89034661

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※　多自由度人工義肢肌電控制器之研究　　　　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑個別型計畫　□整合型計畫

計畫編號：NSC89－2212－E－002－066－

執行期間：88年8月1日至89年7月31日

計畫主持人：黃漢邦

共同主持人：

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份

□赴大陸地區出差或研習心得報告一份

☑出席國際學術會議心得報告及發表之論文各一份

□國際合作研究計畫國外研究報告書一份

執行單位：國立台灣大學機械工程學系

中　華　民　國　89　年　9　月　29　日

行政院國家科學委員會專題研究計畫年度報告

多自由度人工義肢肌電控制器之研發

(The Development of a Myoelectric Controller for a Multi-Degree Prosthetic Hand)

計畫編號：NSC 89-2212-E-002-066
執行期限：88年8月1日至89年7月31日
主持人：黃漢邦　　研究人員：蔡柏修
執行機構：國立台灣大學機械工程學系
Email: hanpang@ccms.ntu.edu.tw

中文摘要

本研究的主要目的是發展一個肌電信號（EMG）辨識器並用來控制一多自由度的機械手，而成爲一個多手指的人工義手系統。此 DSP-based 肌電信號辨識器主要是用來辨識出手部的八種動作，它們分別是：power grasp、hook grasp、wrist flexion、lateral pinch、flattened hand、centralized grip、three-jaw chuck 與 cylindrical grasp。當抓握的姿態決定了之後，多手指的機械手就可以被用來執行這項抓握的任務。

爲了要控制 NTU-Hand III 這個多手指的機械手，本文發展出一個 PC-based 機械手控制卡。此外，一個 OpenGL 的 3D 圖形使用者介面也被開發來監控 NTU-Hand III。最後，爲了提升此人工義手系統的操控性，乃發展主僕式架構與力的彈簧模型並以模擬和實驗來驗證此系統的可行性。

關鍵詞：DSP、肌電信號辨識器、多手指人工義手、NTU-Hand III、OpenGL、主僕式架構、力的彈簧模型

Abstract

The major objective of this research is to develop an EMG discriminator and use it to control a multi-degree robot hand so that a prosthetic system can be constructed. The DSP-based EMG discriminator is used to discriminate the eight hand motions, which are power grasp, hook grasp, wrist flexion, lateral pinch, flattened hand, centralized grip, three-jaw chuck and cylindrical grasp. After the prehensile posture is obtained, the multi-fingered robot hand is controlled to perform the grasp.

A PC-based hand control card is developed to control the multi-fingered robot hand (NTU-Hand III). An OpenGL 3D graphic user interface is also developed to control and monitor the status of the NTU-Hand III. Finally, the master/slave structure and the spring force model are applied to improve the manipulability of the prosthetic hand. Simulations and experiments are performed to justify the integrated prosthetic system.

Keywords: DSP, EMG discriminator,

multi-degree robot hand, NTU-Hand III, OpenGL, master/slave structure, spring force model.

計畫緣由及目的

The human hand has more than 25 degrees of freedom (DOFs) and can affect both an adroit manipulation and a powerful grasp [8]. The human hand is the best helper of people in daily life. However, some people lose their hands due to workshop accidents, traffic accidents and diseases. In order to improve the life quality, many of them tend to use an artificial limb. But the most popular commercial prosthetic hands today are either the cosmetic hand or the body driven gripper. The former does not have any DOFs, the latter has only one DOF. Nowadays, more and more people concentrate their researches on prosthetic hand design [6, 9, 13]. The major requirements of modern prosthetic hands are summarized as follows:

a. Distinguish different prehensile postures required by the amputee.

b. Map postures into the prosthetic hand.

c. Design and control the prosthetic hand.

Since electromyography (EMG) is the most simple and direct way to represent the contracting information of a muscle, many researches use the EMG signals to discriminate prehensile postures and control the prosthesis [2, 3, 4, 12]. However, the human hand is a complex system with more than 25 DOFs and is driven by numerous of muscles. It is almost impossible to duplicate the complete functions of the human hand. A prosthetic hand should be able to map daily use postures into joint space and preshape the joint space position into a suitable and reachable posture to grasp an object [5]. Namely, the grasp planning and task planning of the prosthetic hand (or called dexterous hand) should be taken into account. The grasp planning involves in selecting a grasp posture for the dexterous hand and determining a trajectory so that the hand contacts and approaches the object. Task level planning simplifies the process of controlling the dexterous hand [10].

研究方法

(1) EMG Discriminative System

The EMG signal is easily gathered from skin via surface electrodes, but it is a nonstationary signal. In a small time period (0.3-0.4 s) contraction, however, they can be treated as a stationary signal. The amplitude of the signal ranges from 0 to 6 mV (peak-to-peak) or 0 to 1.5 mV (rms). Most of its power occurs in the frequency range of 5 to 500 Hz, while the dominant power spectrum density lies in the range of 50-150 Hz.

The entire EMG discriminative system is composed of four main parts: EMG signal collection, signal processing, feature extraction and classification. The general abstract model is shown in Fig. 1.1.

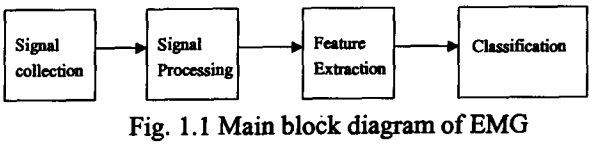

Fig. 1.1 Main block diagram of EMG discriminative system

Eight types of prehensile postures are selected from [11] for pattern recognition. Each of them is a typical posture in the daily

activity. The eight hand motions are shown in Fig. 1.2.

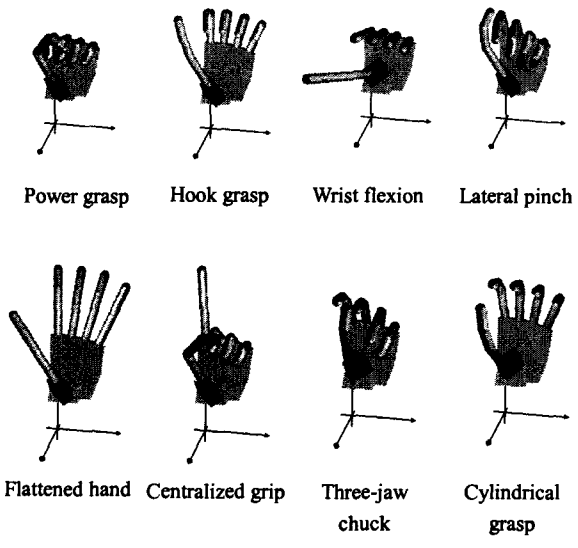

Fig. 1.2 Eight motions of the prosthetic hand

In order to choose the meaningful EMG signals for the above eight kinds of prehensile postures, the locations of electrodes are important. According to the relations between the muscle locations and the prehensile postures [7], three channel electrodes are placed on flexor pollicis longus, flexor digitorum superficialis and extensor carpi radialis brevis. This arrangement focuses on short below elbow disarticulation.

After amplifying the EMG signal about 312 times by the electrode and 20 times by the operation amplifier, the controller is then developed using DSP code. First, the average IEMG value of 25 ms raw window length is calculated to judge whether the muscle contracts or not. The threshold of the muscle contraction is added to compensate the surrounding environment noise. When the average IEMG value exceeds the defined threshold, 1000 running EMG data are collected within 04.ms. It can be illustrated by the following equation,

$$if \quad \overline{IEMG} = \frac{1}{N} \sum_{i=t}^{t+N} |X_i| > T_H$$

X_i = *collect EMG raw data for analyzing* $\quad i = 1$ *to* 1000

where N = *windows length* ; T_H : *threshold value*

After collecting 1000 points EMG data, the primary thing is to filter the uncertain signal. The bandwidth of the EMG signal is about 30-400 Hz and the environment noise is about 60 Hz. Thus, a bandpass filter with pass-band from 30Hz to 400Hz and a 60 Hz notch filter are designed. The Butterworth IIR digital filter is shown below.

30Hz to 400Hz bandpass filter

$$H_1(z) = \frac{Y_1(z)}{X_1(z)} = \frac{0.047946z^6 - 0.143838z^4 + 0.143838z^2 - 0.047946}{z^6 - 4.015503z^5 + 6.783575z^4 - 6.32227z^3 + 3.503237z^2 - 1.090804z + 0.141969}$$

60Hz notch filter

$$H_2(z) = \frac{Y_2(z)}{X_2(z)} = \frac{z^4 - 3.96z^3 + 5.9256z^2 - 3.970296z + 1.00520676}{z^4 - 3.92z^3 + 5.8045z^2 - 3.847284z + 0.963242}$$

The sampling periods are both 0.4 ms.

