



CHALMERS

Chalmers Publication Library

A Universal Framework for Lean Design and Control of Automated Material Handling Systems

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

IEEE 17th International Conference on Emerging Technologies and Factory Automation, ETFA 2012

Citation for the published paper:

Mashaei, M. ; Lennartson, B. (2012) "A Universal Framework for Lean Design and Control of Automated Material Handling Systems". IEEE 17th International Conference on Emerging Technologies and Factory Automation, ETFA 2012

<http://dx.doi.org/10.1109/ETFA.2012.6489603>

Downloaded from: <http://publications.lib.chalmers.se/publication/171011>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

A Universal Framework for Lean Design and Control of Automated Material Handling Systems

Maziar Mashaei^{1,2}, Bengt Lennartson¹

¹Department of Signals and Systems, Chalmers University of Technology, Sweden

²FlexLink AB, Göteborg Sweden

Abstract—Lean design and control of an automated material handling system is investigated in this study. A universal framework for modeling and analysis of different types of material handling mechanisms is introduced to obtain a minimum number of resources in a system design and fulfill a desired throughput. This framework is developed in a discrete event simulation environment and applied to a case study based on a real pallet system technology. The minimal design of the pallet system is realized by devising the system universal model.

Key words: Automated material handling system, Lean design, Universal model, Discrete event simulation model.

I. INTRODUCTION

An automated material handling system (AMHS) is integrated in a modern production environment to transport, buffer, and locate various parts for processing machines. Different types of technologies have been considered for an AMHS. Pallet and Conveyor systems, Automated Guided Vehicles (AGVs), Cranes, and Handling robots are major examples of these technologies.

High quality of products, fulfillment of a desired throughput, and cost minimization are the main objectives for design and control of a manufacturing plant. Defining waste as a resource or effort that does not transform a part (does not add any value to a part), Lean design philosophy [1] sums up these objectives with the concept of waste minimization. According to this philosophy, a material handling system is a waste but necessary (auxiliary system), because it does not transform a part, but it supports adding value activities by reaching the part to operation machines. It is clear that a complete elimination of an AMHS is not possible (unless it is assumed that all operations are carried out in one processing machine). However, a lean design to provide the right amount of the right material, in the right condition, at the right place, at the right time, in the right position, in the right sequence, for the right cost, and using the right methods is a vision for AMHS designers, [2].

In practice, large percentages of operational cost (15% to 70%), factory space (55%), and production time (87%) of a manufacturing sector are attributed to its material handling system, according to [2] and [3]. This introduces a vast gap between the visionary design and practical one. Numerous design complexity issues in the integrated AMHS and the lack of a systematic approach to deal with these complexities are speculated for a big part of this gap.

To discuss this, a graph model of an AMHS is presented. In this model, locating, connecting, and intersection positions are represented by nodes, and flow paths are specified by edges. For example, Fig. 1 shows a complete graph for 6 nodes. Regarding the graph model, the structure design

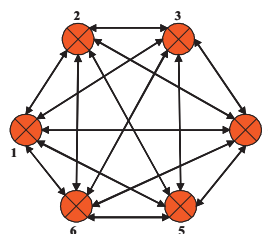


Fig. 1. A complete graph model of an AMHS, including 6 nodes

procedure should specify number and configuration of nodes, technology and equipment type for each node, number and configuration of edges, and handling mechanism for each edge. On the other hand, the control design procedure may characterize the number of parts in the system (work-in-process), the direction and velocity of parts in each flow path, the buffer level and queuing policy in each edge, scheduling and time instances for releasing parts in each node, and distributing parts to the edges connected to a node.

These design procedures should meet the vision described earlier. For a given system (known number of nodes and edges, system configuration, and technology), the control design procedure can efficiently be conducted to enhance the system performance [4], [5], [6], [7], minimize the makespan [8], [9], [10], and reduce the cycle time and work-in-process (WIP) [11], [12], [13], [14], [15] for given AMHSs.

However, in the scope of the lean philosophy, the structure and control of an automated material handling system must be concurrently designed to estimate the right type and number of resources (nodes and edges), while fulfilling the operational issues. As a matter of fact, optimal values of the system control parameters should be set such that the desired production rate is obtained with the minimum number of AMHS resources.

This introduces a big challenge because there is no framework that includes all design variables (structure and control parameters) in one model. In a classical approach, the structure and resources of an AMHS are first specified, sacrificing the precise modeling of the system dynamic. Rough approximated explicit formula, [1], and optimization models, [2], [16], are the main methods for the structure design. Then for the given structure, control set points are calculated as previously mentioned. This approach does not guarantee an AMHS lean design, because the impact of control parameters on the structure design is not fully and precisely investigated.

To tackle the above challenge and cope with the lean

design paradigm, a universal modeling framework for an AMHS that captures both structure and control design variables is addressed. Furthermore, the modeling concept is implemented in CPN-tool, a discrete event simulation environment. A case study is introduced, and its universal CPN-model is developed and analyzed according to the lean paradigm.

II. STRATEGIES FOR THE LEAN STRUCTURE AND CONTROL DESIGN OF AN AMHS

In this section, we explain the basis of the lean structure and control design for an AMHS with devising a simple example. Fig. 2 illustrates a set of processing machines operating on parts A and B in a manufacturing plant. Having specifications about the plant layout, sequence and time of operations, part features (types, shapes, weights, and etc), and parts production rate, the goal is to design an automated material handling system for the plant that meets the specifications with the minimum cost and number of used resources.

