The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Optimal scheduling of automatic guided vehicle transportation system based on MLD system modeling

Kenji Sawada[1] and Seiichi Shin[1]

[1]The University of Electro-Communications, Japan
(Tel: 81-42-443-5253, Fax: 81-42-443-5253)

[1]{knj.sawada, seiich.shin}@uec.ac.jp

Abstract: This paper proposes an optimal scheduling method of transportation systems in semiconductor manufacturing within MLD (mixed logical dynamical) modeling framework. We consider an optimal scheduling problem of AGV (automatic guided vehicle) system transfer problem, which is to control the AGV congestion around the meeting points and the dividing points of the transportation road in this paper. The problem is recast as an ILP (Integer Linear Programming) problem within model predictive control framework.

Keywords: automatic guided vehicle, MLD modeling, scheduling

1 INTRODUCTION

Recently, there is a growing need to increase productivity in semiconductor manufacturing with the improvement of the producing technology. In semiconductor fabrication (FAB) in **Fig. 1**, a few hundred of Automatic Guided Vehicles (AGVs) transpose Front Open Unified Pods (FOUPs) that store semiconductor wafers [1].

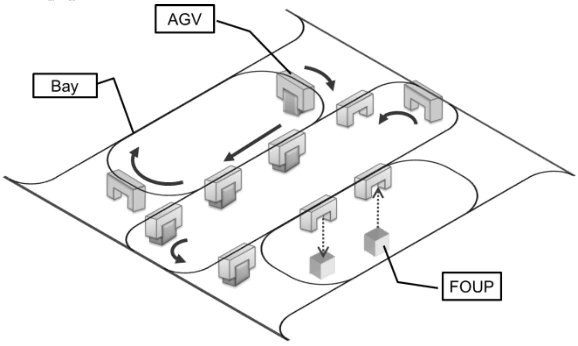

Fig. 1. Semiconductor fabrication

Nowadays, the size of the wafers will be getting lager (from 300mm to 450mm) and the number of the vehicles in a factory will be increasing. This implies that each of the AGVs must follow the vehicle in front of one closely and transpose heavy wafers within a set time. In particular, it is important to consider the AGV congestion on the transportation roads which consist of inter-bays and intra-bays since the congestion results in the production delay. The congestion tends to occur around the junctions between the bays and/or the transfer equipment of the FOUPs. In particular, this paper focuses on an optimal scheduling problem of the meeting points and the dividing points of the FAB junctions, while the existing results [2-4] tend to consider scheduling problems that focus on all areas of FAB.

The typical scheduling process on the junctions is as follows: (i) The prioritized AGV which goes through the meeting point without stopping is automatically determined by the distance from the meeting point. (ii) When some of the other AGVs around the meeting point come close to colliding with the prioritized one, they continuously slow down or stop according to their collision avoidance systems. Such a scheduling procedure does not previously consider both the positional relation and the number of the vehicles which may collide, and/or the influence of the AGVs' stopping and slowing down on other AGVs except the meeting point area. These factors can lead to the AGV congestion as shown in **Fig. 2**. Also, it is important to consider the behaviors of the AGVs around the dividing point area in front of the meeting point area. If we know in advance that some AGVs go to the parking area through the dividing point, such the information makes it easier to maximize the number of the AGVs come out from the meeting point area.

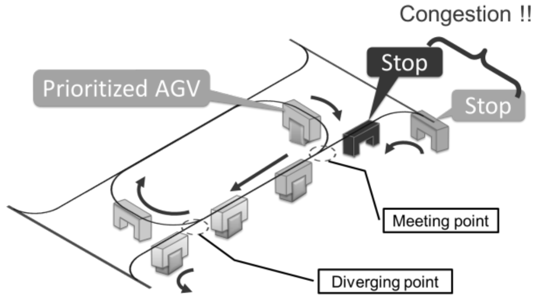

Fig. 2. Congestion on junction

Motivated by the above, our early work considers the AGV congestion problem of the meeting point based on the state space realization modeling [5] and the model predictive control framework. As an extension of our existing result [5], this paper proposes a mixed logical dynamical (MLD) modeling [6-8] of the meeting and the dividing points considering the positional relation (the area of the bay model), the number of the vehicles, and the time of the AGV congestion. By using the model, in addition, this paper considers the optimal scheduling problem considering the AGV congestion within model predictive control framework. In this case, the problem is recast as an Integer Programming.