Since the bandpass and notch filter function are both linear systems, they can be combined to form a tenth-order filter that includes their characteristics. The augmented system transfer function can be written as

$$H(z) = \frac{Y(z)}{X(z)} = H_1(z) \times H_2(z)$$

The frequency response of $H(z)$ is shown in Fig. 1.3.

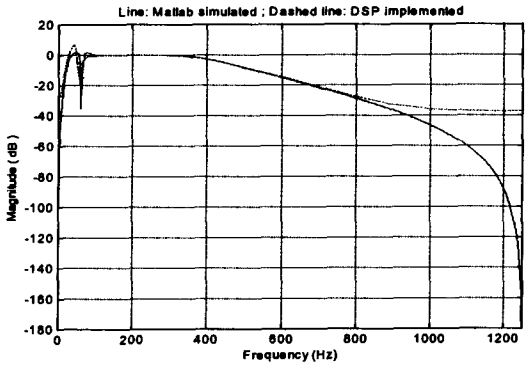

Fig. 1.3 Frequency response comparison of $H(z)$

In this project, several kinds of features will be used to represent the myoelectric signal patterns.

Integral of EMG (IEMG):

$$IEMG = \sum_{k=1}^{N} |x_k|$$

where x_k is the k-th sample data out of N samples raw data.

Waveform Length (WL):

$$WL = \sum_{K=1}^{N} |\Delta x_k|$$

Variance (VAR):

$$VAR = \frac{1}{N-1} \sum_{k=1}^{N} x_k^2$$

Zero Crossings (ZC):

$$ZC = \sum_{i=1}^{N} [\mathrm{sgn}(-x_k \times x_{k+1}) \ and \ |x_k - x_{k+1}| \geq 0.02]$$

$$\mathrm{sgn}(x) = \begin{cases} 1, & if \ x > 0 \\ 0, & otherwise \end{cases}$$

Slope Sign Changes:

The number of slope sign change increases, if

$$(x_k - x_{k-1}) \times (x_k - x_{k+1}) \geq 0.03 \quad for \ k = 1, \ldots, N$$

Willison Amplitude (WAMP):

$$WAMP = \sum_{i=1}^{N} f(|x_k - x_{k+1}|)$$

$$f(x) = \begin{cases} 1, & if \quad x > 0.3 \\ 0, & otherwise \end{cases}$$

Six parameters are extracted according to the above equations. But they are in sufficient to describe an EMG signal. Thus, 4 parameters of the AR model are used to improve the performance.

Autoregressive (AR) Model:

It is difficult to analyze the EMG signal due to its nonlinear and nonstationary nature. But in a short time the EMG signal can be regarded as a stationary Gaussian process. Graupe et al. [3] addressed a linear model for a Gaussian process. They used a pure autoregressive (AR) model to identify the EMG time series as

$$y_k = -\sum_{i=1}^{N} a_i y_{k-i} + \omega_k$$

where y_k is the EMG time series, and k is the interval. N is the order of AR model, a_i are the estimate of the AR parameters and ω_k is the white noise. A least squares method that minimizes the sum of squared ω_k is used to obtain the parameters of the signal model.

The above few parameters are selected to characterize the EMG signal features. Ten parameters for each channel signal and thirty parameters for each motion type will be obtained. These thirty parameters form the input vector and they are discriminated by a BP neural network to obtain one of the eight types hand motions.

(2) Control Philosophy of the Multi-Degree Prosthetic Hand

For a prosthetic hand, the magnitude of the grasping force directly affects the stability of manipulation. But the force should not be too large to damage the manipulated object. The force should be effectively limited in a range. Many control rules, such as computed torque method, are implemented by torque command and directly deal with the force. However, the control system of NTU-Hand III is position based rather than torque regulation. Consequently, position control rules would be more suitable. Hence, how to use the position control rule to achieve force control is very important. Since the NTU-Hand III is

integrated with tactile sensors, the simple spring model can realize the force control of the finger. Fig. 2.1 shows the representation of the spring force model.

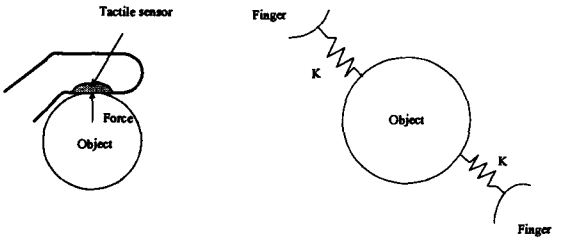

Fig. 2.1 The spring force model

Since the FSR (Force Sensing Resistor) used on NTU-Hand III is a single point type sensor, it can only measure the normal force exerted on the contact area. Namely, it cannot point out the actual location where the force is exerted in the contact area. Under this restriction, the model can be further simplified.

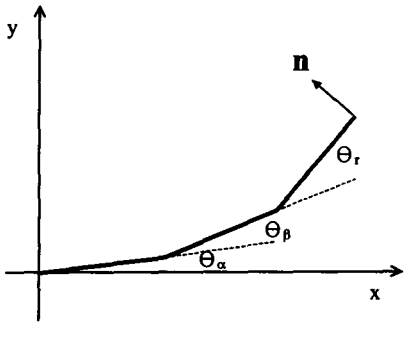

Fig. 2.2 The finger expressed in coordinate frame {0}

Fig. 2.2 shows the finger expressed in coordinate frame {0}. The unit normal vector **n** at the contact point is:

$$\mathbf{n}=\left[\cos(\theta_\alpha+\theta_\beta+\theta_\gamma+\frac{\pi}{2}) \quad \sin(\theta_\alpha+\theta_\beta+\theta_\gamma+\frac{\pi}{2})\right]^T$$

For simplicity, it is denoted as

$$\mathbf{n}=\left[-\sin(\theta_\alpha+\theta_\beta+\theta_\gamma) \quad \cos(\theta_\alpha+\theta_\beta+\theta_\gamma)\right]^T=\left[-S_{\alpha\beta\gamma} \quad C_{\alpha\beta\gamma}\right]^T$$

The position of the contact point can be expressed as a function of joint angles by the forward kinematics $\mathbf{X} = F(\mathbf{q})$

$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}\ell_\alpha C_\alpha+\ell_\beta C_{\alpha\beta}+\ell_\gamma C_{\alpha\beta\gamma}\\ \ell_\alpha S_\alpha+\ell_\beta S_{\alpha\beta}+\ell_\gamma S_{\alpha\beta\gamma}\end{bmatrix}$$

ℓ_n : length of link n, n = α, β, γ

C_n and S_n : $\cos\theta_n$ and $\sin\theta_n$, n = α, β, γ

$C_{\alpha\beta}$ and $S_{\alpha\beta}$: $\cos(\theta_\alpha+\theta_\beta)$ and $\sin(\theta_\alpha+\theta_\beta)$

$C_{\alpha\beta\gamma}$ and $S_{\alpha\beta\gamma}$: $\cos(\theta_\alpha+\theta_\beta+\theta_\gamma)$ and $\sin(\theta_\alpha+\theta_\beta+\theta_\gamma)$

For a small variation, we have $\delta\mathbf{X}=\mathbf{J}\,\delta\mathbf{q}$ or in the form

$$\begin{bmatrix}\delta x\\ \delta y\end{bmatrix}=\begin{bmatrix}-\ell_\alpha S_\alpha\delta\theta_\alpha-\ell_\beta S_{\alpha\beta}(\delta\theta_\alpha+\delta\theta_\beta)-\ell_\gamma S_{\alpha\beta\gamma}(\delta\theta_\alpha+\delta\theta_\beta+\delta\theta_\gamma)\\ \ell_\alpha C_\alpha\delta\theta_\alpha+\ell_\beta C_{\alpha\beta}(\delta\theta_\alpha+\delta\theta_\beta)+\ell_\gamma C_{\alpha\beta\gamma}(\delta\theta_\alpha+\delta\theta_\beta+\delta\theta_\gamma)\end{bmatrix}$$

Then, the length of projection onto the normal vector can be obtained as

$$^{n}\delta X=\frac{\delta\mathbf{X}\cdot\mathbf{n}}{\|\mathbf{n}\|}$$

Thus, we have

$$^{n}\delta X=\delta\theta_\alpha(\ell_\alpha C_{\beta\gamma}+\ell_\beta C_\gamma+\ell_\gamma)+\delta\theta_\beta(\ell_\beta C_\gamma+\ell_\gamma)+\delta\theta_\gamma\ell_\gamma$$