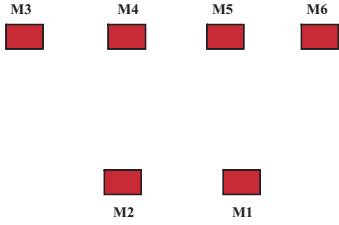


Fig. 2. Manufacturing plant with 6 processing machines

Now, consider the operation sequences for parts A and B as follows:

- A: $O_1(M^1) \rightarrow O_2(M^2) \rightarrow O_3(M^3) \rightarrow O_4(M^4) \rightarrow O_5(M^5) \rightarrow O_6(M^6) \rightarrow O_7(M^3) \rightarrow O_7(M^6)$
- B: $O_1(M^1) \rightarrow O_2(M^2) \rightarrow O_3(M^3) \rightarrow O_4(M^5) \rightarrow O_5(M^6)$

A flow path design of the AMHS, which captures these sequences, is shown in Fig 3. The big nodes indicate the locating points, at which the parts are positioned for operating actions, and the small ones represent the intersections between the paths. This design determines 10 nodes and 13 edges. In practice, these nodes and edges imply resources such as fixture, robot, conveyor, aisle, diverter, merger, etc.

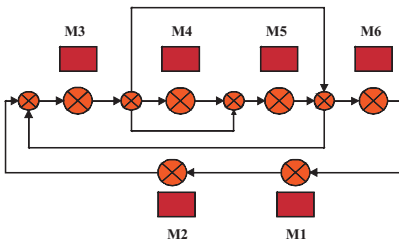


Fig. 3. Manufacturing plant with an AMHS

Another design of the AMHS with the simplest configuration form can be realized by the closed loop depicted in Fig 4. Although this design has the lowest number of

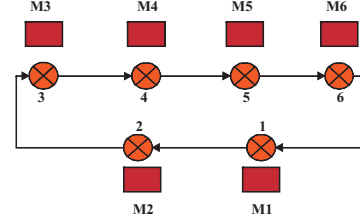


Fig. 4. Manufacturing plant with one closed loop AMHS

configuration resources, it can not be applied if the desired production rate is not fulfilled. The major problem in this design is that the flow time of parts A and B (especially A) in the plant can be significantly increased due to the rise of unnecessary transportation and waiting actions. As a result, the average throughput TH decreases according to Hopp and Spearman equation

$$TH = \frac{WIP}{FT} \quad (1)$$

which relates TH to the average flow time FT and work-in-process WIP , [17]. One solution for this problem is to increase WIP in order to maintain the desired FT . But, this solution can not always be practised because a part carrier could be the most expensive resource in the AMHS (for example AGV), or there could be a barrier for increasing WIP (limited buffer spaces).

Although the closed loop form may not give the optimal configuration (satisfy the desired production rate), it can still be devised as a base for the analysis leading to the lean design. For the rest of this section, we try to explain this analysis for our example.

For the closed loop configuration, parts A and B pass the nodes as bellows.

- A: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$
- B: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$

This means that part A circulates two times, and part B circulates one time in the loop. In the steady state behavior, it can be assumed that three parts are in the loop, A1 for the first circulation of A, A2 for the second circulation of A, and B for itself. Mapping part numbers $I = \{1, 2, 3\}$ respectively to the part set $\Pi = \{A1, A2, B\}$, the operation action of the part $i \in I$ in the node $\ell \in \{1, \dots, 6\}$ can be specified by the operation time measure τ_i^ℓ . If there is an operation, $\tau_i^\ell > 0$; otherwise, $\tau_i^\ell = 0$. For instance, regarding part A2, we have $\{\tau_2^1 = 0, \tau_2^2 = 0, \tau_2^3 > 0, \tau_2^4 = 0, \tau_2^5 = 0, \tau_2^6 > 0\}$.

For the one-loop configuration, let us denote the edge ℓ and its downstream node ℓ as the segment ℓ . Similar to the operation time τ_i^ℓ , we introduce transportation time $T_{r_i}^\ell$ and waiting time W_i^ℓ for the part i in the segment ℓ . Therefore, the flow time of part i is this segment is

$$FT_i^\ell = T_{r_i}^\ell + \tau_i^\ell + W_i^\ell \quad (2)$$

and the relation

$$TH = \frac{WIP}{\frac{1}{3} \sum_{i=1}^3 \sum_{\ell=1}^6 (T_{r_i}^\ell + \tau_i^\ell + W_i^\ell)} \quad (3)$$

is obtained for the one loop design according to (1). The operation times are fixed, but the transportation and waiting

times can theoretically be reduced to maintain the desired production rate for the plant, regarding (2) and (3). The velocity of part i in segment ℓ , v_i^ℓ , and the length of the segment for this part, L_i^ℓ , control $T_{r_i}^\ell$. Besides, the schedule of the part set Π for releasing actions from the node ℓ , I^ℓ , and the bypassing behavior of part i for this node, and the buffer size of the related segment control the waiting time W_i^ℓ .

Assume that after theoretical analysis, we obtain $T_{r_2}^\ell = 0$ for each segment ℓ . In theory, this leads to $v_2^\ell \rightarrow \infty$ or $L_2^\ell = 0$. It is obvious that these values are not reasonable for the practice. However, they give important clues to the AMHS designer to reduce the transportation times of A2 as much as possible. One option may be realized by introducing a feedback as shown in Fig 3. The other option may be to decrease W_2^ℓ through changing the part set schedules, while $T_{r_2}^\ell$ is tuned in a practical range.