Notation: The set of $n \times m$ real (integer) matrices is denoted by $\mathbf{R}^{n\times m}$ ($\mathbf{N}^{n\times m}$). The set of $n \times n$ diagonal matrices with the diagonals being 1 or 0 is denoted by $\mathbf{D}^n$. $O_{n\times m}$, I_n, $\mathbf{0}_m$ and $\mathbf{1}_m$ (or for simplicity of notation, O, I, $\mathbf{0}$ and $\mathbf{1}$) denote the $n \times m$ zero matrix, the $n \times n$ identity matrix, the $n \times 1$ vector whose all elements are zero and the $n \times 1$ vector whose all elements are one, respectively. For a matrix M, M^{T} denotes its transpose. For a vector x, x_i is the i-th entry of x. $\mathrm{diag}(D_1,\dots,D_n)$ denotes the block diagonal matrix of matrices $D_1,\dots,D_n$. $\{0,1\}^n$ denotes the set of n-dimensional vectors, which consists of elements 0 and 1.

2 PROBLEM FORMUILATION

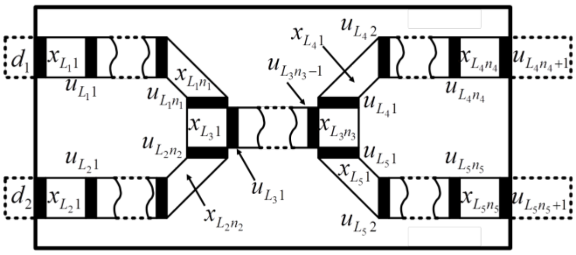

Fig. 3. Bay model of meeting and dividing points

Fig. 3 shows a bay model of the meeting and dividing points. The area surround by the black border is the area of the bay model that we focus on. The bay is divided into several parts x_{L_ij} by the gates u_{L_ij} and d_i similar to our earlier work. Each AGV behavior is expressed by the gate opening/closing. In particular, the gate d_i implies the behavior of the AGVs that come in from the outside.

We define the notations. The bay model consists of 5 links L_i $(i = 1,\dots,5)$ as shown in **Fig. 3**. $x_{L_ij}(t)$ indicates the number of the AGVs on the corresponding j-th area of the link L_i at the t-th time. For example, if one AGV moves on the 4-th area of the link L_2 at the 2-nd time, $x_{L_24}(2) = 1$ holds. $u_{L_ij}(t) = 1$ or 0 indicates the corresponding j-th gate opening or closing of the link L_i at the t-th time, respectively. For the exogenous gate $d_i(t) \in \{0,1\}$, $d_i(t) = 1$ indicates that one AGV comes in through the corresponding i-th exogenous gate at the t-th time. Denote by $x_{L_i}(t) \in \mathbf{R}^{n_i}$, $u_{L_i}(t) \in \mathbf{R}^{m_i}$ and $d(t) \in \mathbf{R}^2$ the vectors of which the i-th entries are $x_{L_ij}(t)$, $u_{L_ij}(t)$ and $d_i(t)$, respectively.