For a spring model of the contact, the position of the contact point is adjusted along the force direction. Suppose that the force is in the normal direction. The force can be obtained as

$$F=K\cdot^{n}\delta X=K[\delta\theta_\alpha(\ell_\alpha C_{\beta\gamma}+\ell_\beta C_\gamma+\ell_\gamma)+\delta\theta_\beta(\ell_\beta C_\gamma+\ell_\gamma)+\delta\theta_\gamma\ell_\gamma]$$

For the NTU-Hand III, each finger has at least two tactile sensors and Θ_1 is always equal to Θ_2 (joint 1 and joint 2 are coupled). Thus, we can obtain the force spring model of each joint as follows

$$F_3=K\cdot(\delta\theta_3\cdot d_3)$$

$$F_1=K\cdot[\delta\theta_3\cdot(l_3\cdot C_{11}+l_2\cdot C_1+d_1)+\delta\theta_1\cdot(l_2\cdot C_1+2\cdot d_1)]$$

where d_3 (d_1) is the distance between the joint 3 (1) to the center of the tactile sensor on the link 3 (1), l_2 and l_3 are the link lengths of the link 2 and link 3, respectively.

Based on these two equations, we can get a simple spring model for the force control by modifying the position error of the three joint angles without any complex calculations of forward and/or inverse

kinematics.

As manipulating an object, two situations are especially undesired: loss of contact and too large grasping force. Since the fingers can only exert pushing force, a large position error may cause the loss of contact and destroy the grasping stability. In order to avoid these situations, a variable spring constant is used. A large spring constant K means that the position error should be minimized. A smaller K is assigned for the situations of exerting too large force or loss of contact. This can be described by the following linguistic fuzzy rules:

Rule 1: If *contact force* is ***Adequate*** Then *K* is ***Large***.

Rule 2: If *contact force* is ***Too Large*** or ***Too Small*** Then *K* is ***Small***.

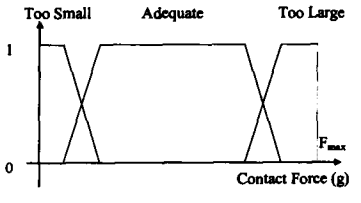

Fig. 2.3 Membership function of the contact force

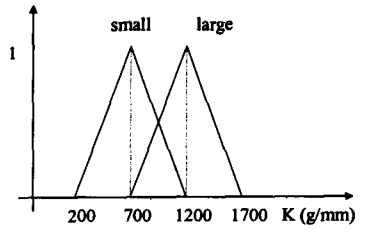

Fig. 2.4 Membership function of the spring constant

Fig. 2.3 and Fig. 2.4 show the membership functions of the input and output. The fuzzy rules are used to avoid the excessively exerting force and loss of contact. K_{max} and K_{min} are chosen according to the stiffness of the object.

With the internal spring model, the position trajectory is modified during the force compensation. If the force is not planned correctly, it will induce a large position oscillation of the object. Hence, the master/slave structure is built to reduce the effect. The master finger only executes the position, while the force on the object is distributed and adjusted by the slave one.

This structure is similar to the common master/slave structure used in [1]. The stand-alone structure is compared with the master/slave structure in Fig. 2.5.

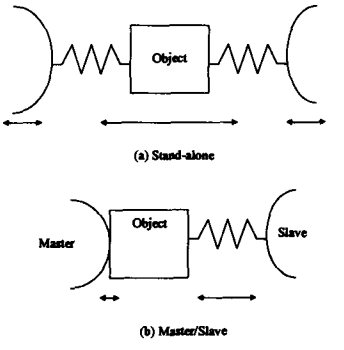

Fig. 2.5 Comparison between stand-alone and master/slave structures

In Fig. 2.5, if each finger stands alone, the object may oscillate between the two springs. For the master/slave structure, a fixed position of the master finger is expected. Then the oscillation range of the object is small.

According to the observation of the grasp of the human hand, the thumb is the most important. It provides the opposite force to generate the grasp. Most functions of the hand fail without the thumb. Hence, the thumb is chosen as the master finger, the other fingers are the slave fingers.

結論與成果

Due to the complexity of EMG signals, a high-speed microprocessor is required to process the EMG data. The TMS320C3x

floating-point DSP is used to build the EMG discriminator in this project. Fig. 3 is the hardware structure of the EMG discriminator.

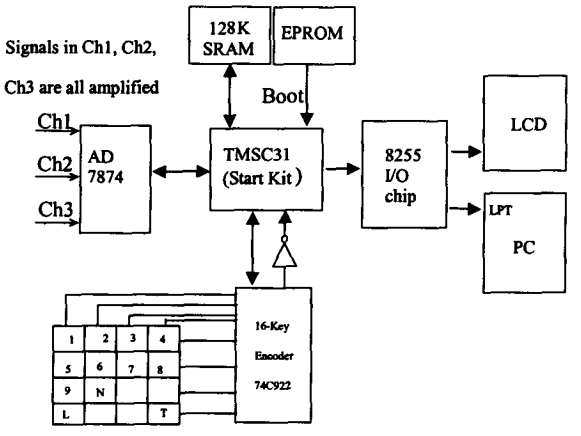

Fig. 3 Hardware structure of the EMG discriminator

This discriminator has on-line learning mode and on-line testing mode. A 4X4 keyboard is used to get the learning goal for on-line learning. After on-line learning, the controller changes to the on-line testing mode. The classification result is sent to PC by an 8051 micro controller. Thus, the amputee can control the prosthetic hand via the NTU-Hand III hand control card.

After testing each hand motion 20 times (on-line testing stage). The correct rate of discriminator is shown in Table 1.

Table 1 Motion correct rates of DSP-based system

Movement Name	DSP-based
Power grasp	95.0%
Hook grasp	90.0%
Wrist flexion	100.0%
Lateral pinch	95.0%
Flattened hand	90.0%
Index hand	60.0%
Three-jaw chuck	90.0%
Cylindrical grasp	100.0%
Average rate	90%

An OpenGL 3D graphic user interface is developed in this project to show the classification result and control the NTU-Hand III. The entire EMG controlled prosthetic system is shown in Fig 4.

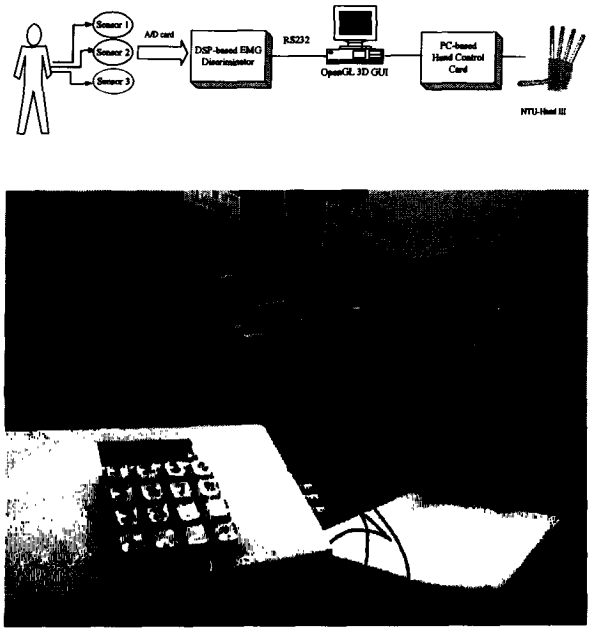

Fig. 4 The entire EMG controlled prosthetic system

The OpenGL 3D graphic user interface (GUI) is designed to control and monitor the status of the NTU-Hand III. After the classification result of the EMG discriminator is sent to PC, the OpenGL 3D GUI can control the NTU-Hand III via the ISA bus hand control card and show the joint angles or contact forces on the graph.

However, the NTU-Hand III is a modular reconfigurable prosthetic hand, the number of DOFs is not always the same and some joints may couple with others due to mechanism constraints. Not all postures can be completely implemented by the NTU-Hand III. It is important to find out a

general and reachable solution to map postures into the prosthetic hand. Here, when two or three joints of the NTU-Hand III are coupled with each others, the largest desired joint angle of one posture is chosen to map into all of those coupled joint of the NTU-Hand III.

In the eight kinds of postures selected in this thesis, the cylindrical grasp is usually used to perform a stable grasp. However, the diameter of the object is unknown, the final posture of cylindrical grasp should be set close enough to grasp even for a small cylindrical object.