For instance, assume that we can achieve the desired throughput with reasonable transportation times for the parts, provided that the schedule $[2, 1, 3]^T$ is set for all nodes except node 3 which demands the schedule $[1, 2, 3]^T$. This means that for all nodes except node 3, the parts A2, A1, and B are sequentially released, in order, and for node 3 the releasing sequence is reversed for parts A2 and A1. Such a schedule change may be applied in the real plant, using tunnel modules for pallet systems, bypassing actions for AGVs, and robots.

How to translate theoretical parameters into practical resources depends on many factors including handling mechanism, state of the art in the AMHS design, the experience of designers, and the cost of resources. However, a lean design can conceptually be formalized in one closed loop framework, by considering a minimum number of changes in the part set schedules and minimal bypassing actions and WIP, while part velocities and segment lengths are tuned in practical ranges. The core of this framework is a dynamic control over parameters influencing the throughput, using the conceptual lean design formalism. Universal model of an AMHS, which is introduced in the following section, characterizes such a framework.

III. UNIVERSAL MODEL OF AN AMHS

A practical handling mechanism has limitations for transportation time, queuing policy, and buffering and locating parts, regarding its technology. For example, a handling equipment may not be faster than 1 m/s, follows First Input First Output (FIFO) queuing policy, and only buffers two parts for locating actions. If these limitations are dropped, then a one-loop universal configuration can be defined to model any type of automated material handling system technologies. In this section, we introduce the important parameters realizing the analysis based on this model.

A. Configurable parameters in a universal segment

Carrier phases for transportation, queuing, and locating actions in a universal segment are controlled by the segment configurable parameters. Table I itemizes these parameters and their descriptions.

A data list which contains values of all configurable parameters is assigned to each part code. From the practical point of view, a part code can be written in a Radio

TABLE I
CONFIGURABLE PARAMETERS OF A UNIVERSAL SEGMENT

Parameter	Description
Part code	Integer number assigned to a part instance
Segment length	Distance between the entrance and exit of the segment
Carrier velocity	Speed of the carrier in the segment
Setup time	Duration for locating a carrier in the segment
Operation time	Duration for processing of a part on a carrier
Part type	Type of a part on a carrier
Operation type	Loading, Unloading, Assembling, Processing, etc
Resources	Type and number of shared resources for an operation
Part priority	Order which prioritizes a part on a carrier for the locating action, compared to other part instances
Buffer size	Maximum capacity for storage of carriers in the segment
Carriers number	Initial number of empty carriers in the segment buffer
Part set number	Number of part instances

Frequency IDentification (RFID) tag and attached to a carrier. When a carrier reaches an RFID tag reader/writer, the tag is read and sent to a control system. The data list related to this tag is fetched by the system, and proper commands according to these data are activated. From the model point of view, in a similar way, the data list is configured for each segment. When a carrier arrives at a segment, its part code is specified. Then proper decisions and actions are taken in the segment to govern the carrier through the phases described here.

1) *Carrier transportation phase*: A carrier standing at the exit of the universal segment $\ell - 1$, $S(L - 1)$, enters to the universal segment ℓ if the segment buffer is not full. After the entrance, the part code of the carrier is added to the segment ℓ buffer list and eliminated from the buffer list of the segment $\ell - 1$. The carrier is in the transportation phase until it reaches the queuing position in the segment. Provided that there is a tunneling permission for the part instance, it bypasses the locating position and arrives at the exit of the segment ℓ , $S(L)$. Otherwise, the carrier is queued in the buffer, $B(L)$. This sequence of actions for the carrier is illustrated in Fig. 5.

2) *Carrier queuing phase*: The carrier, which has passed the transportation phase and not bypassed the locating position, stays at $B(L)$ until the locating position is set free (see Fig. 6). The segment data list contains the order of part instances for locating actions. The lower number in the list has the higher priority for the action. Therefore, if the part instance on the carrier has the highest priority among the other part instances, the carrier enters to the position, $M(L)$, for the locating action.

3) *Carrier locating phase*: The carrier with the highest priority is located after a setup time, provided that there is a free space in the locating position. When the locating action has been finished, the resources, which are required for the operation and shared by other operations, are fetched from the resource available list. In the case that the demanded resources are available in the list, the operation is started and these resources are excluded from the resource available list. Otherwise, the operation is postponed until the list indicates the presence of the resources. After the operation time, the used resources are released and included in the list of available resources. Furthermore, if there is another operation for the part on the carrier, the new part code is considered for that part. Accordingly, the sequence of actions is repeated as shown in Fig. 7. Completing all operations, the carrier is released from the locating position and enters to the exit state

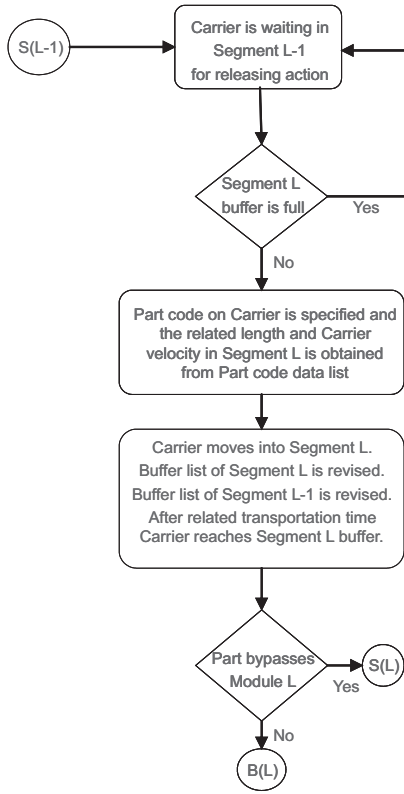


Fig. 5. Transportation phase of a carrier

of the segment ℓ .