According to the above notations, the bay model of the meeting and dividing points can be expressed by the state space realization. First, the link L_1 and L_2 are given by

$$x_{L_i}(t+1) = A_{L_i}x_{L_i}(t) + B_{1L_i}u_{L_i}(t) + B_{2L_i}\,d_i(t)$$

where $i = 1,2$ and the matrices are given by

$$A_{L_i} := I_{n_i},\ B_{1L_i} := \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 \\ 1 & -1 & 0 & \ddots & \vdots \\ 0 & 1 & -1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \in \mathbf{R}^{n_i\times n_i},$$

$$B_{2L_i} := [1 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}} \in \mathbf{R}^{n_i}.$$

Second, the link L_3 is given by

$$x_{L_3}(t+1) = A_{L_3}x_{L_3}(t) + \sum_{i=1}^{2} B_{13L_i}u_{L_i}(t) + B_{1L_3}u_{L_3}(t) + \sum_{i=4}^{5} B_{13L_i}u_{L_i}(t)$$

where the matrices are given by

$$A_{L_3} := I_{n_3},\ B_{1L_3} := \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 1 & -1 & \ddots & \vdots \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbf{R}^{n_3\times(n_3-1)},$$

$$B_{13L_i} := \begin{cases} \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 \end{bmatrix} \in \mathbf{R}^{n_3\times n_i}, & i = 1,2, \\ \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ -1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbf{R}^{n_3\times(n_i+1)}, & i = 4,5. \end{cases}$$

Finally, the link L_4 and L_5 are given by

$$x_{L_i}(t+1) = A_{L_i}x_{L_i}(t) + B_{1L_i}u_{L_i}(t)$$

where $i = 4,5$ and the matrices are given by

$$A_{L_i} := I_{n_i},\ B_{1L_i} := \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \in \mathbf{R}^{n_i\times(n_i+1)}.$$

By defining the vectors as follows:

$$x(t) := \left[x_{L_1}(t)^{\mathrm{T}}\, x_{L_2}(t)^{\mathrm{T}}\, x_{L_3}(t)^{\mathrm{T}}\, x_{L_4}(t)^{\mathrm{T}}\, x_{L_5}(t)^{\mathrm{T}}\right]^{\mathrm{T}},$$

$$u(t) := \left[u_{L_1}(t)^{\mathrm{T}}\, u_{L_2}(t)^{\mathrm{T}}\, u_{L_3}(t)^{\mathrm{T}}\, u_4(t)^{\mathrm{T}}\, u_{L_5}(t)^{\mathrm{T}}\right]^{\mathrm{T}},$$

we obtain the state space realization of the bay model:

$$x(t+1) = Ax(t) + B_1u(t) + B_2d(t) \tag{1}$$

where the matrices are given by $n \equiv \sum_{i=1}^{5} n_i$, $A := I_n$,

$$B_1 := \begin{bmatrix} B_{1L_1} & O & O & O & O \\ O & B_{1L_2} & O & O & O \\ B_{13L_1} & B_{13L_2} & B_{1L_3} & B_{13L_4} & B_{13L_5} \\ O & O & O & B_{1L_4} & O \\ O & O & O & O & B_{1L_5} \end{bmatrix} \in \mathbf{R}^{n\times(n+1)},$$

$$B_2 := \begin{bmatrix} B_{2L_1}^{\mathrm{T}} & O & O \\ O & B_{2L_2}^{\mathrm{T}} & O \end{bmatrix}^{\mathrm{T}} \in \mathbf{R}^{n\times 2}.$$

3 SCHEDULING ALGORITHM

3.1 Modeling of AGV scheduling

By using the bay model (4), this paper considers the scheduling problem **(I)**: *Suppose that the area of the bay model, the number of the vehicles, and the time of the AGV congestion are given. Formulate a scheduling algorithm (i) maximizing the number of the AGVs come out from the area and (ii) avoiding the AGV collision.*

Denote by M the time of the AGV congestion (the horizon length). To achieve the property (i), first, we consider the following cost function:

$$\max_{u(0),\dots,u(M-1)} J \ \text{ s.t. } J := q_M^{\mathrm{T}} x(M) + \sum_{t=0}^{M-1} \left(q^{\mathrm{T}} x(t) + r^{\mathrm{T}} u(t)\right),$$

$$q_{\mathrm{M}} \in \mathbf{R}^n,\ q \in \mathbf{R}^n \ \text{and}\ r \in \mathbf{R}^{n+1}. \quad (2)$$

