

January 01, 2004

Push and pull control systems in disassembly lines

Surendra M. Gupta
Northeastern University

Gun Udomsawat
Northeastern University

Yousef A. Y. Al-Turki
Northeastern University

Recommended Citation

Gupta, Surendra M.; Udomsawat, Gun; and Al-Turki, Yousef A. Y., "Push and pull control systems in disassembly lines" (2004). . Paper 97. <http://hdl.handle.net/2047/d10013684>



Laboratory for Responsible Manufacturing

Bibliographic Information

Udomsawat, G., Gupta, S. M. and Al-Turki, Y. A. Y., "Push and Pull Control Systems in Disassembly Lines", *Proceedings of the 2004 Decision Sciences Institute Conference*, Boston, Massachusetts, pp. 7811-7816, November 20-23, 2004.

Copyright Information

Copyright 2004, Surendra M. Gupta.

Contact Information

Dr. Surendra M. Gupta, P.E.
Professor of Mechanical and Industrial Engineering and
Director of Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, U.S.A.

(617)-373-4846 **Phone**
(617)-373-2921 **Fax**
gupta@neu.edu **e-mail address**

<http://www.coe.neu.edu/~smgupta/> **Home Page**

PUSH AND PULL CONTROL SYSTEMS IN DISASSEMBLY LINES

Gun Udomsawat, Northeastern University, Boston, MA 02115, gudomsaw@coe.neu.edu, (617)-373-7635

Surendra M. Gupta*, Northeastern University, Boston, MA 02115, gupta@neu.edu, (617)-373-4846

Yousef A. Y. Al-Turki, King Abdulaziz City for Science and Technology, Riyadh, (Saudi Arabia)

ABSTRACT

Increasing environmental concerns during the last decade have caused many governments to persuade manufacturers to take back used products so that the components and materials recovered from the products could be reused and/or recycled. The disassembly process is the first step for recovering components and materials. We discuss some of the complications in planning and scheduling of disassembly on a disassembly line. We show how to overcome them by implementing a multi-kanban mechanism in the disassembly line setting. We then investigate the multi-kanban mechanism using simulation and demonstrate that in the multi-product, multi-demand environment, such mechanism performs superior to the traditional push system.

Keywords: Disassembly Line, Just-In-Time Systems, Production and Inventory Control Systems.

INTRODUCTION

Disassembly is the process of separating the desired components, subassemblies, and materials from end-of-life or returned products. Its key objectives include recovering components and/or materials in order to preserve their functionality and/or values for future use [4]. For further information on disassembly and product recovery, see Brennan et al. [1] and Lambert [11]. Disassembly line has become the subject of recent interest [2], [3]. A disassembly line is quite different from an assembly line in terms of material movement, demand arrival and inventory level fluctuation. For example, as opposed to the normal "convergent" flow on an assembly line, in a disassembly line, the flow process is "divergent" (a single product is broken down into many subassemblies and parts). A disassembly line has significant inventory problems because of the disparity between the demands for certain parts or subassemblies and their yield from disassembly. In addition, on a disassembly line, not only can the demand arrive at the last workstation, it can also arrive at any of the other workstations in the system. Moreover, end-of-life products can arrive at any workstation and can be in various forms or combinations requiring different disassembly sequences. It is crucial to consider many factors when choosing a proper production control mechanism for a disassembly line. Push system and pull system are two broad types of systems used to control the operations of an assembly line. However, disassembly lines tend to be highly inventory intensive and thus it makes the pull system a viable alternative here as well. Kanbans are often used in a pull system to control inventory levels and can be easily implemented on an assembly line. However, designing a kanban mechanism for a disassembly line has not been thoroughly studied in the past. In this paper, we discuss the complications encountered in a disassembly line and suggest a proper way to control it using the pull system concept. We introduce a multi-kanban system and its routing mechanism in order to direct kanbans in an efficient way. We demonstrate the effectiveness of the proposed methodology by comparing it to the push system using a simulation model and a numerical example.

DISASSEMBLY LINE

Disassembly line can be described as a series of workstations operating in a sequence to disassemble the end-of-life (EOL) products to meet certain demand for subassemblies and/or components. In a disassembly line, the arriving products may consist of different combinations of components from a given

* Corresponding author

set of components. From a set of N components, the total number of possible combinations of components, $Q_{(N)}$ is given by:

$$Q_{(N)} = 2^N - N - 1 \quad (1)$$

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, that is, the number of combinations increases exponentially with the increase in the number of components. The variety of EOL products entering the system destabilizes 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.