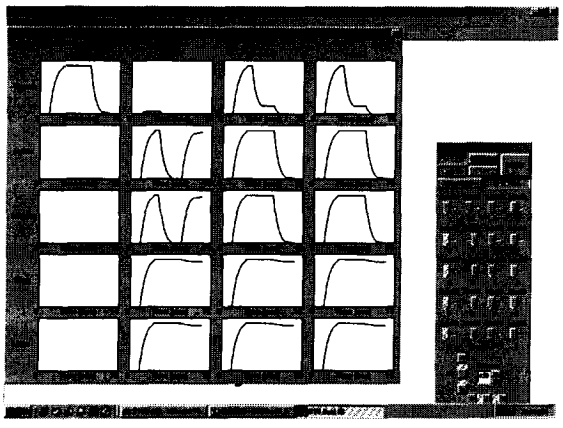

Fig. 5 Joint angles of NTU-Hand III

Fig. 5 shows each joint angle when the NTU-Hand III performs a power grasp, followed by a hook grasp, then by a wrist flexion. The sampling period is 100 msec. Fig. 6 shows the experiment when the NTU-Hand III performs the cylindrical grasp to grasp a hard cylindrical object.

Fig. 6 The experiment of grasping a hard cylindrical object

參考文獻

[1] S. Arimoto, F. Miyazaki, and S. Kawamura, "Cooperative Motion Control of Multiple Robot Arms or Fingers," *Proceedings of IEEE International Conference On Robotics and Automation*, pp.1407-1412, 1987.

[2] P.C. Doerschuk, D.E. Gustafson and A.S. Willsky, "Upper extremity Limb Function Discrimination Using EMG Signal analysis," *IEEE Transactions on Biomedical Engineering*, Vol. BME-30, No.1, pp.18-28, January 1983.

[3] D. Graupe, J. Magnussen and A.A. Beex, "A Microprocessor System for Multifunctional control of Upper-Limb Prosthesis via Myoelectric Signal Identification," *IEEE Transactions on Automatic Control*, Vol. AC-23, No.4, pp. 538-544, August 1978.

[4] D. Grupe and W.K. Cline, "Functional Separation of EMG Signals via ARMA Identification Methods for Prosthesis Control Purpose," *IEEE Transactions on*

Systems, Man and Cybernetics, Vol. SMC-5, No.2, pp.252-259, March 1975.

[5] T. Iberall, G. Sukhatme, D. Beattie and G.A. Bekey, "Control Philosophy and Simulation of a Robotic Hand as a Model for Prosthetic Hands," *Proceedings of IEEE International Conference on Intelligent Robots and Systems,* Vol.2, pp.824 –831, 1993

[6] T. Iberall, G.S. Sukhatme, D. Beattie and G.A. Bekey, "On the Development of EMG Control for a Prosthesis Using a Robotic Hand", Proceedings of IEEE International Conference on Robotics and Automation, Vol.2, pp.1753–1758, 1994.

[7] F.P. Kendall, E.K. McCreary and P.G. Provance, Muscles Testing and Function: with Posture and Pain, Baltimore, Md. : Williams & Wilkins, 1993.

[8] J.M.F. Landsmeer, "Power grip and precision handling," *Annals of the Rheumatic Diseases*, Vol. 21, pp.164-170, 1962.

[9] L.R. Lin and H.P. Huang, "Mechanism Design of A New Multifingered Robot Hand," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol.2, pp. 1471 –1476, 1996.

[10] H. Liu, T. Iberall, G.A. Bekey, "The multi-dimensional quality of task requirements for dextrous robot hand control," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol.1, pp.452–457, 1989.

[11] C.S. Pattichis, C.N. Schizas and L.T. Middleton, "Neural Network Models in EMG Diagnosis," *IEEE Transactions on Biomedical Engineering*, Vol.42, No.5, pp.486-496, May 1995.

[12] G.N. Saridis and T.P. Gootee, "EMG Pattern Analysis and Classification for a Prosthetic Arm," *IEEE Transactions on Biomedical Engineering*, Vol. BME-29, No.6, pp.403-412, June 1982.

[13] M.I.Vuskovic,A.K.Marjanski, "Program med Synergy in Dextrous Robotic Hands," *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, pp.449-455, 1993.

行政院國科會補助參加國際會議報告

一. 參加會議經過

二. 與會心得

三. 參觀活動

四. 建議

五. 攜回資料

參加 2000 年 IEEE 國際機器人暨自動化會議報

黃漢邦 教授　台灣大學機械系

一. 參加會議經過

本次會議在美國舊金山舉行，會議期間從四月二十四日至四月二十八日。因為我從三月一日至五月三十一日在日本名古屋大學訪問(國科會短期進修計劃)，乃於四月二十三日從名古屋搭機，經由東京到美國舊金山。在前往舊金山希爾頓飯店途中，頗驚訝舊金山的改變。日本商店、百貨公司、影視公司、餐館，甚至日本國旗在舊金山鬧區到處可見，日本藉由經濟力量和科技力量的擴張，大量輸出日本文化並有計劃的移民。繁榮的美國表象下，似乎也落出它的疲態。

本次大會在希爾頓飯店舉行，該飯店位於市中心，離市政大樓很近，位置和景觀都不錯。因飯店地處鬧區，在入夜後仍見遊客熙來攘往，只要不在十點以後孤身走在街上，還算安全。大會在四月二十四日晚上舉行歡迎酒會，並舉行討論會，直到晚上九點半結束。在四月二十五日傍晚，大會特別安排舊金山

灣區的船上之旅。從下午四點半在碼頭搭上船，沿著舊金山灣區航行，一邊吃晚餐，一邊觀賞舊金山大橋、惡魔島和附近的山光水色，直到晚上十點才結束這浪漫之旅。四月二十六日中午，大會舉辦今年最佳論文及晉升 Fellow 頒獎儀式。今年，華裔學者有兩位獲獎，一位獲得大會最佳論文獎；另一位獲得 Early Carrier Award。

當天晚上為大會晚宴及專題演講，在十點半左右結束。在晚宴結束後，來自台灣、大陸、新加坡、香港及美國的華人一群浩浩蕩蕩前往中國城聚餐和討論。由於台灣的總統大選才剛結束，大家的話題一直圍繞在選舉的問題上，討論異常熱鬧。四月二十七日是大會論文發表最後一天，二十八日仍有討論會。我於二十八日當天從舊金山搭飛機經由東京返回名古屋。

二. 與會心得

此次會議共有 1100 篇投搞，僅接受 641 篇，接受率約 58%，全部文章分 111 個場次發表，同一時段則有九個場次同時進行。本人在會中一共發表三篇文章，分別是"DSP-Based Controller for a Multi Degree Prosthetic Hand," "Modeling and Performance Evaluation of a Controlled IC Fab Using Distributed

Colored Timed Petri Net,"和 "Queueing Network Analysis for an IC Foundry."三篇文章都得到不少回響。

本次會議安排兩個 video sessions，由 video proceeding 的作者各自解說五分鐘，現場反應熱烈。此外，大會安排了三個座談會和五次大會特別演講。三次座談會的主題分別是"Robotics：The 20th century and Beyond," "National and International R&D Programs,"及"Mechatronics Education."大家的重點集中在未來的方向、研發計劃和機電整合教育. 五次大會特別演講的主題分別是 "Vision-Enabling Robots to Sense, Control and Interact," "Controlling Intelligent Machine," "Motion Planning：A Journey of Robots, Molecules, Digital Actors and other Artifacts," "Brains and Brawn in Humanoid Autonomy," "Hands On-Haptics and Telesurgery."