We consider the "throughput maximization" of the links 4 and 5. In this case, the throughputs of the links 4 and 5 are the opening and closing times of $u_{L_4 n_4+1}(t)$ and $u_{L_5 n_5+1}(t)$, the weight vector r is given by

$$r := \left[0 \cdots 0\ q_{L_4}^{\mathrm{T}}\ q_{L_5}^{\mathrm{T}}\right]^{\mathrm{T}},\quad q_{L_i} := [0 \cdots 0\ \ 1]^{\mathrm{T}} \in \mathbf{R}^{n_i+1}.$$

We denote by $\lambda_i^{d_4} \in \mathbf{R}$ or $\lambda_i^{d_5} \in \mathbf{R}$ the distance between x_i $(i \in \{1, n\})$ and $x_{L_4 n_4}$ or $x_{L_5 n_5}$, respectively. In this case, the sum of the distances from $x_{L_4 n_4}$ or $x_{L_5 n_5}$ to any vehicles in the bay model at the *t*-th time (we denote it $d_s(t)$) is expressed by

$$d_s(t) = \sum_{j}^{2} \Lambda_j^{\mathrm{T}} x(t),\quad \Lambda_j := \left[\lambda_1^{d_j} \cdots \lambda_n^{d_j}\right]^{\mathrm{T}} \in \mathbf{R}^n.$$

If $\sum_{t=0}^{M} d_s(t)$ is minimized, the throughput within the time $t \in [0, M]$ of the bay also may increase. Then, we set

$$q = q_M = -(\Lambda_1 + \Lambda_2).$$

Also, we consider the throughput difference $\theta \in \mathbf{R}$ between the links 4 and 5, which is expressed by

$$-\theta \le g_\theta^{\mathrm{T}} u(t) \le \theta, \quad (3)$$

$$g_\theta := \left[\mathbf{0}^{\mathrm{T}} \quad q_{L_4}^{\mathrm{T}} \quad -q_{L_5}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbf{R}^{n+1}.$$

To achieve the property (ii), next we consider the following rules.

(a) Capacity of each area: For the areas $x_{L_i j}(t)$ $(i = 1,\dots,5, j = n_1,\dots,n_5)$, each capacity is set to be up to one AGV. This rule is expressed by the constraint:

$$\mathbf{0} \le x(t) \le \mathbf{1}. \quad (4)$$

(b) Meeting and dividing points: To avoid the collision at the meeting of two gates $u_{L_1 n_1}(t)$ and $u_{L_2 n_2}(t)$, we consider the rule that does not allow the two gates to open at the same time. Also, we consider the rule that does not allow the two gates $u_{L_4 1}(t)$and $u_{L_5 1}(t)$ of the dividing point to open at the same time. These rules are given by the constraint:

$$\mathbf{0} \le G_b u(t) \le \mathbf{1}, \quad (5)$$

$$G_b := \begin{bmatrix} g_{bL_1}^{\mathrm{T}} & g_{bL_2}^{\mathrm{T}} & O & O & O \\ O & O & O & g_{bL_4}^{\mathrm{T}} & g_{bL_5}^{\mathrm{T}} \end{bmatrix} \in \mathbf{R}^{2\times(n+1)},$$

$$g_{bL_i} := [0 \ \ \cdots \ \ 0 \ \ 1]^{\mathrm{T}} \in \mathbf{R}^{n_i},\ \ i = 1,2,$$

$$g_{bL_i} := [1 \ \ 0 \ \ \cdots \ \ 0]^{\mathrm{T}} \in \mathbf{R}^{n_i+1},\ \ i = 4,5.$$

(c) Speed condition: The acceleration of AGV around the meeting point can lead to the collision. Then we consider the logic:

$$u_i(t) = 1 \to x_i(t) = 1,\ \ i \in \left\{1,\dots,\sum_{i=1}^{3} n_i - 1\right\},$$

$$u_{L_4 1}(t) + u_{L_5 1}(t) = 1 \to x_{L_3 n_3}(t) = 1,$$

$$u_{L_i j+1}(t) = 1 \to x_{L_i j}(t) = 1, i \in \{1,\dots,n_i\}, j \in \{4,5\}.$$