IV. UNIVERSAL SEGMENT MODEL IN CPN TOOL

CPN Tool is a discrete event modeling environment based on Colored Petri Nets [18]. CPN supports hierarchical and modular modeling, and also facilitates a model verification, using simulation, state space analysis, and a global reference time.

In this section, CPN tool is employed to characterize the technical and dynamic aspects of the carrier phases in a universal segment. Fig. 8 demonstrates the model details and the related activities for the phases of a carrier.

The color token (N, j, k, p, q) represents a carrier, which holds information about its number N , part code j , cycle number k , part type p , and operation q . When the carrier enters to a segment, the token (N, j, k, p, q) appears at Segment Entrance. This token reaches Segment Buffer after the duration calculated by Transportation transition. The token stays at the buffer until the condition on Entrance to Module is fulfilled. Then this transition transfers the token to Segment Module where it is either directed to Segment Locating Position by Locating transition, or buffered until Releasing Action is initiated.

In the case that the locating action happens, if Resource Available List contains all demanded resources for the part code, the operation is started. Otherwise, the operation is halted until these resources are released and appeared in Resource Available List. After the operation accomplishment, if Segment Module is empty, the token is transferred to this place by Unlocating transition. Finally, the token will appear in Segment Exit (and Segment Entrance of the consecutive

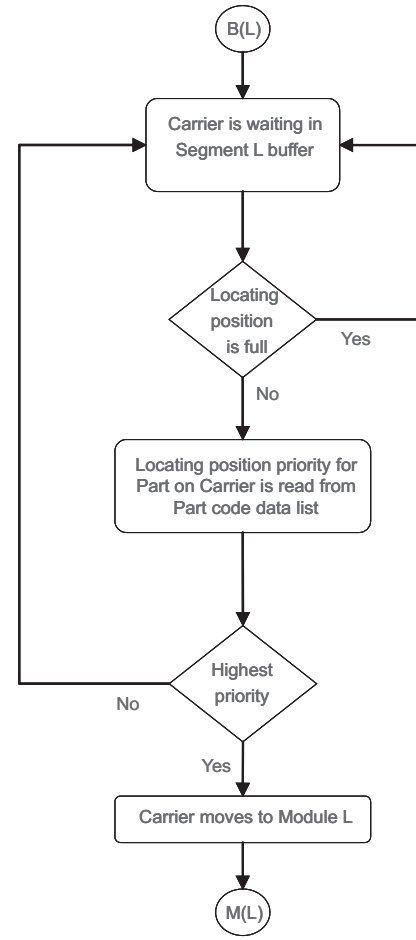


Fig. 6. Queuing phase of a carrier

segment) when Releasing Action transition is activated. Other transitions and places in the model are used to control the token play according to the rules and actions in the phase charts.

The details in the model can be hidden from a user view, devising the hierarchical and modular features of CPN Tool. Fig. 9 depicts a universal segment in a modular form in the system level. The configurable parameters in the figure specify the characteristics of the segment regarding a part instance. These parameters should be tuned according to the following rules.

- Each part instance passing the segment has a design vector, which its elements are values of the configurable parameters.
- Sequence of actions in the segment for the carrier (N, j, k, p, q) are specified regarding the design vector having $j_1 = j$.
- If $j_1 = j_2$, there is only one operation for the carrier (N, j, k, p, q) in the locating position. On the other hand, when the first operation is finished according to j_1 , the value of j_2 is parameterized in j . Hence, a new part code and part instance are considered on this carrier.
- $i = 0$ indicates that the part instance j_1 should bypass the locating position. Furthermore, a part instance has the highest priority for the locating action in cycle k if

