*Computer-Aided Design and Applications*


# NURBS Interpolation for Motion Systems with Actuator Saturation

Syh-Shiuh Yeh[1] and Jin-Tsu Sun[2]

[1]National Taipei University of Technology, ssyeh@ntut.edu.tw
[2]Industrial Technology Research Institute, ricky@itri.org.tw

**ABSTRACT**

This study focuses on the design of a NURBS interpolation algorithm that considers the motion control system in the presence of actuator saturation. Since large driving forces are usually required for high-speed motion, the interpolation must consider actuator saturation in high-speed motion control systems for maintaining motion accuracy. This study proposes an interpolation algorithm that can predict the actuator saturation and adjust the parameter step size such that the generated driving forces are confined to a certain range to prevent the actuator saturation. This study also analyzes the relations among interpolation parameters, axial speed command, and the PD+ motion control law for high-speed motion control systems. Some motion tests are performed on a die bonder machine for testing the proposed approaches. The experimental results show that the designed algorithm is feasible for high-speed motion control systems for maintaining contouring accuracy during high-speed motion.

**Keywords:** NURBS interpolation, interpolation algorithm, actuator saturation, motion control.
**DOI:** 10.3722/cadaps.2008.801-810

## 1. INTRODUCTION

NURBS curves are usually applied to motion control systems because of the following reasons:

- A common mathematical model can be used to represent both standard analytic shapes and free-form curves.
- The shapes of NURBS curves can be modified easily by manipulating the shape-control parameters such as control points, weights, and knots.

However, because large driving forces are usually required to achieve high-speed motion, the motion control design for high-speed motion control systems must consider the adverse effects induced by actuator saturation including low-frequency oscillations, vibrations, overshoots, and instability of servomechanisms [10]. In general, there are two approaches to solve the motion problems induced by actuator saturation. An antiwindup compensator [2],[10],[17] is usually employed with the motion controllers for decreasing the nonlinearity effects caused by the actuator saturation. However, the feedback gains of the compensators may significantly affect both the stability and motion of the servomechanisms; thus, well-tuned feedback gains are usually required for obtaining good stability and motion results. Although the existing antiwindup compensators can provide stability and good execution in high-speed motion systems, the uncertainties and the modeling errors of the controlled plants could significantly deteriorate the executions. The command shaping method [8],[13],[15] is another approach to deal with actuator saturation problems. Fig. 1. shows a typical feedback motion control system. The command generator generates a motion command for the feedback control system, and the feedback controller generates the driving force $u_a$ for the controlled servomechanism.

By applying the command shaping method, the generated motion commands are modified such that the generated driving forces are within the linear range. Therefore, the motion control systems that employ the command shaping method are linear systems. Linear controllers such as PID and lead-lag controllers can be directly applied to the motion

control systems without considering the actuator saturation. Moreover, the command shaping method can be robustly applied to plant uncertainties and modeling errors in practical applications. In this study, the concept of the command shaping method is used to design a new NURBS interpolation algorithm such that the motion control systems with a NURBS command generator can provide good motion results.

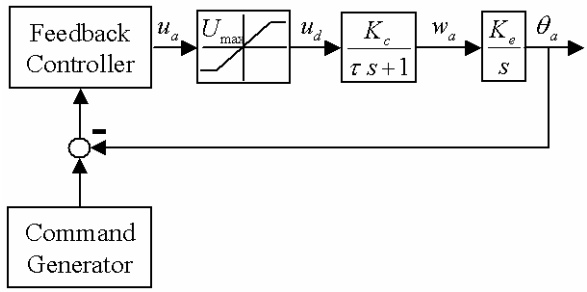


Fig. 1: Feedback control structure of motion control systems.

In the recent years, many NURBS interpolation algorithms have been developed for multi-axis motion control systems. The algorithms that consider the ACC/DEC properties [7],[11],[21], jerk limitation [11],[12], and dynamics [11],[12],[22],[26] of machine tools have been developed for improving the motion accuracy. Moreover, look-ahead functions [7],[11],[12],[21],[23] and feedrate adjustment methods [3-5],[19],[20],[24-26] have also been developed to reduce the contouring errors of the motion paths along sharp corners. Tikhon et al. [18] have proposed an interpolation algorithm that can achieve a constant material removal rate during cutting, and Ko et al. [9] have proposed an interpolation algorithm to control the cutting load for protecting cutting tools and for improving machinability. Choi et al. [6] have proposed an interpolation algorithm that can control the surface roughness during surface machining. Thus far, the interpolation algorithm that considers motion systems with actuator saturation has not attracted considerable attention and has thus not been developed. In this study, a new interpolation algorithm that considers actuator saturation is developed for high-speed motion control systems.

