abundant at a workstation, it would most likely become the desirable workstation because the mechanism would consider it to be the one that has a tendency to generate the residual subassembly at a higher rate. Figure 5 demonstrates the example of a blocking workstation and starving buffers. Solid arrows show a sample of how a starving buffer C could receive component C from two candidate workstations (viz., 1 or 2) upstream from the blocking workstation 3. Also, in this case, candidates for component kanban B can be either workstation 1 or 2. The mechanism weighs the priority between the demand of residual parts at each workstation, viz. component A at workstation 1, and subassembly DEF or component B at workstation 2.

preventive maintenance mode. In this mode, all the calculated extra demand and supply are added to the actual rate of demand and supply from external source.

5 CASE EXAMPLE AND SOLUTION METHODOLOGY

To demonstrate how the multi-kanban model can be applied to the disassembly line with occurrence of preventive maintenance, we consider a disassembly of six different EOL products (viz., ABCDE, ABC, AC, BCDE, BC, and CDE). They are made up of various combinations of components from a set of five possible components, viz., A, B, C, D, and E, that have to be disassembled to fulfill the demands for each of the components. The disassembly line has 4 workstations. The input location for an EOL product depends on its configuration. According to the precedence relationships of that product, the input location is the most upstream workstation that disassembles the first component. For simplicity, we assume that it takes the same amount of time to disassemble each component.

The component disassembled at a workstation, $s$, is placed in the component buffer, $B_s$. The rest of the subassembly is routed to the subassembly buffer, $B_{ij}$, corresponding to the next component to be disassembled. Only one type of component is disassembled at a given workstation except when there are only two components left in the product. At each workstation, there are two types of output buffers, viz., component buffer and subassembly buffer. The subassembly buffer becomes the input buffer for the subsequent workstation to further disassemble the subassembly according to its disassembly sequence.

In studying the case example using simulation, the following assumptions were made:

- Customer backorder is allowed.
- External demand is for component only and can arrive at any workstation.
- Components must be disassembled according to their precedence relationships one type at a time until the last component in disassembly sequence is disassembled.
- Products and subassemblies may enter at any workstation along the line depending on its configuration.
- Only maintenance on single workstation is allowed.

We used ARENA® software [34] to simulate the model. We ran two sets of experiments representing the traditional push system (TPS) and the multi-kanban system. For each experiment, we collected the data over a 24-hour period. In the push system, all arriving products are processed continuously in the order of their arrival. The demand is fulfilled as soon as the components are available. In the multi-kanban pull control system, we utilize smart-routing for subassembly kanban (as explained in the Kanban Routing Mechanism subsection) in order to reduce the inventory built up caused by disparity in demands among components. We also utilize product selection method (as explained in the Selection of Products subsection). In these experiments, statistics on the following two performance measures were collected: system’s ability to fulfill demand and average inventory level. Maintenances on the third workstation are scheduled to occur three times daily and eight hour apart. Each maintenance incidence takes an average of 30 minutes to bring machine up to operating condition. Both TPS and MKS systems experience maintenance and at the same location. Table 1 shows the input data for the disassembly line. The number of kanbans is calculated using the suggested method.

<table>
<thead>
<tr>
<th>Product, Subassembly, or Component</th>
<th>Mean EOL Product Arrival Rate</th>
<th>Mean Disassembly Time (minutes)</th>
<th>Mean Demand Arrival Rate</th>
<th>Number of Kanbans</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDE</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ABC</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AC</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BCDE</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>CDE</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>BC</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>15</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>15</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>15</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>15</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>15</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
6 EXPERIMENTATION AND RESULTS

The simulation results are shown in Figures 6 and 7. It is clear from the figures that the multi-kanban mechanism significantly reduces average inventory while maintaining the components' demands. In TPS, the system builds up inventory in order to fulfill components' demands. Hence, a fluctuation in demands is coped by the large amount of inventory. In this case, the interruption caused by preventive maintenance has minimal effect on the service level provided that the component buffer size is large enough.

The multi-kanban mechanism deals with fluctuation among component demands and subassembly supply by manipulate routing mechanism of the kanbans. Prior to the schedule maintenance, the mechanism increases activity in disassembly process just enough for the system to replenish the demand while the station is down. The system also reduces disassembly output of subassembly that will cause blocking in just right amount and at the right time. This results in the lower amount of inventory needed to prevent blocking and starving during the scheduled maintenance. In the case example considered here, a component kanban C could be routed to either workstation 1 or workstation 2 or workstation 3. By examining the number of parts being requested in real time, the system selected the appropriate destination for the kanban. For the example considered, the system was able to reduce the inventory levels by an average of 73% while fulfilling components’ demands using only three or kanbans in the system.

7 CONCLUSIONS

Regardless of serious complications a disassembly line faces in attempting to manage fluctuation in inventory, this paper demonstrated that pull system could be adapted to significantly improve the performance of the line. With the help of simulation and an example, it was shown that the proposed adaptation of multi-kanban mechanism could be implemented effectively in the disassembly line with preventive maintenance is incorporated into. The multi-kanban mechanisms relies on real time manipulating of kanban routing and preserves serviceability to customer comparable to push system.

8 ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI Grant Number 25242031.

9 REFERENCES