從今年的論文發表及大會所安排的特別演講與座談會來看，有下面幾個特點：

1. 全方位的發展機器人 - 從工業、農業、營建和辦公室的應用，到娛樂、家庭、個人、軍隊、太空、醫療和海底的應用。這方面以日、美、德三國為發展重心。

2. 人性納入自動化技術 - 純技術無法根本解決問題，而需考慮以人為主體的製造自動化技術。這方面以日本和美國為代表。

3. 人形化機器人的發展 – 從研究走向娛樂及軍事運用。從今年起也另外有一個探討人形化機器人的國際會議。今年會議在MIT 舉行。

4. 精細組裝 – 目前 MEMS 仍以 micro sensor 及 micro actuator 為發展主軸，但如何組裝成一有用的裝置，為目前另一發展重點。另一方向是奈米技術及有機裝置的應用。

5. Medical Robot 的發展 – 將 Robot 及自動化技術應用到 Health care，Surgery，Rehabilitation 為一發展重點，人工義肢即為一例。而遠端醫療及遠距開刀亦是一重點。

6. 機器人及自動化技術的倫理觀 – 由於機器人及自動化系統愈來愈聰明，甚至具有某種程度的思考能力，因而它們與人的互動關係愈顯微妙。

7. 機電整合教育 – 強調創新、整合與應用。須結合機械、電機、電子、資訊及網路技術，以因應二十一世界之教育新趨勢。

三. 參觀活動

在開會期間，大會安排參觀史丹佛大學及加州大學柏克萊分校，因與我論文發表時間衝突，故未前往。

四. 建議

本會議是機器人及自動化領域最重要也是最具水準的會議之一，國科會每年應儘量補助所有論文發表者與會。

五. 攜回資料

大會論文集的光碟片一張及大會輪文摘要一本。

Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Queueing Network Analysis for an IC Foundry

Jia-Yang Juang * and Han-Pang Huang ◆
Department of Mechanical Engineering
National Taiwan University
Taipei, TAIWAN 10674, R.O.C.
TEL/FAX: (886) 2-23633875
e-mail: hphuang@w3.me.ntu.edu.tw
◆Professor and correspondence addressee *Graduate students

Abstract

A hybrid decomposed queueing network model is developed based on actual operating data from one particular foundry fab for rapid analysis of several performance measures. The queueing model includes analyzer and predictor modules. The system analyzer module provides the analysis of arrival pattern and service pattern for each tool group. The system predictor module provides the forecast of important performance measures, such as products cycle time, lots remaining cycle time, tool utilization, queueing length, tool moves, stage moves. Utilizing the decomposition concept in the network model, there is no limitation on the number of tool groups as well as product families, and hence priority queue model can be applied. In addition, a modularized system, QFAB, is designed to implement the model proposed in this paper. A systematic analysis of arrival pattern and service pattern for tool groups is proposed. An approach to compute the effective tool number for tool groups is also addressed. Based on the analysis, the supervisors can gain more insights and choose the proper queueing models. Comparing the obtained results to the actual fab data, the accuracy of prediction of cycle time is satisfactory. The predicted results are much better than those obtained by the original approach used in the fab.

1 Introduction

Semiconductor manufacturing factories suffer from the problems of long manufacturing cycle time (or flow time), high work-in-process (WIP) level, poor due date performance, and expensive equipment. A lot of factors make the wafer fabrication process highly stochastic. This stochastic phenomenon makes the handling and prediction of important performance measures, such as cycle time, difficult tasks. One of the most important distinctions between pure foundry fabs and R&D fabs is that the former has to meet the requirements of customers. Accordingly, how to precisely handle the cycle time of all products (or orders) as well as other performance measures, such as average work-in-process at tool groups, throughput, and tool group utilization, becomes an important and challenging task for managers and engineers. In order to accomplish the above task, a detailed hybrid decomposed queueing network model for semiconductor foundries is proposed. Though the use of queueing models for performance evaluation of semiconductor manufacturing systems is not new, our models differs from others in the broader sense. The model can be utilized in more complex foundry fabs and extended to more general queueing models, including G/G/c priority queues.

The main goal of this paper is to develop an effective and efficient queueing analytical model, as opposed to simulation studies, for rapid and accurate performance evaluation of a complex semiconductor foundry. The model is aimed at attaining the following objectives:

- Predict several important performance measures, especially product cycle times.
- Provide an approach to analyze the arrival pattern and service pattern for each tool group.
- Incorporate with the database system in the fab.
- Be suitable for a complex foundry consisting of hundreds of tools and having highly product mix.

The implemented software package is called "QFAB" (*Q*ueueing *FAB*).

In the paper, we do not attempt to propose a new exact solution or approximation for specific queueing models, such as the solution of M/G/c or G/G/c queues. Instead, we intend to develop a procedure of performance estimation with available queueing models in the literature. Only the process and inspection tools are considered. Other types of machines, such as material handling systems and storage systems are not modeled here. The model developed here is not concerned with describing individual processes, and the physical and chemical principles that determine how and why a process operates. The information needed is the nature of the disturbances that influence time and quality, and particularly its frequency, duration, and pattern of occurrence. Scheduling problems, such as lots' release policy and lots' dispatching policy, are not considered.

2 Literature Review

In the literature, there are piles of papers proposed to model manufacturing facilities and to predict product cycle times. Such research can be classified into five types: direct estimation from historical operation data, computer simulation, analytical model, statistical model, and neuro-fuzzy based model. However, some research combines two or three methods of the five. Only the analytic model is described here.

The model proposed in this paper is one kind of analytical model. Different from simulation, an analytical model is used to determine system parameters by mathematical methods. The commonly used approach is queueing theory.

Whitt [10] pioneered modeling manufacturing processes using a general type queueing network, QNA. Snowdon et al. [8] gave a detailed survey of analytical-based queueing network computational tools relevant for manufacturing systems analysis. Chen et al. [2] proposed a naive BCMP queueing network models for an analysis of wafer fabrication facilities. The model is used to predict certain key system performance measures for an R&D fab.

In 1996, Connors et al. [3] addressed a benchmark paper. Based on other research in the literature, in particular QNA [10], Connors et al. proposed a sophisticated queueing network model for semiconductor manufacturing fabs. The model is designed for rapid performance analysis of semiconductor fabs. The model considers many detailed

analyses, such as scrap and rework processes, tool breakdown and PM. Similar to other models, there are some main drawbacks of this model. First, it assumes that products follow FCFS policy. Second, the fab itself is rather simple – there are only 72 tool groups; the maximum number of tools in one tool group is 5; about 80% of the tool groups have only one tool; it does not consider the tool group overlapping phenomenon; the authors do not explain how to obtain the given probability distributions. Third, the prediction results are compared with simulation model rather than actual fab data.

3 Model Formulation and Analysis

In this section, we describe the formulation and analysis of our queueing network model. First, we propose the concept of hybrid decomposed queueing model and several model assumptions. Second, we make a classification of typical tool types in a fab for the model. Then, the relevant queueing models are derived. At last, the procedure of hybrid decomposed queueing network model is addressed.

3.1 Hybrid Decomposed Queueing Network Model and Model Assumptions

A semiconductor foundry in abstraction is a set $\Gamma = \{1,...,G\}$ of distinct tool groups among which lots can be moved from one tool group to another. These lots are mainly controlled by a central transportation center between areas and by operators between tool groups. Lots belonging to a set $\Phi = \{1,...,F\}$ of different product families are released into the foundry. Each lot has a prescribed sequence of tool groups or tools to visit before completion.

We are particularly interested in the congestion measures of the foundry, for example, the number and cycle times of lots in the system and at tool groups. Typically, a semiconductor foundry can be viewed as a multiple-class open queueing network. Each tool group is modeled as a node in the network. Unlike the traditional open queueing network models in the literature, here we bypass the determination of traffic equations, and use the empirical data of lot arrivals as the input for each node. The reasons to do so are: first, traditional queueing network analysis is often accompanied with calculation of traffic equations or normalizing constant. As the number of tool groups or product families increases, this approach becomes infeasible. Second, in the literature, the queue discipline of all queueing network models is "first come and first serve". Unfortunately, almost all foundry fabs apply "priority policy" as dispatching rule. In our model, the nodes are treated as being stochastically independent and are approximated by a GI/G/c queue having a renewal arrival process independent of service times that are independently and identically distributed with a general distribution. Specifically speaking, we construct individual models for each tool group found in a fab. Obviously, this approach is indeed an approximation. The independence can be regarded as a generalization of the product-form solution that is valid for Markovian networks. Fig. 1 shows the concept of the hybrid decomposed queueing model.

In Fig. 1, each dot with one color denotes individual tool in the fab. Note that one specific tool may belong to several different tool groups. The "Raw Data" database is the original database in the fab and the "QFAB" database represents the designed database in a local sever.

The following assumptions are made in this paper.

1) System-level Assumptions

- The network is open rather than closed. Lots come from outside, receive operation at prescribed tool groups, and eventually leave the system.
- Each tool group has one queue. There are no capacity constraints of the queues (or waiting space) for tool groups, i.e., the buffer size of tool groups is infinite.
- The arrival and service pattern for each tool group are stationary.

2) Lot Assumptions

- Lot transportation times are included in the waiting times.
- Each lot is considered as an individual entity even though it often consists of several wafers.

3) Tool Assumptions

- Each tool group consists of one or several identical tools.
- The mean and variance of the duration and interarrival times of events that interrupt the operation are known. Such events are called non-available events in this paper.