Uniform interpolation [1] is the simplest algorithm to interpolate the parametric curves in the motion control systems. Eqn. (1.1) shows the iterative equation for computing the parameter sequence in uniform interpolation.

$$u_{i+1} = u_i + \varepsilon_i \tag{1.1}$$

where $u_i$ is the parameter at the $i$th step and $u_{i+1}$ is the parameter at the $(i+1)$th step. $\varepsilon_i$ denotes the parameter step size, and an incorrect value of $\varepsilon_i$ may deteriorate the motion results. For instance, a large value of $\varepsilon_i$ usually causes large tracking errors because of the actuator saturation. Therefore, to maintain the tracking accuracy and eliminate the adverse effects caused by the actuator saturation, $\varepsilon_i$ must be changed according to the motion status of the servomechanisms. In this study, an interpolation algorithm that can predict the actuator saturation and adjust $\varepsilon_i$ is developed such that the generated driving forces are confined to a certain range in order to prevent the actuator saturation. Fig. 2 shows the motion control system with the proposed interpolation algorithm. For a given value of $\varepsilon_i$, according to the feedback control structure and the present motion status, the proposed interpolation algorithm computes the possible driving force and predicts the occurrence of actuator saturation. If the computed possible driving force is larger than the input limit of the applied actuator, the actuator saturation may occur and $\varepsilon_i$ must be adjusted in order to prevent the actuator saturation. In this study, the speed-controlled interpolation algorithm [25] is employed to adjust $\varepsilon_i$. This algorithm generates a parameter sequence that can drive servomechanisms with the desired curve speed. To prevent actuator saturation, a suitable curve speed is derived according to the feedback control structure and the present motion status. Then, the speed-controlled interpolation with the derived suitable curve speed is applied to compute the parameter modified step size such that the actual generated driving force is less than the input limit of the applied actuator, and this prevents the saturation. This study also identifies the relations among $\varepsilon_i$, curve speed, and motion control law.

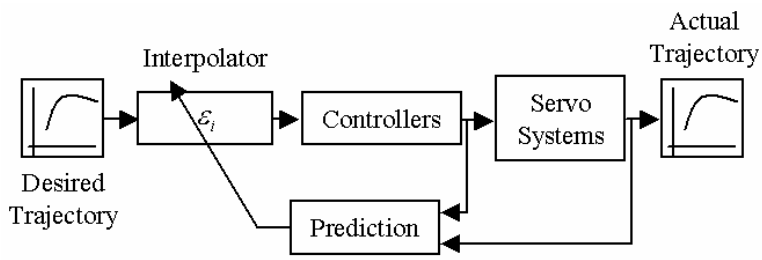


Fig. 2: Proposed motion control system.

The rest of this paper is organized as follows. Section 2 reviews the conventional design of PD and PD+ control laws for the high-speed motion control systems. Section 3 details the design of the proposed NURBS interpolation algorithm that considers actuator saturation. The design of the speed-controlled interpolation algorithm, the speed limitation for a given PD+ controller with actuator saturation, and the iteration algorithm that considers actuator saturation are also described in detail in section 3. In section 4, the proposed approach is evaluated through experiments using a biaxial die bonder machine. Section 5 concludes this study.

## 2. REVIEW OF PD AND PD+ CONTROL LAWS

The dynamic equation of a single-axis servo system can be written as follows:

$$\dot{x}(t) = Ax(t) + bu(t) \tag{2.1}$$

where $x(t) = [w(t) \quad \theta(t)]^T$ denotes the state vector of Eqn. (2.1) and $u(t)$, $w(t)$, and $\theta(t)$ are the driving force, speed, and position, respectively, of the applied servo system at time *t*. Matrix *A* and vector *b* are given by

$$A = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ K_e & 0 \end{bmatrix} \text{ and } b = \begin{bmatrix} \frac{K_c}{\tau} \\ 0 \end{bmatrix} \tag{2.2}$$

where $K_c$ and $K_e$ are the electrical gain and mechanical gain of the applied servo system, respectively, and $\tau$ is the time constant. Let the state error vector $e(t)$ be given by

$$e(t) = x_d(t) - x(t) = \begin{bmatrix} w_d(t) \\ \theta_d(t) \end{bmatrix} - \begin{bmatrix} w(t) \\ \theta(t) \end{bmatrix}$$

where $x_d(t) = [w_d(t) \quad \theta_d(t)]^T$ denotes the desired state vector and $w_d(t)$ and $\theta_d(t)$ are the desired speed and position, respectively, of the applied servo system at time *t*. Then, the error equation for the applied servo system can be obtained as follows:

$$\dot{e}(t) = \dot{x}_d(t) - \dot{x}(t) = \dot{x}_d(t) - Ax(t) - bu(t) = Ae(t) + (\dot{x}_d(t) - Ax_d(t)) - bu(t) \tag{2.3}$$