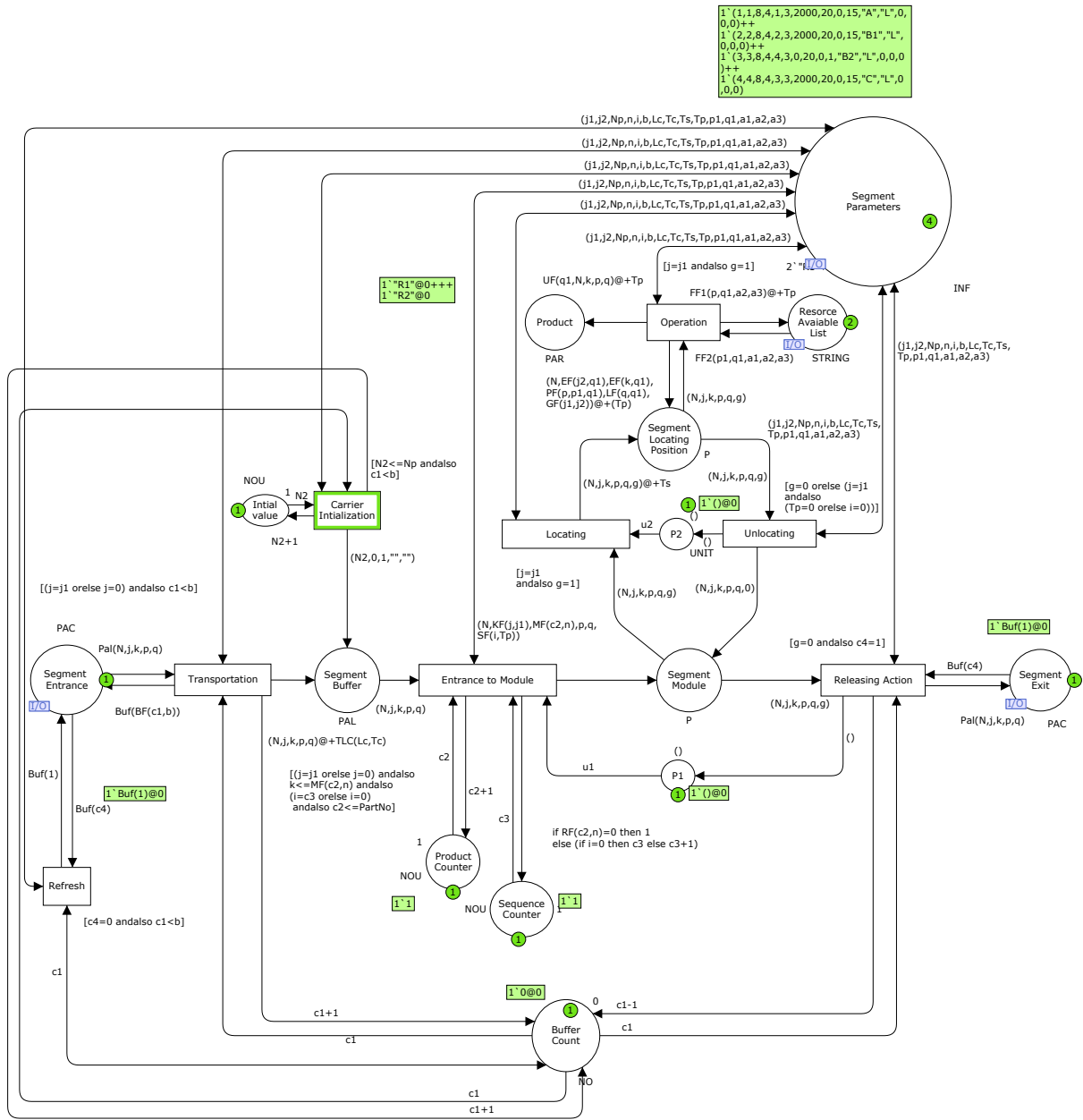


Fig. 8. CPN model of a universal segment

the value of i is the smallest among the part instances, which have initialized for the segment but not located in this cycle.

- T_c is the time for the carrier transporting the part instance ($j = j_1$) 1 meter. Accordingly, $L_c T_c$ gives the transportation time of this carrier in the segment.
- The value of q_1 determines the operation type for the part instance $j = j_1$ in the segment. The options for this parameter are:
 - 1) $q_1 = L$ indicates Loading action; the value of p_1 is loaded on the empty carrier (N, j, k, p, q) , ($p = ""$). This means that $p_1 \rightarrow p$.
 - 2) $q_1 = U$ indicates Unloading action; the value of p in the carrier (N, j, k, p, q) is unloaded ($"" \rightarrow p$) and appeared in Resource Available List location.

- 3) $q_1 = As$ indicates assembly operation; the value of p_1 is concatenated to the value of p in (N, j, k, p, q) , ($p p_1 \rightarrow p$).
- 4) Except mentioned values, another value of q_1 indicates a processing operation (for example, $q_1 = W$ denotes a welding operation). The carrier (N, j, k, p, q) keeps the value of p as before for a processing operation.

- a_1, a_2, \dots, a_r respectively indicates the number of share resources R_1, R_2, \dots, R_r , demanded for the operation q_1 on the part instance $j = j_1$.

A user can connect a number of modular segments to make a universal model of an AMHS in the system level and configure the related parameters to evaluate the system. In the system model, each segment has its own Segment Parameter