The arrival pattern of demand in a disassembly line is much more complicated than in a typical assembly line. The key reason is the multilevel arrival pattern of demand. Demand can occur at any level of the disassembly process. In most assembly lines, demand arrives only at the last workstation. In that case, even if multilevel arrival of demand were considered, its effect would be relatively 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 one in the end 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. All this creates chaos in the system. The problem becomes even more critical when products cannot be stored for long periods of time due to the market conditions, space limitations, or need for controlled environments. Even though mechanisms such as kanban, base stock and CONWIP have been used to control inventory in an assembly line setting, they have not been deployed in a disassembly line setting. This is true because these mechanisms are not suited for the disassembly environment in their current forms. This, therefore, leads to the need for developing better production control systems in order to efficiently work in a disassembly line setting.

In general, there are two types of control mechanisms: *push mechanism* and *pull mechanism*. The push mechanism relies on a predetermined production schedule based on the expected demand of finished products. On the other hand, the production in pull mechanism is triggered by the actual demand and causes a flow of materials throughout the system. The push system has advantages in terms of experience in implementing it and providing higher levels of customer service in certain production scenarios because the system tends to build up inventory. On the other hand, the pull mechanism has an advantage that it does not generate large amounts of inventory. However, it relies heavily on consistency of raw materials supplies and agility of the server. This is the main reason why pull mechanism is more likely to perform better than push mechanism in a disassembly line. Kanban [7] is one of the most commonly used pull mechanism tools available. Advantages of the kanban system include its ability to control production, restriction of system's inventory, its simplicity in production scheduling, reduced burden on operators, ease of identification of parts by the kanbans attached to them, and substantial reduction in paper work. Traditionally, kanbans are designed for a deterministic environment. Its performance is, therefore, optimum in that environment. However, once implemented in a disassembly line setting, the kanban system is fraught with numerous types of uncertainties [5], [6]. A modification of the mechanism is therefore needed to improve its performance by reducing the impact of these uncertainties [9].

MULTI-KANBAN MODEL

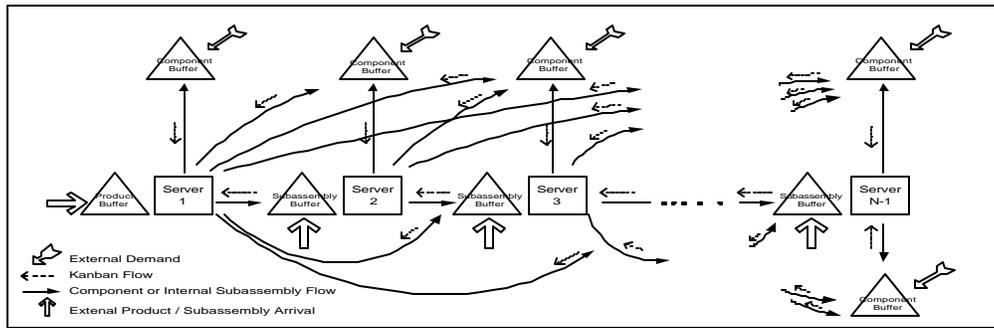


FIGURE 1. Kanbans and Materials Flows in a Disassembly Line

Kanban Routing Mechanism

Two basic types of materials in the system are *components* and *subassemblies*. A component is a single item that cannot be further disassembled and is waiting to be retrieved via a customer demand. On the other hand, a subassembly is something that can still be disassembled.

Consider workstation j , where $1 \leq j \leq N-1$. 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 j 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 $1 \leq i < j$, 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 $1 \leq i \leq (j-1)$, 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 i , then workstation i 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 i 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.

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 (ij_x). 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 (ij) from the product buffer with no other options because the only subassembly that results in immediate separation of component j is the subassembly (ij). 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_x) 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.

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:

$$k_i = \max(1, R_i / F_i) \quad (2)$$

$$k_j^* = \max(1, R_j^* / F_j^*) \quad (3)$$

where R_i 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:

$$R_i = d_i, \text{ for } 1 \leq i \leq N \quad (4)$$

$$F_i = \sum_{w=1}^i s_{(i,w)}, \text{ for } 1 \leq i \leq N \quad (5)$$

$$R_j^* = s_i, \text{ } i \text{ is the next component to be disassembled in the sequence} \quad (6)$$

$$F_j^* = a_j^* + \sum_{w=1}^{m-1} s_{(i,w)}, \text{ } i \text{ is the latest component disassembled in the sequence} \quad (7)$$

Where d_i 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.

