AN ADVANCED NURBS INTERPOLATOR FOR MECHANICAL-COUPLED LINEAR SERVO SYSTEMS

Chung-Wei Cheng¹, Chun-Hsien Yang², and Mi-Ching Tsai²
¹ITRI South, Industrial Technology Research Institute, Tainan County, Taiwan, ROC
²Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan, ROC

ABSTRACT
This study presents a real-time motion control scheme for the control of mechanical-coupled linear servo systems. The scheme comprises a variable-feedrate non-uniform rational B-spline (NURBS) curve interpolator with an S-curve velocity planning function and a synchronous controller, which are implemented in real-time using a digital signal processor. In the proposed approach, acceleration/deceleration (ACC/DEC) planning on the feedrate command is performed prior to the feedrate interpolation process such that the path command errors caused by conventional schemes in which ACC/DEC planning is executed after feedrate interpolation are eliminated. The effectiveness of the proposed control structure is demonstrated experimentally using a NURBS circle for illustration purposes.

INTRODUCTION
To satisfy the requirements for higher machining speeds, the ball-screw driven mechanisms used in computer numerical control (CNC) machines have been replaced by direct-drive linear motors nowadays. Typically, such machines physically couple two linear motors in parallel so as to achieve a sufficient thrust to accomplish one-degree movements. The resulting structure is conventionally described as a mechanical-coupled linear servo system. However, such servo systems require the use of precise synchronous control techniques to optimize their performance [1][2][3].

Synchronous control techniques can be classified as either parallel or series schemes. The former regard the two coupled motors as independent units and control them separately. In such schemes, the state signals of one motor are not fed back to the other and hence external disturbances may prevent accurate synchronous movement despite the fact that the two motors receive an identical command signal. By contrast, in series-type control schemes, the motion commands of the second motor (i.e. the slave) are not given directly, but are determined by the states in the servo loop of the first motor (known as the master). As a result, any change in the response of the master is automatically reflected in a corresponding change in the command signal provided to the slave motor. However, series-type control schemes can not handle the synchronous motion error caused by external disturbance to slave motor. For coping with the above shortcoming, this paper adopted an advanced synchronous controller which is a series-type-like control scheme [3], while the synchronous errors are feedback and controlled to each motor.

The problem of improving the accuracy of mechanical-coupled linear servo systems using advanced interpolators has attracted relatively little attention in the literature. Conventionally, CNC motion controllers only provide line or arc interpolators, and motion paths are usually given by a set of lines or arc segments. However, this approach has several disadvantages, namely (1) a significant volume of data must be processed to ensure the accuracy of the motion path and (2) the practice of executing ACC/DEC planning after the feedrate has been interpolated introduces a low-pass filter effect, which causes servo lag and hence results in machining errors.

To improve the precision of the machining performance of a CNC machine actuated using mechanical-coupled linear servo systems, this study develops a real-time, variable-feedrate NURBS curve interpolator with an S-curve velocity planning function which executes prior to the feedrate interpolation process such that the precision of the motion path is enhanced.
An integrated motion control scheme is implemented using a TI TMS320C32 digital signal processor (DSP). The performance of the proposed scheme is evaluated experimentally using a prototype mechanical-coupled linear servo system. Having determined the minimum sampling period required to enable real-time interpolation, the control scheme is used to drive the motors in such a way as to describe a NURBS circle. The actual feedrate and position responses are then compared with the instructed values in order to assess the effectiveness of the proposed controller.

NURBS CURVE INTERPOLATOR

In general, a NURBS curve in 3-D space can be described as [4]

\[ P(u) = \sum_{i=0}^{n} N_{i,k}(u)W_i = \sum_{i=0}^{n} V_iR_{i,k}(u) \quad u \in [0,1] \] (1)

and

\[ R_{i,k}(u) = \frac{N_{i,k}(u)W_i}{\sum_{j=0}^{n} N_{j,k}(u)W_j} \] (2)

where \( P(u)=(x(u) \, y(u) \, z(u))^T \), \( Q_i \) is the control point, \( W_i \) is its weighting factor, \( n+1 \) is the number of control points and \( k \) is the order of the NURBS. The terms \( N_{i,k}(u) \) and \( R_{i,k}(u) \) are referred to as the \( k \)th order basis function and the rational basis function, respectively.

The purpose of the variable-feedrate NURBS curve interpolator is to convert the NURBS curve segments into each axis’s motion command so that all axis motions are coordinated with a desired feedrate. To achieve the desired variable feedrate along a NURBS curve, it is necessary to adjust the incremental distance specified for each time interval \( T \). Accordingly, at each sampling period, a method is required to determine successive values of \( u \) such that appropriate incremental distance along the curve can be accurately generated. Applying the method proposed in [5][6], the basic procedure required to determine successive values of \( u \) can be summarized as follows.

In general, the feedrate along the NURBS curve \( P(u) \) given in Eq. (1) is defined as

\[ V(t) = \left\| \frac{dP(u)}{dt} \right\| = \left\| P^{(1)}(u) \right\| \frac{du}{dt} \] (4)

and therefore

\[ \frac{du}{dt} = \frac{V(t)}{\left\| P^{(1)}(u) \right\|} \] (5)

Furthermore, the second derivative of \( u(t) \) has the form

\[ \frac{d^2u}{dt^2} = \frac{A(t)}{\left\| P^{(1)}(u) \right\|} \cdot V^2(t) \left( \frac{P^{(1)}(u) \cdot P^{(2)}(u)}{\left\| P^{(2)}(u) \right\|} \right) \] (6)

where \( A(t)=dV(t)/dt \), \( \cdot \) denotes the inner product, and \( P^{(m)}(u) \) is the \( m \)th derivative of \( P(u) \).