4) Operation Assumptions

- The operators in fab follow a strict priority policy, i.e., when the tool become idle, the operator will choose the highest priority lot waiting in queue for operation.
- All non-available events are non-preemptive.

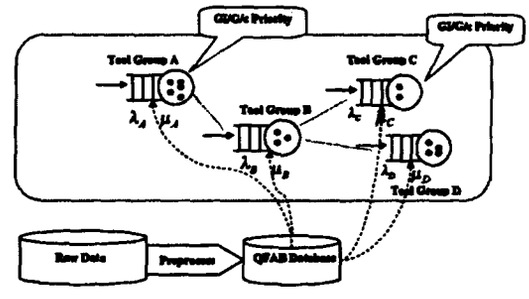

Fig. 1 Illustration of hybrid decomposed queueing network model

3.2 Classification of Tool Types

In this subsection, we focus on the classification of common tools found in the fab and derive the mean and variance expressions of processing time for each type of tools. The criterion of tool classification is according to its operation characteristics. Once the mean and variance of the processing time are derived, combined with the first two moments of interarrival times, we can determine the interested performance measures for each tool group by using the queueing formulas derived in Section 3.3.

Following the spirit of Connors et al. [3], we propose a new classification of tool groups. According to the number of wafers being process simultaneously, the tools can be roughly categorized into single-run and batch-run tools. In our classification, single-run tools consist of single-wafer, conveyor, inspection, and multi-chamber tools. Batch-run tools consist of normal batch-run and multi-stage tools.

All the operation information or probability distribution described in the following section can be obtained from empirical data in the foundry.

3.3 Derivation of Queueing Models

In this subsection, several candidate queueing formulas are examined for the corresponding tool group types. The typical queueing formulas utilized here are to calculate the

mean queueing delay (mean waiting time) or mean queue size in terms of the first two moments (mean and variance) of interarrival times and process times. Once one particular performance measure is obtained, the others can be calculated by Little's formula. In our model, we take non-available events and priority of normal products into consideration. Here we assume that non-available events are the highest priority "customer" and there are κ priority classes of normal products in the foundry.

Before the discussion of queueing model, we describe each type of non-available events that can affect the tool utilization. Then, the subsequent subsections will discuss two categories of queueing models: Single-run and batch-run tools.

1) Non-Available Events

In the foundry, a tool can stay at several possible status. The status can be roughly classified into two types: available status and non-available status. When a tool is in one of the available status, it is or has the potential to process the lots. On the other hand, when a tool is in one of the non-available status, such as PM, breakdown and etc., it cannot proceed any normal operation.

In our model, the non-available events are modeled as non-preemptive priority lots that arrive to each tool group according to renewal processes with known distributions.

Let Ω_g denotes the set of non-available events that affect tools at tool group g. The arrival rate of non-available event of type $\omega \in \Omega_g$ is denoted by $\lambda_{g,\omega}$. $\lambda_{g,\omega}$ is the reciprocal of MTBF (mean time between failures) for event ω. The processing time or MTTR (mean time to repair) for event ω at tool group g is a random variable $S_{g,\omega}$ with a known distribution, which represents the duration of the non-available event. Hence, the first two moments of $S_{g,\omega}$ can be obtained. Unfortunately, due to the absence of proper data, the second moment (or variance) of MTBF can not be obtained in our application.

2) Single-Run Tools

Single-run tools are the majority in the fabs. A great portion of tools falls into this type. For single-run tools, we derive two different queueing models: M/G/c/Priority and G/G/c/Priority. It should be noted that all the models used are priority queues in order to capture the operation traits in a real foundry.

M/G/c/Priority queues [1, 3, 6, 9]

In many real job shop systems it has been observed that the Poisson process is an adequate representation of the arrival process. Exponential distributions may not be good representations of the processing times. Therefore, M/G/c model becomes valuable in the applications of shop floor modeling.

In this case, we still assume that both lots and non-available events arrive following Poisson processes, but the assumption that the processing time of all lots is exponentially distributed is removed.

There is no exact explicit solution for the M/G/c/Priority model when c is greater than one. The approximation by Sakasegawa [9] and Connors et al. [3] is adopted. Consequently, the mean waiting time at tool group g, for Poisson arrivals, is

$$W_{q,g}^{(i)} = \frac{\rho_g^{\sqrt{2(c_g+1)}}}{c_g^2} \cdot \frac{\sum_{k=1}^{\kappa} \lambda_{g,k} E[S_{g,k}^2] + \sum_{\omega \in \Omega_g} \lambda_{g,\omega} E[S_{g,\omega}^2]}{2(1-\sigma_{g,j-1})(1-\sigma_{g,j})} \quad (1)$$

where ρ_g is the utilization of tool group g, including non-available events. Note that Eq. (1) is exact for the case c=1.

Other performance measures can be obtained by Little's formulas.

GI/G/c/Priority queues [1, 3, 10]

Queueing theory has been studied thoroughly throughout 1950s, but many problems still remain unsolved, especially the related research about GI/G/c queues. There exist several approximations for the mean queueing delay (or waiting time) of GI/G/c queues, but up to now there is no specific formula suitable for all kinds of circumstances and there is no specific approximation absolutely overreaching the others. In our model, we adopt the approximations proposed by Connors et al. [3], Whitt [10], and Buzacott et al. [1].

For GI/G/c/Priority queueing system, the mean waiting time is approximated by incorporated an adjustment factor φ into Eq. (1). The mean waiting time at tool group g is given by

$$W_{q,g}^{(i)} = \frac{\rho_g^{\sqrt{2(c_g+1)}}}{c_g^2} \cdot \frac{\sum_{k=1}^{\kappa} \lambda_{g,k} E[S_{g,k}^2] \phi_{g,k} + \sum_{\omega \in \Omega_g} \lambda_{g,\omega} E[S_{g,\omega}^2] \phi_{g,\omega}}{2(1-\sigma_{g,j-1})(1-\sigma_{g,j})} \quad (2)$$

where $\phi_{g,k} = (c_{a,g,k}^2 + c_{s,g,k}^2)/(c_{s,g,k}^2 + 1)$, the quantities $c_{a,g,k}^2$ and $c_{s,g,k}^2$ represent the squares of the variation coefficients (SVC) of the interarrivals and processing times of the lots with priority k at tool group g, respectively. While $\phi_{g,\omega} = (c_{a,g,\omega}^2 + c_{s,g,\omega}^2)/(c_{s,g,\omega}^2 + 1)$, the quantities $c_{a,g,\omega}^2$ and $c_{s,g,\omega}^2$ represent the squares of the variation coefficients (SVC) of the MTBF and MTTR of non-available events at tool group g, respectively.

Other performance measures can be obtained by Little's formula. This model is called QFAB_G/G/c model 1 in this paper.

Our second approach to the solution of mean waiting time of GI/G/c/Priority model is to substitute the results of GI/G/c/FCFS formula proposed by Whitt [10] into the GI/G/c/Priority model proposed by Buzacott et al. [1]. The analysis is as follows. A simple approximation for mean waiting time of a lot at tool group g based on heavy-traffic limit theorems is

$$W_{q,g} = \left(\frac{c_{a,g}^2 + c_{s,g}^2}{2} \right) \cdot W_{q,g\ M/M/c/FCFS} \quad (3)$$

then,

$$W_{q,g} = \left(\frac{r_g^{c_g}}{c_g!(c_g\mu_g)(1-\rho_g)^2} \right) \cdot \left(\sum_{n=0}^{c_g-1} \frac{r_g^n}{n!} + \frac{r_g^{c_g}}{c_g!(1-\rho_g)} \right)^{-1} \quad (4)$$

where the definitions of the variables are the same as before, $r_g = \rho_g / c_g = \lambda_g / \mu_g$, and $W_{q,g\ M/M/c/FCFS}$ is the mean waiting time at tool group g for an M/M/c/FCFS model.

Eq. (3) has been frequently used for M/G/c/FCFS queues and is known to perform quite well in that case. When the utilization of the tool group, ρ_g, approaches to 1, Eq. (3) is asymptotically correct for GI/G/c systems. Some additional study indicates that Eq. (3) is also reasonable for moderate values of ρ_g when $c_{a,g}^2 > 0.9$ and $c_{s,g}^2 > 0.9$, or $c_{a,g}^2 < 1.1$ and $c_{s,g}^2 < 1.1$ [10].