EXPERIMENT, RESULTS AND DISCUSSION

We created a simulation model using the PC version of SIMAN and ARENA [8]. Run time was set to 48 hours after 5 hours of warm-up period for each replication. The steady state was reached and confirmed using Welch Procedure [10]. We collected data for three performance measures, viz., the number of satisfied demand (SD), the average work in process (WIP), and the average order completion time (OCT), for the traditional push system (TPS) and the multi-kanban system (MKS). In order to determine the central tendency and variability in the performance measures, we performed 20 independent replications. The example considered is a disassembly line consisting of 4 workstations. Eight different products (ABCDE, ACDE, ABC, AC, BCDE, BC, CDE and DE) are to be disassembled in order to fulfill the demands for five components. Note that we allow any of the subassemblies to arrive from external sources at the appropriate point on the disassembly line. The kanban level is calculated using equations (2) and (3). We ran two sets of experiments representing the push system and the multi-kanban system. In the push system, all arriving products were processed continuously under the FCFS rule. Table 1 presents a summary of average values of the three performance measures with their standard deviations shown in parentheses. The difference in WIP is statistically significant at 0.05 level. The variability in OCT is significantly reduced because the MKS allows the disassembly process to start only when there is a demand for a component. This should be helpful when predicting the projected order completion time. The difference in average inventory for each component (WIP) for the TPS and MKS is statistically significant from each other because the MKS routes kanban to the station that always results in the most needed component. This creates the pulling of materials exactly where and when they are needed. Results confirm that the MKS is capable of meeting customer demand that is comparable to the TPS. Figure 2 shows that the MKS has lower average inventory of all five components than the TPS. The TPS builds up inventory in order to fulfill customers' demands. On the other hand, the MKS system deals with fluctuation among demands by routing the kanbans to the most suitable workstation after determining that there is no overflow part available. By examining the number of parts being requested in real time and the number of available source products, the MKS selects the best destination for the kanban. In this case, the system was able to reduce the inventory level by an average of 81% while fulfilling customers' demands using the suggested kanban levels in the system.

TABLE 1. The Performance of TPS and MKS

	A		B		C		D		E	
	TPS	MKS	TPS	MKS	TPS	MKS	TPS	MKS	TPS	MKS
DS	471 (7.3)	474 (8.7)	376 (6.7)	379 (6.3)	638 (5.8)	634 (7.7)	474 (9.7)	472 (8.9)	475 (6.8)	473 (7.8)
WIP	88 (5.6)	16 (2.3)	65 (4.9)	9 (1.4)	129 (6.9)	24 (3.9)	159 (7.1)	35 (3.2)	152 (7.3)	34 (3.3)
OCT	26.4 (2.6)	26.2 (2.1)	24.9 (5.3)	21.6 (3.7)	38.1 (5.7)	37.8 (3.3)	45 (8.3)	44.7 (4.2)	46.3 (8.4)	45.9 (4.7)

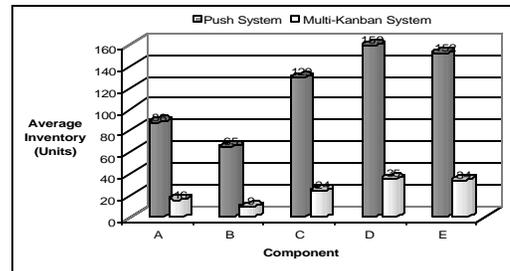


FIGURE 2. WIP level for TPS and MKS

REFERENCES

- [1] Brennan L., Gupta S. M. and Taleb K. N., "Operations Planning Issues in an Assembly/Disassembly Environment", *International Journal of Operations and Production Management*, Vol. 14, No. 9, 57-67, 1994.
- [2] Gungor A. and Gupta S. M., "A Solution Approach to the Disassembly Line Balancing Problem in the Presence of Task Failures", *International Journal of Production Research*, Vol. 39, No. 7, 1427-1467, 2001.
- [3] Gungor A. and Gupta S. M., "Disassembly Line in Product Recovery", *International Journal of Production Research*, Vol. 40, No. 11, 2569-2589, 2002.
- [4] Gungor A. and Gupta S. M., "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey", *Computer and Industrial Engineering*, Vol. 36, No. 4, 811-853, 1999.
- [5] Gupta S. M. and Al-Turki Y. A. Y., "The Effect of Sudden Material Handling System Breakdown on the Performance of a JIT System", *International Journal of Production Research*, Vol. 36, No. 7, 1935-1960, 1998.
- [6] Gupta S. M., Al-Turki Y. A. Y. and Perry R. F., "Flexible Kanban System", *International Journal of Operations and Production Management*, Vol. 19, No. 10, 1065-1093, 1999.
- [7] Hopp W. J. and Spearman M. L., "Factory Physics", Second Edition, McGraw-Hill, New York, 2001.
- [8] Kelton D. W., Sadowski R. P. and Sadowski, D. A., "Simulation with Arena®", WCB, McGraw-Hill, New York, 1998.
- [9] Korugan, A. and Gupta, S. M., "Adaptive Kanban Control Mechanism for a Single Stage Hybrid System", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing II*, Newton, Massachusetts, October 28-29, pp. 175-182, 2001.
- [10] Law, A. M. and Kelton, D.W., "Simulation Modeling and Analysis", McGraw-Hill, New York, 1991.
- [11] Lambert A. J. D., "Disassembly Sequencing: A Survey", *International Journal of Production Research*, Vol. 41, No. 16, 3721-3759, 2003.