The proportional-derivative (PD) control law is the conventional design for point-to-point motion control systems. The PD control law is given by

$$u(t) = Ke(t) = K(x_d(t) - x(t)) = [K_d \quad K_p] \begin{bmatrix} w_d(t) - w(t) \\ \theta_d(t) - \theta(t) \end{bmatrix} \tag{2.4}$$

Substituting Eqn. (2.4) into Eqn. (2.3), the error equation can be rewritten as follows:

$$\dot{e}(t) = (A - bK)e(t) + (\dot{x}_d(t) - Ax_d(t)) \tag{2.5}$$

In Eqn. (2.5), the state feedback gain *K* can be designed by the pole placement method. However, the perturbation term $(\dot{x}_d(t) - Ax_d(t))$ deteriorates the tracking accuracy of the applied motion control system. To reduce the adverse effects caused by the perturbation term, the control law in Eqn. (2.4) is modified to

$$u(t) = Ke(t) + \hat{u}(t) \tag{2.6}$$

Substituting Eqn. (2.6) into Eqn. (2.3), the error equation can be rewritten as follows:

$$\dot{e}(t) = (A - bK)e(t) + (\dot{x}_d(t) - Ax_d(t)) - b\hat{u}(t) \tag{2.7}$$

Because $b$ is a column vector, the control law $\hat{u}(t)$ can be given by

$$\hat{u}(t) = (b^T b)^{-1} b^T (\dot{x}_d(t) - Ax_d(t)) \tag{2.8}$$

Substituting Eqn. (2.8) into Eqn. (2.7), the error equation can be further rewritten as

$$\dot{e}(t) = (A - bK)e(t) + \left[ I - b\left(b^T b\right)^{-1} b^T \right] \cdot \left(\dot{x}_d(t) - Ax_d(t)\right) \tag{2.9}$$

The magnitude of the perturbation term in Eqn. (2.5) is given by

$$\left\| \dot{x}_d(t) - Ax_d(t) \right\| = \sqrt{\left( \dot{w}_d(t) + \frac{w_d(t)}{\tau} \right)^2 + \left( w_d(t) - K_e w_d(t) \right)^2} \tag{2.10}$$

and the magnitude of the perturbation term in Eqn. (2.9) is given by

$$\left[ I - b\left(b^T b\right)^{-1} b^T \right] \cdot \left(\dot{x}_d(t) - Ax_d(t)\right) = \left| w_d(t) - K_e w_d(t) \right| \tag{2.11}$$

Obviously, the perturbation term $\left( \dot{w}_d(t) + \frac{w_d(t)}{\tau} \right)$ can significantly degrade the tracking accuracy in motion control systems with high-speed and high-acceleration motion commands, and it can be canceled using the control law in Eqns. (2.6) and (2.8). By substituting Eqns. (2.2) and (2.4) into Eqns. (2.6) and (2.8), the control law can be further given by

$$\begin{aligned} u(t) &= \begin{bmatrix} K_d & K_p \end{bmatrix} \begin{bmatrix} w_d(t) - w(t) \\ \theta_d(t) - \theta(t) \end{bmatrix} + \begin{bmatrix} \left( \frac{K_c}{\tau} \right)^{-1} & 0 \end{bmatrix} \begin{bmatrix} \dot{w}_d(t) + \frac{1}{\tau} w_d(t) \\ w_d(t) - K_e w_d(t) \end{bmatrix} \\ &= K_d \left( w_d(t) - w(t) \right) + K_p \left( \theta_d(t) - \theta(t) \right) + \frac{\tau}{K_c} \dot{w}_d(t) + \frac{1}{K_c} w_d(t) \end{aligned} \tag{2.12}$$

Because the control law in Eqn. (2.12) consists of the PD control and feedforward control, it is referred to as the PD+ control law in this paper. However, in practice, the actuator saturation should be considered in motion control design, and the control system should be modified as shown in Fig. 3.

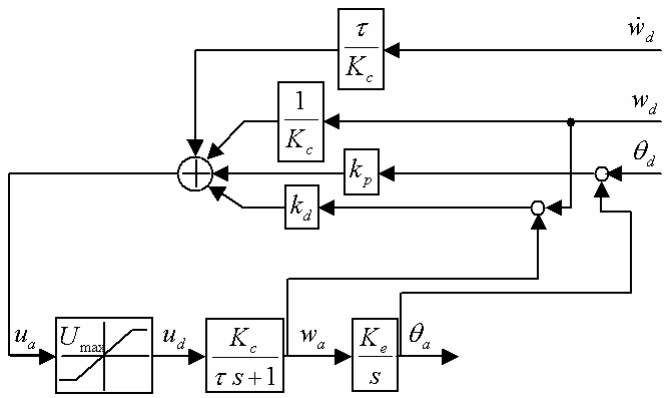


Fig. 3: Motion control system with actuator saturation.