The approximation of the mean waiting time for the GI/G/c non-preemptive queue is given by

$$W_{q,g}^{(i)} = \left(\frac{1-\rho_g}{(1-\sigma_i)(1-\sigma_{i-1})}\right) \cdot W_{q,g\ GI/G/c/FCFS} \tag{5}$$

where the variables are the same as before and $W_{q,g\ GI/G/c/FCFS}$ is substituted by Eq. (4).

Eq. (5) is called QFAB_G/G/c model 2 in this paper.

3) Batch-Run Tools [3, 4, 5, 6]

Batch-run tools process in batch. This type of tool group can be modeled as a bulk service queueing system. Here, we will discuss two kinds of bulk service queues: M/M$^{[K]}$/c/FCFS and G/G$^{[K]}$/c/FCFS. Ghare [5] obtained the steady-state joint distribution of the number in the queue and the number of busy channels. Cromie et al. [4] extends Ghare's analytical results to obtain the explicit expressions of the measures of efficiency and delay distributions. The derivation for the mean waiting time of a lot at tool group g is then given by

$$\begin{aligned}
(P_{0,0})^{-1} &= \frac{(c_g r_g)^{c_g}}{c_g!}\left(1-\frac{1}{V}\right)^{-1} + \sum_{i=0}^{c_g-1}\frac{r_g^i}{i!} \\
P_{m,0} &= P_{0,0}\frac{r_g^m}{m!}, \quad m = 1,\ldots,c_g-1 \\
P_{c_g,n} &= P_{0,0}\frac{r_g^{c_g}}{c_g!}\left(\frac{1}{V}\right)^n, \quad n = 0,1,\ldots
\end{aligned} \tag{6}$$

where c_g is tool number of tool group g, $r_g = \lambda_g/\mu_g$, $g \in \Gamma$. Γ is the set of tool groups, and V is the single real root, lying in the interval $\left(1, c_g K_g / r_g\right)$, of the following equation

$$f(V) = \frac{r_g}{c_g}V^{(K_g+1)} - \left(1+\frac{r_g}{c_g}\right)V^{K_g} + 1 = 0$$

where K_g is the maximum batch size. The above equation can be solved using the Secant root-finding method.

The mean waiting time is then given by

$$W_{q,M/M^{K_g}/c_g/FCFS} = \frac{P_{c_g,0}V}{\lambda_g(V-1)^2} \tag{7}$$

For G/G^K/c$_g$/FCFS queueing systems, Connors et al. [3] mimicked the well-known approximation for GI/G/c queues as

$$W_{q,G/G^{K_g}/c_g/FCFS} \approx \frac{P_{c_g,0}V}{\lambda_g(V-1)^2}\cdot\frac{\left(c_{a,g}^2 + c_{s,g}^2\right)}{2} \tag{8}$$

where $c_{a,g}^2$ and $c_{s,g}^2$ are the squares of the variation coefficients of tool group g; $g \in \Gamma$. V is the same as the one in Eq. (6).

In this paper, we use FCFS queueing models rather than priority ones for batch-run tools. The reason is twofold: batch-run tools process and the batch operation are time-consuming. Hence, the operator is apt to make the batch size per operation as large as possible. Consequently, the priority of lots does not make much influence; and the queueing model G/G^K/c$_g$/Priority is too complicate to have any explicit form solution. In order to take into consideration of non-available events of batch-run tool groups, $r_g = \lambda_g/\mu_g$ in Eq. (6) should be adjusted as $r_g = \lambda_g \cdot E[\tilde{S}_g]$, where $\tilde{S}_g$ is the adjusted processing time of tool group g, which incorporates the duration of non-available events. The adjusted processing time $\tilde{S}_g$ is defined by

$$\tilde{S}_g = S_g + X_g \tag{9}$$

where S_g is the processing time of normal lots and X_g is defined as

$$X_g = \begin{cases} S_{g,\omega} & \text{w.p.} \quad \left(\lambda_{g,\omega}/\lambda_g\right), \ \omega \in \Omega_g \\ 0 & \text{w.p.} \quad \left(1 - \sum_{\omega\in\Omega_g}\lambda_{g,\omega}/\lambda_g\right) \end{cases} \tag{10}$$

The first two moments of $\tilde{S}_g$ are then given by

$$E[\tilde{S}_g] = E[S_g] + E[X_g]$$

$$E[\tilde{S}_g^2] = E[S_g^2] + E[X_g^2] + 2E[S_g]E[X_g]$$

The SVC for the adjusted process time, $c_{\tilde{s},g}^2$, used to replace $c_{s,g}^2$ in Eq. (8), is

$$c_{\tilde{s},g}^2 = \frac{E[\tilde{S}_g^2] - E^2[\tilde{S}_g]}{E^2[\tilde{S}_g]} \tag{11}$$

3.4 Modeling Procedure

1) Input Model for Each Tool Group

Probably the least glamorous and most essential task in model building is gathering data for parameter estimation. The required raw data consists of the time at which each successive lot arrives, the time at which the lot begins and ends processing, and the tool status. Once the raw data is collected, it should be preprocessed to an appropriate form.

The number of tools for each tool group recorded in database is called nominal tool number.

Once the data has been prepared adequately, we can analyze the arrival pattern and processing pattern for each tool group. The main steps are empirical distribution, parameter estimation, and goodness-of-fit test.

2) Queueing Model for Each Tool Group

After input modeling procedure, the distributions of interarrival times and processing times for each tool group are determined. As a result, we can chose the most appropriate queueing model prepared in Section 3.3 for each tool group.

However, there is still one problem remaining unsolved. As mentioned earlier, the overlapping phenomenon between tool groups is heavy. Therefore, the nominal tool number of a tool group cannot be used as the number of servers in service facility while utilizing the queueing formulas.

Here, we propose an approach to tackle this problem by making a modification on the number of tools for each tool group. The modified number of tools for tool group g is called effective tool number and is denoted by c_g^* in contrast to the nominal one, $\bar{c}_g$. Assume that the operation of tool groups is stationary during a short time span.

Let random variable $W_{q,g}$ denote the waiting time of a lot at tool group g, $f_g\left(\lambda, c_a^2, \mu, c_s^2, c\right)$ be the function of mean waiting time of tool group g, and n be the sample size. Denote the tool set vector as $\mathbf{c} = (c_1,\ldots,c_G)$. The optimal tool set vector, or effective tool number vector, $\mathbf{c}^*$ is then determined by

$$\min_{\mathbf{c}}\left(\sum_{n=1}^{n}\left|f_g(\lambda, c_a^2, \mu, c_s^2, c) - \Sigma_{i=1}^{n} w_{q,g,i}/n\right|\right) \tag{12}$$

given $f_g(\cdot), \lambda_g, c_{a,g}^2, \mu_g, c_{s,g}^2$ for $g \in \Gamma$ and n observed

waiting times $w_{q,i}$ for each tool group.

Note that Eq. (12) turns out to be a first-order linear equation with one set of independent variables, c, after substituting the given values. Hence, the existence and uniqueness of the solution of c is guaranteed. The solution of Eq. (12) is the optimal value c^* and it is served as the number of tools for each tool group in our queueing models.

Once the effective tool number for each tool group is obtained, the performance measures for each tool group as well as the ones for the entire system can be calculated.

3) Functions Provided

The functions provided by our queueing model, QFAB, can be partitioned into two parts. One provides the analysis of entire system. The other provides prediction functions. The former is called *System Analyzer Module*; the latter is the *Prediction Module*. The principal aim of the system analyzer module is to analyze the fab performance via the analysis of arrival and service patterns.

The main objective of Prediction Module is to forecast the cycle time of products and some other performance measure, such as WIP, utilization, tool group move, and stage move. Interested readers may refer to Juang [7].

Mean Cycle Time of Products

From our previous analysis, the mean waiting time and mean processing time are calculated for each tool group. This information, together with the routing flow information for each product family, allows us to compute the estimates of the average cycle time for each product family.

Suppose that we wish to calculate the average cycle time for product family $f \in \Phi$. Let $N^f = \{1,2,...,N\}$ denote the set of nominal operations for product family f. Then the cycle time of product family f with priority i, $CT^{f,(i)}$, is approximately given by

$$CT^{f,(i)} \approx \sum_{n \in N^f} \left(W^{(i)}_{q,g(n)} + S_{g(n),r(n)} \right) \tag{13}$$

where the mean waiting time and mean processing time is the same as the previous ones, and functions $g(n)$ and $r(n)$ are defined as

$g(n)$: $\{1,...,N\} \rightarrow \{1,...,G\}$ specifies the tool group at which operation n is performed;

$r(n)$: $\{1,...,N\} \rightarrow \{1,...,R\}$ specifies the recipe of operation n.