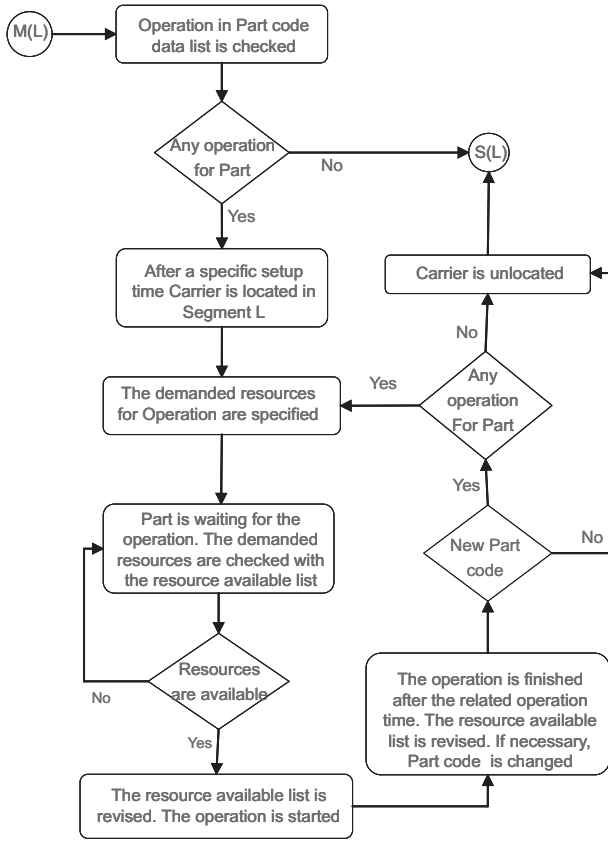


Fig. 7. Locating phase of a carrier

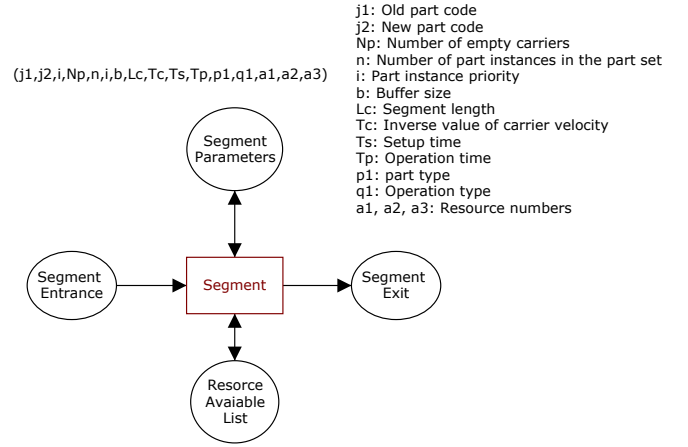


Fig. 9. Modular form a universal segment

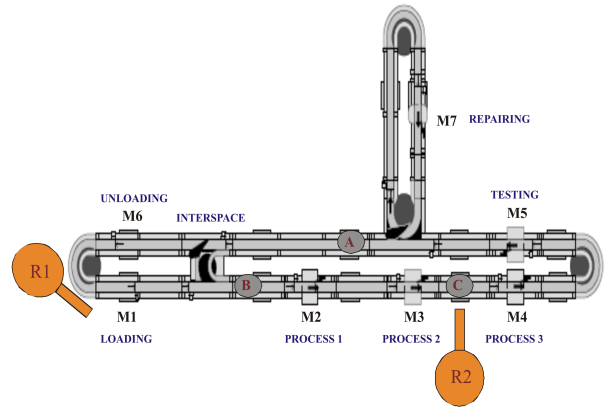


Fig. 10. X85 HPLS technology for tree part types A, B, and C

place, but Resource Available List is unique for all segments.

V. CASE STUDY: LEAN DESIGN AND CONTROL OF X85

X85 is one type of the pallet systems, which are designed and constructed by FlexLink company, [19]. As demonstrated in Fig. 10, a design of this system is used to handle three part types A, B, and C for the related operations.

In the case that there is no defect, the sequence of operations ($O_s(M^l) : s \in \mathbb{N}$) for each part type is as follows:

- A : $O_1(M^1) \rightarrow O_2(M^2) \rightarrow O_3(M^4) \rightarrow O_4(M^5) \rightarrow O_5(M^6)$
- B : $O_1(M^1) \rightarrow O_2(M^3) \rightarrow O_3(M^4) \rightarrow O_4(M^2) \rightarrow O_5(M^3) \rightarrow O_6(M^5) \rightarrow O_7(M^6)$
- C : $O_1(M^1) \rightarrow O_2(M^4) \rightarrow O_3(M^5) \rightarrow O_4(M^6)$

Based on the operation sequences, part B has a reentrance behavior with bypassing the loading and unloading modules through the interspace module. Besides, robot R1 is a shared resource between the loading and unloading operations, and robot R2 is shared between $O_5(M^3)$ for B and $O_2(M^4)$ for C. The design specifications are:

- 1) Each part type should be produced with the production rate 1/42 parts per second.
- 2) The minimum number of resources should be used in the design of the pallet system.

To properly tune the configurable parameters to fulfil these specifications, the following design steps are considered.

A. Step 1

The system design data are fetched and presented in Table II. The data for part type B have been separated into data

for the first entrance (B1) and data for the second entrance (B2), assuming that B1 and B2 are two individual part types in the steady state behavior of the system.

TABLE II
X85 PALLET SYSTEM DESIGN DATA

M^l	τ_A [s]	τ_{B1} [s]	τ_{B2} [s]	τ_C [s]	L_A^l [m]	L_{B1}^l [m]	L_{B2}^l [m]	L_C^l [m]	b^l
M^1	15	15	0	15	2	2	0	2	3
M^2	60	0	40	0	3	3	1.5	3	4
M^3	0	75	35	0	1.5	1.5	1.5	1.5	2
M^4	50	45	0	30	1.5	1.5	1.5	1.5	2
M^5	20	0	25	20	3	3	3	3	4
M^6	10	0	10	10	6	4	6	6	6

B. Step 2

The universal model of the pallet system in CPN tool is developed (see Fig. 11). Resource Available List is a common place between all universal segments. In other words, each segment has access over this place and can change its value (token). Moreover, SP1,...,SP7 are places for the segment parameters.