The command generator shown in Fig. 1 generates the desired acceleration $\dot{w}_d(t)$, speed $w_d(t)$, and position $\theta_d(t)$ commands. However, the interpolation algorithm in the command generator must consider actuator saturation such that the generated acceleration, speed, and position commands can confine the driving force $u_a(t)$ in order to prevent the actuator saturation.

## 3. INTERPOLATION ALGORITHM CONSIDERING ACTUATOR SATURATION

### 3.1 Design of the Speed-controlled Interpolation Algorithm by Taylor's Expansion

Suppose that $C(u)$ is a NURBS curve function [14] and the time function $u$ is the curve parameter. The speed-controlled interpolation algorithm [25] is derived as

$$u_{i+1} = u_i + \frac{V(u_i) \cdot T_s}{\left\| D_i^u \right\|} - \frac{1}{2} \cdot \frac{\left\langle D_i^u, D_i^{u2} \right\rangle \cdot V^2(u_i) \cdot T_s^{\,2}}{\left\| D_i^u \right\|^4} \tag{3.1}$$

where $u_i$ denotes the value of $u$ at time $t = t_i$; $V(u_i)$ is the curve speed along the tangent at $u = u_i$; $T_s$ is the time step in interpolation, and the unit is second; $D_i^u = \left.\frac{dC(u)}{du}\right|_{u=u_i}$ and $D_i^{u2} = \left.\frac{d^2C(u)}{du^2}\right|_{u=u_i}$; $\|\cdot\|$ is a Euclidean norm operator and $<\ ,\ >$ is an inner product operator. Eqn. (3.1) can be further simplified to the first order by

$$u_{i+1} = u_i + \frac{V(u_i)\cdot T_s}{\left\|D_i^u\right\|} \tag{3.2}$$

For a given $u_i$, $T_s$, $V(u_i)$, and curve information such as $D_i^u$ and $D_i^{u2}$, the parameter $u_{i+1}$ can be obtained from Eqn. (3.1) or Eqn. (3.2). Moreover, we can also estimate the curve speed by using Eqn. (3.1) or Eqn. (3.2) for a given parameter step size $(u_{i+1} - u_i)$, time step, and curve information.

### 3.2 Speed Limitation for a Given PD+ Controller with Actuator Saturation

For the motion control system shown in Fig. 3, the signals are defined as follows:

$$\theta_a[k] = \theta_a(k\cdot T_s)$$
$$\theta_d[k] = \theta_d(k\cdot T_s)$$
$$w_a[k] = w_a(k\cdot T_s) \text{ can be approximated by } \frac{\theta_a[k]-\theta_a[k-1]}{T_s}$$
$$w_d[k] = w_d(k\cdot T_s) \text{ can be approximated by } \frac{\theta_d[k]-\theta_d[k-1]}{T_s}$$
$$\alpha_d[k] = \dot{w}_d(k\cdot T_s) \text{ can be approximated by } \frac{w_d[k]-w_d[k-1]}{T_s}$$

Then, the driving force $u_a[k] = u_a(k\cdot T_s)$ is derived by

$$u_a[k] = k_p\cdot(\theta_d[k]-\theta_a[k]) + k_d\cdot(w_d[k]-w_a[k]) + \frac{1}{K_c}\cdot w_d[k] + \frac{\tau}{K_c}\cdot\alpha_d[k] \tag{3.3}$$

By using the approximation of $w_a[k]$, $w_d[k]$, and $\alpha_d[k]$, Eqn. (3.3) can be rewritten as follows:

$$\begin{aligned} u_a[k] &= k_pT_s\left(\frac{\theta_d[k]-\theta_d[k-1]}{T_s} + \frac{\theta_d[k-1]-\theta_a[k]}{T_s}\right) + k_d\cdot(w_d[k]-w_a[k]) \\ &\quad + \frac{1}{K_c}\cdot w_d[k] + \frac{\tau}{K_c\cdot T_s}\cdot(w_d[k]-w_d[k-1]) \\ &= k_p\cdot T_s\cdot w_d[k] + k_p\cdot(\theta_d[k-1]-\theta_a[k]) + k_d\cdot(w_d[k]-w_a[k]) \\ &\quad + \frac{1}{K_c}\cdot w_d[k] + \frac{\tau}{K_c\cdot T_s}\cdot(w_d[k]-w_d[k-1]) \end{aligned} \tag{3.4}$$

Let

$$k_p^{'} = k_p\cdot T_s;\ k_v = \frac{1}{K_c};\ k_a = \frac{\tau}{K_c\cdot T_s}$$