This logic indicates that an AGV does not go through more than two gates on the corresponding areas at a time. By using the lemma in [6-8], the above logic can be expressed by the inequality:

$$G_{cu} u(t) \le G_{cx} x(t), \quad (6)$$

$$G_{cu} := \begin{bmatrix} G_{cL} & \mathbf{0} & O & \mathbf{0} & O \\ O & 1 & O & 1 & O \\ O & \mathbf{0} & G_{cL_4} & \mathbf{0} & O \\ O & \mathbf{0} & O & \mathbf{0} & G_{2cL_5} \end{bmatrix},$$

$$G_{cL} := \mathrm{diag}\left(G_{cL_1}, G_{cL_2}, G_{cuL_3}\right),\ \ G_{cuL_3} \in \mathbf{D}^{n_3-1},$$

$$G_{cx} := \mathrm{diag}\left(G_{cL_1}, G_{cL_2}, G_{cxL_3}, G_{cL_4}, G_{cL_5}\right),$$

$$G_{cL_i} \in \mathbf{D}^{n_i},\ \ i \in \{1,2,4,5\},\ \ G_{cxL_3} \in \mathbf{D}^{n_3}.$$

If G_{cL_i}, G_{cuL_3}, G_{cxL_3} are set to be the identity matrix, the all areas does not allow the AGVs to speed up.

Therefore, the bay model with the control rules **(a)**-**(c)** can be expressed by the pair of the linear state space system (1) and the linear inequalities (3)-(6). We consider the initial time t=0. In this case, the problem (I) is recast as the following problem **(II)**: *Suppose that the initial position* $x(0) := x_0$ *and the sequence* $d(0), d(1), \dots, d(M-1)$ *of the AGVs which come in are given. Find* $u(0), u(1), \dots, u(M-1)$ *maximizing the cost function (2) for the bay model with the control rules realized by*

$$\begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2d(t) \\ Cx(t) + Du(t) \le E \end{cases}, \tag{7}$$

$$C := \begin{bmatrix} 0 \\ 0 \\ I \\ -I \\ \mathbf{0} \\ \mathbf{0} \\ -G_{cx} \end{bmatrix},\ D := \begin{bmatrix} g_\theta^{\mathrm{T}} \\ -g_\theta^{\mathrm{T}} \\ O \\ O \\ G_b \\ -G_b \\ G_{cu} \end{bmatrix},\ F := \begin{bmatrix} \theta \\ \theta \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ O \end{bmatrix}.$$

3.2 ILP based scheduling

In this subsection, we reduce the problem (II) to an ILP problem by using the model predictive control technique in [6] similar to our early work [5]. For the time M, we define vectors as follows:

$$x_M(t) := [x(t)^{\mathrm{T}}\ x(t+1)^{\mathrm{T}} \cdots\ x(t+M)^{\mathrm{T}}]^{\mathrm{T}},$$

$$u_M(t) := [u(t)^{\mathrm{T}}\ u(t+1)^{\mathrm{T}} \cdots\ u(t+M-1)^{\mathrm{T}}]^{\mathrm{T}},$$

$$d_M(t) := [d(t)^{\mathrm{T}}\ d(t+1)^{\mathrm{T}} \cdots\ d(t+M-1)^{\mathrm{T}}]^{\mathrm{T}}.$$

In this case, the cost function can be recast as

$$J = Q_M x_M(t) + R_M u_M(t) \tag{8}$$

and $x_M(t)$ is given by

$$x_M(t) = A_M x(t) + B_{M_1} u_M(t) + B_{M_2} d_M(t)$$

where the matrices are defined by

$$A_M := \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^M \end{bmatrix},\ B_{M_i} := \begin{bmatrix} O & O & \cdots & O \\ B_i & O & \cdots & O \\ AB_i & B_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ A^{M-1}B_i & \cdots & AB_i & B_i \end{bmatrix},$$

$$Q_M := [q^{\mathrm{T}}\ \ \cdots\ \ q^{\mathrm{T}}\ \ q_M^{\mathrm{T}}],\ R_M := [r^{\mathrm{T}}\ \ \cdots\ \ r^{\mathrm{T}}].$$