Each of these places contains a number of tokens representing design vectors. A design vector includes $(j_1, j_2, N_p, n, i, b, L_c, T_c, T_s, T_p, p_1, q_1, a_1, a_2, a_3)$ ele-

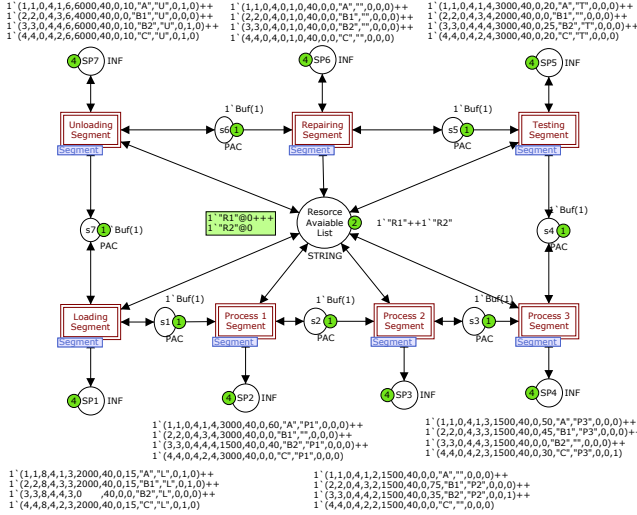


Fig. 11. The universal model of the case study

ments, which are defined according to Table I. Most of these elements are parameterized, using the given system data. As an example, the design vector related to the part type A and the loading segment has the elements $(1, 1, N_p, 4, i, 3, 2000, T_c, 0, 15, "A", "L", 0, 1, 0)$. Except the design variables N_p , T_c , and i , the other elements are parameterized as follows:

- $j_1 = 1$ denotes that the part code is 1 for the part instance.
- $j_2 = 1$ indicates that there is only one operation for the part instance.
- $n = 4$ denotes that the part set has four part instances.
- $b = 3$ denotes that the buffer size of the segment is 3.
- $L_c = 2000$ denotes that the length of the segment is 2000 mm for this part.
- $T_s = 0$ indicates that the pallet setup time is neglected or included in the operation time.
- $T_p = 15$ denotes that the part instance operation time is 15 s.
- $p_1 = "A"$ denotes that the part type is A .
- $q_1 = "L"$ indicates that the part A should be loaded on an empty pallet.
- $a_1 = 0$ indicates that no work piece should be assembled to the part.
- $a_2 = 1$ denotes that Robot 1 should be used for the operation (loading action).
- $a_3 = 0$ denotes that Robot 2 is not used for the operation.

C. Step 3

The resources which are used in a X85 design are described in Table III. The design policy is to minimize these resources and provide 1/42 parts per second.

An X85 segment contains a conveyor line, a Stop module, and a Basic module. If the system is based on one closed loop, one electrical motor and a conveyor chain can be shared for all segments, and there is no need for directing modules including Diverter, Merger, Combiner, and Interspace. Thus, adding only one segment to the loop significantly increases

TABLE III
X85 PALLET RESOURCES

Resource	Function Description
Pallet	carries a part in the pallet system
Conveyor chain	transports a pallet from one machine to another
Electrical motor	runs a conveyor chain
Basic Locating	locates a pallet for a process and blocks another pallet flow
Tunnel Locating	Locates a pallet for a process with the option that other pallets can pass through the module
Stop	stops a pallet for buffering or loading/unloading actions
Diverter	directs pallets from one main line to a number of lines
Merger	merges pallets from two branch lines into one main line
Combiner	merges and diverts pallets from two branch lines into one main line and vice versa
Interspace	bypasses the route of a pallet by diverting and merging

the number of resources. To have one electrical motor, the conveyor chain in the segments should have the same velocity. In addition, to eliminate the directing modules and Tunnel modules in the design, a part should have the same priority in all segments, without a bypass behavior.

Of course, this minimal design policy is acceptable if the desired production rate (1/42 parts per second) is satisfied. For the case that the production rate in the minimal design is lower than 1/42, the values of the configurable parameters T_c and i in segments are modified to keep the desired level.

Based on the minimal design policy, a brute force search for the optimal values of N_p , T_c , and i are conducted by devising the CPN simulation model. This leads to $N_p^1 = 12$, $N_p^{2-6} = 0$, $T_{ci}^l = 4$, and

$$I \begin{pmatrix} A \\ B1 \\ B2 \\ C \end{pmatrix}^{2-5} = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 2 \end{pmatrix}, \quad I \begin{pmatrix} A \\ B1 \\ B2 \\ C \end{pmatrix}^{1,6} = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 3 \end{pmatrix} \quad (4)$$

The schedule releasing vectors in (4) show that part types A and $B2$ have the releasing orders 1 and 4 respectively for all segments. Furthermore, the orders of $B1$ and C are changed between Loading and Process 1 segments as well as Testing and Unloading segments. Therefore, a mechanism, which can switch the locating orders of $B1$ and C in the mentioned segments, is needed. From X85 technology point of view, the switch of the orders between $M1$ and $M2$ can be performed by using a Tunnel module placed behind $M2$. Part C can bypass part $B1$ located by the Tunnel module.

On the other hand, for the switch of the orders between the testing module, $M5$, and the unloading module, $M6$, an Interspace module can be devised. Before C arrives at $M6$, $B1$ bypasses the unloading and loading modules through the Interspace. This module holds $B1$ until the new part C , which has been released from $M1$, arrives at $M2$. Then, $B1$, which is now changed to $B2$, is released from the Interspace module. The Tunnel and Interspace modules are added to the pallet system loop in order to provide the desired production rate.

Before the number of pallets (WIP) is minimized for the pallet system, the current design should be examined and justified to give a chance for the elimination of the added resources. The next step concerns about the design justification and WIP minimization.