Then, Eqn. (3.4) can be rewritten as follows:

$$\begin{aligned} u_a[k] &= k_p^{'}\cdot w_d[k] + k_p\cdot(\theta_d[k-1]-\theta_a[k]) + k_d\cdot(w_d[k]-w_a[k]) + k_v\cdot w_d[k] \\ &\quad + k_a\cdot(w_d[k]-w_d[k-1]) \\ &= (k_p^{'} + k_d + k_v + k_a)\cdot w_d[k] + k_p\cdot(\theta_d[k-1]-\theta_a[k]) - k_d\cdot w_a[k] - k_a\cdot w_d[k-1] \end{aligned} \tag{3.5}$$

The term $v[k]$ is defined by

$$v[k] = k_p\cdot(\theta_d[k-1]-\theta_a[k]) - k_d\cdot w_a[k] - k_a\cdot w_d[k-1]$$

Eqn. (3.5) can be rewritten as follows:

$$u_a[k] = (k_p^{'} + k_d + k_v + k_a)\cdot w_d[k] + v[k]$$

Suppose the driving force $u_a[k]$ has to be confined to $[-U_{\max}, U_{\max}]$; then, the speed command $w_d[k]$ must be limited as follows:

$$\frac{-U_{\max} - v[k]}{k_p^{'} + k_d + k_v + k_a} \le w_d[k] \le \frac{U_{\max} - v[k]}{k_p^{'} + k_d + k_v + k_a} \tag{3.6}$$

In other words, the speed command $w_d[k]$ must be confined to

$$\left[\frac{-U_{\max} - v[k]}{k_p^{'} + k_d + k_v + k_a}, \frac{U_{\max} - v[k]}{k_p^{'} + k_d + k_v + k_a}\right]$$

such that $u_a[k]$ is confined to $[-U_{\max}, U_{\max}]$.

### 3.3 Iteration Algorithm Considering Actuator Saturation

Since the estimated curve speed $V(u_i)$ can be obtained by using Eqn. (3.2) for a given parameter step size $(u_{i+1} - u_i)$, time step, and curve information, the estimated axial speed commands $w_d^x[k]$ and $w_d^y[k]$ of a biaxial motion control system are obtained by

$$\begin{bmatrix} w_d^x[k] \\ w_d^y[k] \end{bmatrix} = \frac{D_i^u}{\left\| D_i^u \right\|} \cdot V(u_i) \tag{3.7}$$

Clearly, for preventing the actuator saturation, $w_d^x[k]$ and $w_d^y[k]$ must be individually confined to a certain range, as shown in Eqn. (3.6). However, if the estimated curve speed $V(u_i)$ causes even a single axial speed command to be beyond the confinement range, the parameter step size $(u_{i+1} - u_i)$ must be adjusted before the interpolation. In this study, the speed-controlled interpolation algorithm shown in Eqn. (3.2) is applied to compute the parameter modified step size $\varepsilon_m$ as follows:

$$\varepsilon_m = \frac{V_m(u_i) \cdot T_s}{\left\| D_i^u \right\|} \tag{3.8}$$

where $V_m(u_i)$ is the modified curve speed. The parameter iteration method in uniform interpolation becomes

$$u_{i+1} = u_i + \varepsilon_m \tag{3.9}$$

Suppose the *j*-axis speed command $w_d^j[k]$ must be confined to $\left[w_d^{j\min}[k], w_d^{j\max}[k]\right]$, the modified curve speed $V_m(u_i)$ is given by

$$V_m(u_i) = \begin{cases} \left| \dfrac{w_d^{j\max}[k]}{D^j(u_i)} \right|, & \text{if } w_d^j[k] > w_d^{j\max}[k] \\ V(u_i), \text{ if } w_d^{j\min}[k] \le w_d^j[k] \le w_d^{j\max}[k] \\ \left| \dfrac{w_d^{j\min}[k]}{D^j(u_i)} \right|, & \text{if } w_d^j[k] < w_d^{j\min}[k] \end{cases} \tag{3.10}$$

where $D^j(u_i)$ is the *j*th element of the vector $\dfrac{D_i^u}{\left\| D_i^u \right\|}$. Eqn. (3.10) must be checked for all axes in order to guarantee that actuator saturation does not occur when the motion control system is executed.

### 3.4 Summary of the Proposed Interpolation Algorithm

The proposed interpolation algorithm is summarized in the following steps:

1. For a given parameter step size $(u_{i+1} - u_i)$, time step $T_s$, and curve information $D_i^u$, compute the estimated curve speed $V(u_i)$ by Eqn. (3.2).
2. Compute the estimated *j*-axis axial speed command $w_d^j[k]$ by Eqn. (3.7).
3. Compute the modified curve speed $V_m^j(u_i)$ by Eqn. (3.10).
4. Repeat steps 2 and 3 and obtain the modified curve speeds for all axes, $V_m^1(u_i)$ and $V_m^2(u_i)$.