4 Results and Discussion

In this section, we demonstrate the results of analysis for a 200mm semiconductor foundry, Fab-X, located in Hsin-chu Science-based Industrial Park in Taiwan by using QFAB and give some discussion. The main products of Fab-X are memory and logic devices. There are about 200 distinct tool groups and 700 distinct tools. In a common condition, there are approximately 100 different products and 1000 different lots in the foundry at the same time. The study time span is from November 1998 to February 1999.

4.1 Arrival and Service Pattern

According to the analysis of actual data, we observe that the processing time distribution of many tool groups follows Erlang distribution, though we do not attempt to make a conclusion about the inherent distribution family of processing time suitable for all tool groups. One important issue should be emphasized is that the distribution of lot processing time *never* becomes exponential. This is why we do not adopt any G/M/c queueing model in our application.

Unlike the processing time distribution, the interarrival times distributions of almost all the tool groups have the "shape" of exponential distribution. As a result, the arrival process of a lot follows Poisson process. This property makes the application more powerful because of the relatively simple analytical structure. However, from the detailed χ^2 test, only a few number of tool groups really have Poisson arrivals. The result is shown in Table 1.

Table 1 The number of tool groups whose interarrivals follow exponential distribution.

Tool Type	# of Tool Groups
Single-Wafer	12
Conveyor	6
Inspection	2
Multi-Chamber	6
Normal-Batch	1
Multi-Stage	0
Total	27

4.2 Mean Cycle Time of Products

Based on the pre-analyzed results about the arrival and service patterns, many important performance measures can be forecasted by QFAB. In this section, the forecast results and accuracy are addressed and compared with actual fab data as well as FOX algorithm, which was proposed by Yeh et al. [11].

In order to validate our model and compare the forecast results with actual data and FOX ones, dozens of lots belonging to six different product families with different routes have been chosen. Among the product families selected, PROD_E has the maximum number of circuitry layers and total operation steps, while PROD_B and PROD_C have the minimum number.

Fig. 2 shows the actual and predicted cycle time of the selected products. The forecasting time instant is on Nov. 18, 1998.

In Fig. 2, six types of models are used. They are QFAB_Adaptive, QFAB_GGc model 1, QFAB_GGc model 2, QFAB_MGc, QFAB_MMc, and FOX. Different from other approaches, QFAB_Adaptive model constructs individual sub-model for each tool group, based on the analyzed information, such as arrival and service patterns. For example, if tool group 1 has Poisson arrivals and no evidence indicates that it has exponential process time distribution, it is modeled as M/G/c priority queue. If both the distribution of arrival and service pattern can not be proven as exponential ones, the tool group is modeled as a G/G/c priority queue. However, all the tool groups are modeled as G/G/c model 1, G/G/c model 2, M/G/c, and M/M/c priority queues in QFAB_G/G/c model 1, QFAB_G/G/c model 2, QFAB_M/G/c, and QFAB_M/M/c, respectively. The cycle time estimation of lots for some specific product is conducted at release time of the lots after the lots complete all the operation steps. From Fig. 2 and our analysis, a number of important points are observed as follows.

- The accuracy of QFAB_Adaptive model is better than others for most product families.
- The error of FOX is approximately 18%. The accuracy is better than QFAB_GGc model 2 and QFAB_MMc, but is worse than the other two models.

- The cycle time estimation of QFAB_Adaptive model is between QFAB_GGc model 1 and QFAB_MGc. In fact, the results of QFAB_GGc model 1, and QFAB_MGc are the upper bound and lower bound of QFAB_Adaptive, respectively.
- Due to high SVC of interarrival times of some tool groups, the cycle time estimation of QFAB_GGc modle 1 is usually greater than that of QFAB_MGc model. Remember that the former is equal to the latter multiplied by a factor. Please refer to Eq. (1) and Eq. (2).
- It is not surprised that QFAB_MMc model has the lowest cycle time estimation.
- The cycle time estimation of QFAB_GGc model 2 is underestimated.

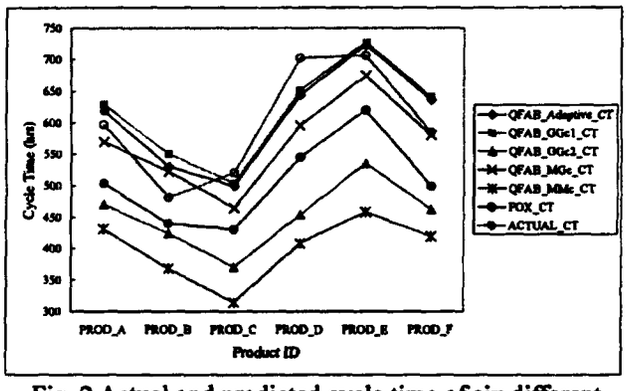

Fig. 2 Actual and predicted cycle time of six different products

Compared with the aggregated cycle time, the forecast error of waiting time for each tool group is rather large. One of the possible reasons is that discrepancies in waiting time estimation tend to cancel each other out when the aggregated cycle time is calculated.

While we cannot report any explicit comparison of cycle times for proprietary reasons, it is observed that the forecast error of QFAB_Adaptive model fall within 10% for a majority portion of products. Note that the values computed in Fig. 2 are average or expected ones for *all* priority classes of lots.

5 Conclusion

In most queueing networks model utilized in semiconductor fab in the literature, the cases under study were extremely oversimplified. The number of tool groups and tools is usually much smaller than the one in an actual fab. From the application aspects, such models may lose the representation of a real fab, which is complex inherently. The analysis of arrival and service pattern for tool groups is often ignored and there is little related research discussing the issue about the phenomenon of tool group overlapping.

According to our analysis, the empirical distribution of interarrival times for most tool groups has the "shape" of exponential distribution. However, χ^2 test indicates that only 15% of the tool groups really have Poisson arrivals. Analysis results show the distribution of service times less dispersive. Though there is no one specific distribution family can completely fit the data, Erlang or hyperexponential distribution is suggested. An approach to calculate effective tool number of tool groups is proposed to overcome the tool group overlapping problem. Comparison results indicate QFAB_Adaptive has the higher accuracy than other models, including FOX, a historical-data-based cycle time estimator. The forecast error of product cycle time is within 10%.

Acknowledgement

This work is partially supported by National Science Council under Grant number NSC 88-2218-E-002-003.

References

[1] J.A. Buzacott and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[2] H. Chen, J. M. Harrison, A. Mandelbaum, A.V. Ackere, and L.M. Wein, "Empirical Evaluation of a Queueing Network Model for Semiconductor Wafer Fabrication," *Operations Research*, vol. 36, no. 2, pp.202-215, 1988.

[3] D.P. Connors, G.E. Feigin, and D.D. Yao, "A Queueing Network Model for Semiconductor Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, no. 3, pp. 412-427, 1996.

[4] M. V. Cromie and M. L. Chaudhry, "Analytically Explicit Results for the Queueing System M/M^x/C with Charts and Tables for Certain Measures of Efficiency," *Operational Research Quarterly*, vol. 27, no. 3, pp. 733-745, 1976.

[5] P. M. Ghare, "Multichannel Queueing System with Bulk Service," *Operations Research*, vol. 16, no. 1, pp. 189-192, 1968.

[6] D. Gross and C.M. Harris, *Fundamentals of Queueing Theory*, 3rd ed., New York: John Wiley & Sons, Inc., 1998.

[7] J. Y. Juang, "Development of Hybrid Decomposed Queueing Network Model for an IC Foundry," Master Thesis, Institute of Mechanical Engineering, National Taiwan University, 1999.

[8] J. L. Snowdon and J. C. Ammons, "A Survey of Queueing Network Packages for the Analysis of Manufacturing Systems," *Manufacturing Review*, vol. 1, no. 1, pp. 14-25, 1988.

[9] H. Sakasegawa, "An Approximation formula $L_q = \alpha\rho^\beta/(1-\rho)$," *Ann. Inst. Statist. Math.*, vol. 29, pp. 67-75, 1977.

[10] W. Whitt, "The Queueing Network Analyzer," *Bell System Technical Journal*, vol. 62, no. 9, pp. 2779-2815, 1983.

[11] C.F. Yeh, H.P. Huang, J.Y. Juang, L.R. Lin, and T. Chen, "Dynamic Average Method for Cycle Time Estimator in an IC Fab," *1998 Semiconductor Manufacturing Technology WorkShop Taiwan*, IEEE Electron Devices Society Taipei Chapter, 1998.