Also, the inequality of (7) can be also rewritten as

$$C_M x_M(t) + D_M u_M(t) \le E_M \tag{9}$$

where the matrices are defined by

$$C_M := [\mathrm{diag}(C, \dots, C)\ \ O],\ D_M := \mathrm{diag}(D, \dots, D),$$

$$E_M := [E^{\mathrm{T}}\ \ \cdots\ \ E^{\mathrm{T}}]^{\mathrm{T}}.$$

By using (8) and (9), we see that the following theorem holds of the problem **(II).**

Theorem 1: *The problem* **(II)** *is equivalent to the following optimization problem* **(III)***:*

given x_0 and $d_{M_{0_M}}$

find $u_{M_0} \in \{0,1\}^{nM}$

max $Q_M B_{M_1} u_{M_0} + Q_M\left(A_M x_0 + B_{M_2} d_{M_0}\right)$

subject to $\left(C_M B_{M_1} + D_M\right) u_{M_0}$
$\le E_M - C_M(A_M x_0 + B_{M_2} d_{M\_}0)$

where the vectors are defined by

$$u_{M_0} := [u(0)^{\mathrm{T}}\ u(1)^{\mathrm{T}} \cdots\ u(M-1)^{\mathrm{T}}]^{\mathrm{T}},$$

$$d_{M_0} := [d(0)^{\mathrm{T}}\ d(1)^{\mathrm{T}} \cdots\ d(M-1)^{\mathrm{T}}]^{\mathrm{T}}.$$

This problem can be solved by the appropriate solvers such as Optimization toolbox of MATLAB in [9] and/or ILOG CPLEX in [10] because of ILP framework. Numerical examples will be shown in the conference room.

4 CONCLUSION

This paper has proposed a dynamic modeling of transportation systems in semiconductor manufacturing. Utilizing our modeling method, we have considered an optimal scheduling problem focusing on the AGV congestion at the meeting point of the transportation road junctions. As a result, the problem can be recast as an ILP problem within model predictive control framework.

REFERENCES

[1] K. Gartland (1999), Automated Material handling system (AMHS) framework use requirements documen t version 1.0, International SEMATECH, Technol. Trans fer #99073793A-TR

[2] G. K. Agrawal and S. Heragu (2006), A survey of automated material handling systems in 300-mm se miconductor fabs, IEEE trans. Semiconductor Manufac turing, 19(1), pp.112-119

[3] F. Kato and S. Shin (2010), Multistep optimal sc heduling of automated guided vehicles in a semiconduc tor fabrication, Proc. of SICE Annual Conference in Ta iwan, pp.985-989

[4] T. Nishi and R. Maeno (2010), Petri Net Decom position Approach to Optimization of Route Planning P roblems for AGV Systems, IEEE Trans. Aut. Sci. Eng., 7(3), pp.523-537

[5] K. Sawada, S. Shin, K. Kumagai and H. Yoneda (2011), Dynamic modeling for optimal scheduling of automatic guided vehicle system, Proc. of SICE Annua l Conference, pp.580-583

[6] A. Bemporad and M. Morari (1999), Control of systems integrating logic, dynamics, and constraints, Au tomatica, 35(3), pp.407-427

[7] T. M. Cavalier, P. M. Pardalos, and A. L. Soyst er (1990), Modeling and integer programming techniqu es applied to propositional calculus, Computer & Opera tions Research, 17(2), pp.561-570

[8] H. P. Williams (1999), Model building in mathe matical programming, 4th ed., John Willey&Sons. Ltd

[9] http://www.mathworks.com/

[10] http://www-01.ibm.com/software/integration/optimi zation/cplex-optimizer/