5. Determine the minimum curve speed from the obtained modified curve speeds.
6. Set the modified curve speed $V_m(u_i)$ as the minimum curve speed obtained from step 5, $V_m(u_i) = \min\{V_m^1(u_i), V_m^2(u_i)\}$ .
7. If actuator saturation occurs within steps 2–6, compute the parameter modified step size $\varepsilon_m$ by Eqn. (3.8) and compute the parameter $u_{i+1}$ by Eqn. (3.9).
8. If actuator saturation does not occur within steps 2–6, compute the parameter $u_{i+1}$ by Eqn. (3.2).
9. Return to step 1 for the next iteration step and control step.

## 4. EXPERIMENTAL RESULTS

Fig. 4 shows the experimental setup of the die bonder machine whose servomechanism is investigated in this study. As shown in Fig. 4, a wafer that is already cut is placed on the wafer-table such that the suction gripper attached to a moving arm can lift off a die and place it on the prepared leadframe. The motion of the moving arm must be rapid enough for increasing throughput; thus, a high-speed motion control system applies to the servomechanism shown in Fig. 4. The applied motion control system consists of an industrial computer and a PC-based motion control card. The industrial computer performs functions such as the interface for human and machine operations, the interpreter for motion codes, and the central processor for handling pick-and-place procedures. The PC-based motion control card is used to generate motion commands and to record signals such as the motion commands for controllers, position outputs, and the driving forces for the applied AC servo packs. The sampling time for motion control and the time step for interpolation are limited to 1 ms.

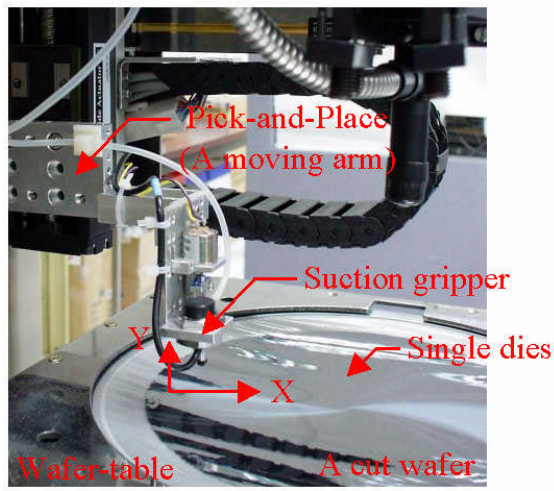


Fig. 4: Experimental setup—a die bonder machine.

Tab. 1 lists the servo parameters of the motion control system (shown in Fig. 4.) obtained by applying the system identification technique [16].

| Parameter \ Axis | X | Y |
|---|---|---|
| $\tau$ | 0.31 | 0.135 |
| $K_c$ | 2506.17 | 3156.23 |
| $K_e$ | 1591.5494 | 1591.5494 |
| $K_p$ | 0.0136 | 0.00577 |
| $K_d$ | 0.03 | 0.028 |

Tab. 1: Servo parameters of the applied motion control system.

Fig. 5 shows the contouring results of the biaxial motion control system with and without considering actuator the saturation. Clearly, because the proposed interpolation algorithm can predict the occurrence of actuator saturation and adjust the parameter step size during interpolation, the contouring accuracy is maintained when the moving arm moves along the desired trajectory. Although two sharp corners exist in the testing path, the following errors for each axis are limited to 100 $\mu$m.

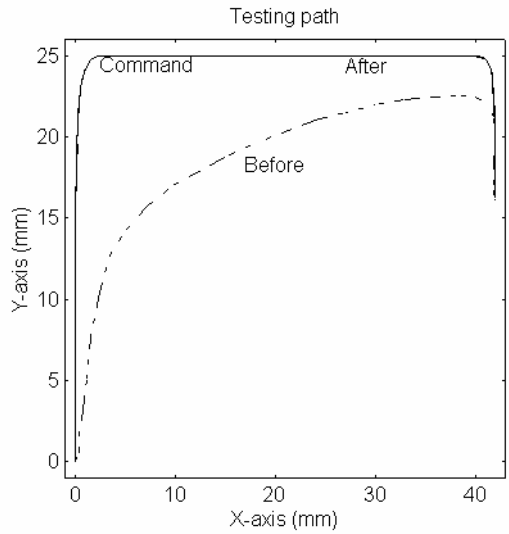


Fig. 5: Contouring results (solid: proposed interpolation algorithm considering actuation saturation; and dashed: interpolation algorithm without considering actuation saturation).