D. Step 4

Throughput of a production plant may be enhanced by increasing the buffer size of the bottleneck machine. For the given system, the machine of Process 3, $M4$, is the bottleneck, and its buffer size is increased to 3. The design procedure mentioned in Step 3 is repeated for the modified system, and the configurable parameters are justified as bellows.

$$N_p^1 = 16, N_p^{2-6} = 0, T_{c_i}^\ell = 4, I \left(\begin{array}{c} A \\ B1 \\ B2 \\ C \end{array} \right)^{1-6} = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 4 \end{pmatrix} \quad (5)$$

This implies that if $b^4 = 3$, the Interspace and Tunnel modules can be removed, while the desired throughput is still satisfied (in practice the interspace module is useful because it increases the flexibility of the system). Furthermore, the minimal WIP is achieved for the new settings

$$N_p^1 = 7, N_p^{2-6} = 0, T_{c_i}^\ell = 2.8, I \left(\begin{array}{c} A \\ B1 \\ B2 \\ C \end{array} \right)^{1-6} = \begin{pmatrix} 1 \\ 4 \\ 2 \\ 3 \end{pmatrix} \quad (6)$$

According to these steps, the minimal design of the pallet system to give 1/42 parts per second is obtained for 6 number of segments and 7 number of pallets if $M4$ buffer size is incremented by 1. In the case that there is no space for the buffer size enlargement, a Tunnel module and an Interspace module should be added to the system design.

VI. CONCLUSION AND FUTURE WORKS

In this paper, a universal framework for modeling and analysis of an automated material handling system was introduced. The main objectives were to obtain a minimal number of resources in the system design and fulfill a desired production rate. This framework was developed in a discrete event simulation environment and applied to a case study based on a real pallet system technology. Using the universal model, a lean design for the pallet system was realized.

A lean design of an automated material handling system demands a brute force search over domains of design parameters in the system universal model. This consumes a lot of time for big systems. One way to reduce the design time is to estimate the initial value of the design parameters in the simulation, using an optimization model of the universal framework. The combination of the universal optimization model and discrete event simulation model can be considered as a continuation of this work.

REFERENCES

- [1] M.P. Groover, "Automation, production systems, and computer-integrated manufacturing", Second Edition, Prentice Hall International, 2001.
- [2] J.A. Tompkins, J.A. White, Y.A. Bozer, and J.M.A. Tanchoco, "Facilities Planning", Third Edition, John Wiley & Sons, NJ, 2003.
- [3] E.H Frazelle, "Material handling: A technology for industrial competitiveness", Material Handling Research Center Technical Report, Georgia Institute of Technology, Atlanta, 1986.
- [4] S. Biller, S.P. Marin, S.M. Meerkov, and L. Zhang, "Closed Bernoulli production Lines: Analysis, Continuous Improvement, and Leanness", IEEE Transactions on Automation Science and Engineering, Vol. 6, No. 1, pp. 168-180, 2009.

- [5] M.S Han, D.J Park" Performance analysis and optimization of cyclic production line", IIE Transactions, Vol 34, pp. 411-422, 2002.
- [6] Y. Frein, C. Commaut, and Y. Dallery, "Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers", IIE Transactions, Vol 28, pp. 545-554, 1996.
- [7] B. Pourbabai, " Performance modeling of a closed loop material handling system", European Journal of Operational Research , Vol 32, pp 340-352, 1987.
- [8] S.P. Sethi, C. Sriskandarajah, S.L. van de Velde, M.Y. Wang, and H. Hoogeveen, "Minimizing makespan in a pallet-constrained flowshop", Journal of Scheduling, Vol. 2, pp. 115-133, 1999.
- [9] M.Y. Wang, S.P. Sethi, C. Sriskandarajah, and S.L. van de Velde, "Minimizing makespan in a flowshop with pallet requirements", Inform. Systems Oper. Res. INFOR, nr. 35, pp. 277-285, 1997.
- [10] C. Chu, J.M. Proth, and S.P Sethi, "Heuristic procedures for minimizing makespan and the number of required pallets ", European Journal of Operational Research, Vol. 86, pp. 491-502, 1995.
- [11] O. Korbaa, H. Camus, and J.C Gentina "A new cyclic scheduling algorithm for flexible manufacturing systems", The International Journal of Flexible Manufacturing Systems, Vol 14, pp. 173-187, 2002.
- [12] F. Chauvet, J.W. Herrmann, and J.M. Proth, Optimization of cyclic production systems: a heuristic approach, IEEE Transactions on Robotics and Automation, Vol 19, No. 1, pp. 150-154, 2003.
- [13] T. Hsu, O. Korbaa b, R. Dupas, and G. Goncalves, "Cyclic scheduling for F.M.S.: Modelling and evolutionary solving approach", European Journal of Operational Research, Vol. 191, pp. 464-484, 2008.
- [14] M. Mashaei, B. Lennartson, F. Sannehed, and G. Abbestam, "Optimal design of a decoupled multiple-loop pallet system for flexible cyclic manufacturing plants", CASE, IEEE Conference on Automation Science and Engineering, pp. 433-438, 2010.
- [15] P. Solot, "A heuristic method to determine the number of pallets in a flexible manufacturing system with several pallet types", The International Journal of Flexible Manufacturing Systems, Vol.2, pp. 191-216, 1990.
- [16] J.M.A. Tanchoco, "material flow systems in manufacturing", Chapman and Hall, London, 1994.
- [17] W.J. Hopp, and M.L. Spearman, "Factory physics: Foundations of manufacturing management", 2nd edition, IrwinMcGraw Hill, NY, 2000.
- [18] K. Jensen, "Coloured Petri nets: A high level language for system design and analysis", Advances in Petri Nets 1990G. Rozenberg, Editor, Lecture Notes in Computer Science Vol. 2483, Springer-Verlag, pp. 343-416, 1991.
- [19] <http://www.flexlink.com>.