Eqn. (3.6) indicates that the axial speed commands must be confined to a certain range for preventing the actuator saturation. The computed lower and upper bounds for X-axis according to Eqn. (3.6) are shown in Fig. 6. The unit of the magnitude is pulse/ms. Fig. 6 shows that the axial speed commands are indeed confined to the lower and upper bounds during the motion processes. Thus, the contouring accuracy is maintained as shown in Fig. 5.

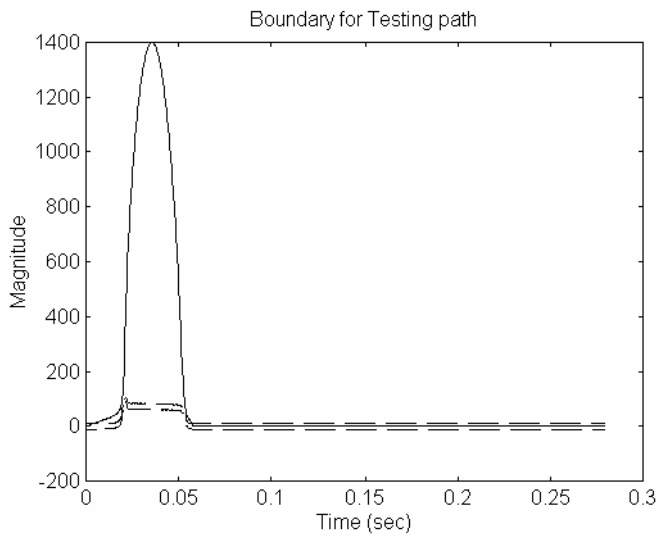


Fig. 6: Computed lower and upper bounds (solid: upper bound; and dashed: lower bound).

Fig. 7 shows the driving forces for the X-axis interpolation with and without considering the actuator saturation. Fig. 7 shows that the motion control system with the proposed interpolation algorithm confines the driving force within the input range of the applied actuator; thus, actuator saturation does not occur during motion. On the contrary, the motion control system with the interpolation algorithm without considering the actuator saturation makes the driving force larger than the input limit of the applied actuator; thus, the contouring accuracy deteriorate as shown in Fig. 5.

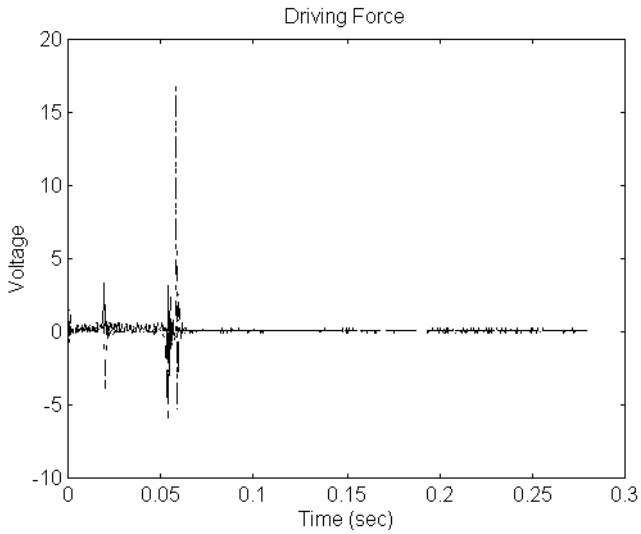


Fig. 7: Driving forces (solid: proposed interpolation algorithm considering actuation saturation; and dashed: interpolation algorithm without considering actuation saturation).

## 5. CONCLUSION

This study has proposed a NURBS interpolation algorithm that considers actuator saturation for providing precise motion results for high-speed motion control systems. Uniform interpolation is the simplest algorithm for motion control systems. However, an improper parameter step size may deteriorate the motion results because of actuator saturation. To maintain the contouring accuracy during the motion, this study has proposed the design of an interpolation algorithm that predicts the actuator saturation along all axes and adjusts the parameter step size by considering the relations among interpolation parameters, feedrate commands, and high-speed motion control laws. The proposed approach has been tested by performing some experiments have been performed on a die bonder machine. The experimental results indicate that the proposed approach can indeed maintain the contouring accuracy in high-speed motion control systems.

## 6. REFERENCES

[1] Bedi, S.; Ali, I.; Quan, N.: Advanced techniques for CNC machines, Journal of Engineering for Industry, Transactions of the ASME, 115, 1993, 329-336.

[2] Cao, Y.-Y.; Lin, Z.; Ward, D. G.: An antiwindup approach to enlarging domain of attraction for linear systems subject to actuator saturation, IEEE Transactions on Automatic Control, 47(1), 2002, 140-145.

[3] Cheng, C.-W.; Tsai, M.-C.: Real-time variable feed rate NURBS curve interpolator for CNC machining, International Journal of Advanced Manufacturing Technology, 23(11-12), 2004, 865-873.

[4] Cheng, C.-W.; Tsai, M.-C.; Maciejowski, J.: Feedrate control for non-uniform rational B-spline motion command generation, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 220(11), 2006, 1855-1861.

[5] Cheng, C.-W.; Tseng, W.-P.: Design and implementation of a real-time NURBS surface interpolator, International Journal of Advanced Manufacturing Technology, 30(1-2), 2006, 98-104.

[6] Choi, I.-H.; Yang, M.-Y.; Hong, W.-P.; Jung, T.-S.: Curve interpolation with variable feedrate for surface requirement, International Journal of Advanced Manufacturing Technology, 25(3-4), 2005, 325-333.

[7] Du, D.; Liu, Y.; Yan, C.; Li, C.: An accurate adaptive parametric curve interpolator for NURBS curve

interpolation, International Journal of Advanced Manufacturing Technology, 32(9-10), 2007, 999-1008.
[8] Eloundou, R.; Singhose, W.: Saturation compensation input shapers for reducing vibration, Proceedings of the 6th International Conference on Motion and Vibration Control, 2002, 625-630.
[9] Ko, T. J.; Kim, H. S.; Park, S. H.: Machineability in NURBS interpolator considering constant material removal rate, International Journal of Machine Tools and Manufacture, 45(6), 2005, 665-671.
[10] Li, Y.; Ang, K. H.; Chong, G. C. Y.: PID control system analysis and design: Problems, remedies, and future directions, IEEE Control Systems Magazine, February 2006, 32-41.
[11] Lin, M.-T.; Tsai, M.-S.; Yau, H.-T.: Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm, International Journal of Machine Tools and Manufacture, 47(15), 2007, 2246-2262.
[12] Liu, X.; Ahmad, F.; Yamazaki, K.; Mori, M.: Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics, International Journal of Machine Tools and Manufacture, 45(4-5), 2005, 433-444.
[13] Niu, W.; Tomizuka, M.: A new approach of coordinated motion control subjected to actuator saturation, Journal of Dynamic Systems Measurement & Control, Transactions of the ASME, 123(3), 2001, 496-504.
[14] Piegl, L.; Tiller, W.: The NURBS Book, New York Springer, 1997.
[15] Robertson, M. J.; Erwin, R. S.: Command shapers for systems with actuator saturation, Proceedings of the 2007 American Control Conference, 2007, 760-765.
[16] Söderström, T.; Stoica, P.: System Identification, Prentice Hall, 1989.
[17] Silva, J.; Tarbouriech, S.: Antiwindup design with guaranteed regions of stability: An LMI-based approach, IEEE Transactions on Automatic Control, 50(1), 2002, 106-111.
[18] Tikhon, M.; Ko, T. J.; Lee, S. H.; Sool Kim, H.: NURBS interpolator for constant material removal rate in open NC machine tools, International Journal of Machine Tools and Manufacture, 44(2-3), 2004, 237-245.
[19] Tsai, M.-C.; Cheng, C.-W.; Cheng, M.-Y.: A real-time NURBS surface interpolator for precision three-axis CNC machining, International Journal of Machine Tools and Manufacture, 43(12), 2003, 1217-1227.
[20] Tsai, M.-C.; Chung, C.-W.: A real-time predictor-corrector interpolator for CNC machining, Journal of Manufacturing Science and Engineering, Transactions of the ASME, 125(3), 2003, 449-460.
[21] Yan, C.; Du, D.; Li, C.: Design of a real-time adaptive interpolator with parameter compensation, International Journal of Advanced Manufacturing Technology, 35(1-2), 2007, 169-178.
[22] Yau, H.-T.; Kuo, M.-J.: NURBS machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces, International Journal of Production Research, 39(1), 2001, 21-41.
[23] Yau, H.-T.; Wang, J.-B.; Hsu, C.-Y.; Yeh, C.-H.: PC-based controller with real-time look-ahead NURBS interpolator, Computer-Aided Design and Applications, 4(1-6), 2007, 331-340.
[24] Yeh, S.-S.; Hsu, P.-L.: Adaptive-feedrate interpolation for parametric curves with a confined chord error, Computer Aided Design, 34(3), 2002, 229-237.
[25] Yeh, S.-S.; Hsu, P.-L.: Speed-controlled interpolator for machining parametric curves, Computer Aided Design, 31(5), 1999, 349-357.
[26] Zhiming, X.; Jincheng, C.; Zhengjin, F.: Performance evaluation of a real-time interpolation algorithm for NURBS curves, International Journal of Advanced Manufacturing Technology, 20(4), 2002, 270-276.