Various methods have been proposed for solving Eq. (5) numerically. One well-known method is based on the Taylor’s expansion, in which the second-order approximation of \( du/dt \) at time instant \( t_k=kt \), is formulated as

\[ u_{k+1} \approx u_k + T \frac{du}{dt} + \frac{T^2}{2} \frac{d^2u}{dt^2} \] (7)

where \( u_k=u(t_k) \) denotes the value of \( u \) at the \( k \)th sampling time instant \( t_k=kt \).

Let \( V_k=V(t_k) \) and \( A_k=A(t_k) \) denote the desired feedrate and acceleration commands at time \( t_k \), respectively. From Eqs. (5)-(7), the second-order Taylor’s expansion interpolator for generating \( u_{k+1} \) can be derived as

\[ u_{k+1} = u_k + VT \frac{\| P^{(1)}(u) \|_{\infty}^{2}}{T^2} \] (8)

For simplicity, the first-order approximation of Eq. (8) is given by

\[ u_{k+1} = u_k + \frac{T}{\left\| P^{(1)}(u) \right\|_{\infty}} V_k \] (9)

Substituting the computed value of \( u_{k+1} \) into the NURBS curve given in Eq. (1) yields the next position command at time \( t_{k+1} \).

REAL-TIME MOTION CONTROL STRUCTURE

Figure 1 presents a flowchart of the proposed motion control structure for a mechanical-coupled linear servo system. As shown, the control tasks include both off-line and real-time tasks. In the off-line stage, the total arc length is
The total arc length, $L$, can be calculated off-line from the given NURBS curve segments, $P_i(u)$, $i=1,2,...,M$, using a numerical integration technique such as Simpson’s 1/3 rule [7]. Meanwhile, in the real-time stage, an S-curve velocity planning function (Figure 2) is employed to generate appropriate feedrate, $V_k$, and acceleration, $A_k$, commands at time $t_k$ which minimize the acceleration jerk ($J=dA/dt$). Having computed the velocity and acceleration commands, a variable-feedrate NURBS curve interpolator is used to compute the position command $P(u_{k+1})$ for the synchronous controller at time $t_{k+1}$.

**EXPERIMENTAL RESULTS**

Figure 3 illustrates the layout of the experimental system. As shown, the hardware components include the mechanical-coupled linear servo system, three drivers, and a PMC32-6000 motion control card equipped with a high performance TI TMS320C32 DSP. The real-time control algorithms shown in Figure 1 are implemented using C-language and are executed using the DSP. The mechanical-coupled linear servo system comprises two parallel linear motors driving the y-axis (where Y1 is the master and Y2 the slave) and a single linear motor driving the x-axis and fixed to a saddle linking the slides of the two y-axis motors.

The applied synchronous controller is depicted as Figure 4 [3], where two PI controllers $C_{mv}$ and $C_{sv}$ for master and slave motors respectively are designed to achieve the same velocity loop response, proportional position controller $C_{mp}$ of master motor is for acquiring higher stiffness, $C_{sym}$ means the synchronous error compensator (PI controller), $x$ denotes the position command after NURBS curve interpolation, $v$ is the velocity command of both motors, and $e$ shows the synchronous position error.

The sampling period $T$ is of fundamental importance in implementing real-time control schemes. Therefore, prior to the current
experimental study, an investigation was performed to determine the minimum sampling period required to allow the real-time implementation of the proposed control scheme. Table 1 presents the computational times of two of the major processing operations, namely variable-feedrate NURBS curve interpolation based on S-curve velocity planning and synchronous controller, respectively. The results indicate that for real-time control, the motion control structure with a first-order NURBS interpolator requires a sampling period of $T \geq 0.64$ ms, while that with a second-order NURBS interpolator requires a sampling period of $T \geq 1.12$ ms.

The performance of the proposed controller was evaluated by considering the case of a NURBS circle with a radius of 200 mm (total arc length is 1256.6371 mm). Setting a maximum feedrate of $V_{\text{max}}=200$ mm/s, a maximum acceleration of $A_{\text{max}}=150$ mm/s², and an average acceleration of $A_{\text{avg}}=100$ mm/s², the total motion time is found to be 8.2832 s. Figure 5 compares the command and actual feedrate and position tracking responses achieved using a second-order variable-feedrate NURBS interpolator with a sampling period of $T=2$ ms. The good agreement between the two sets of results confirms the effectiveness of the proposed control scheme.

### TABLE 1. Computational times of proposed interpolator and control scheme (ms).

<table>
<thead>
<tr>
<th></th>
<th>First-order</th>
<th>Second-order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute $P_1^1(u)$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Compute $P_2^2(u)$</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Variable-feedrate NURBS curve interpolator</td>
<td>0.54</td>
<td>1.02</td>
</tr>
<tr>
<td>Synchronous control</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>Total</td>
<td>0.64</td>
<td>1.12</td>
</tr>
</tbody>
</table>

**CONCLUSIONS**

This study has proposed an advanced motion control structure comprising an S-curve velocity planning function, a variable-feedrate NURBS curve interpolator and a synchronous controller to improve the precision of a mechanical-coupled linear servo system. The proposed control structure has been implemented in real-time using a TI TMS320C32 DSP and its performance evaluated experimentally taking the case of a NURBS circle for illustration purposes. The results of a preliminary investigation showed that when first- and second-order Taylor’s expansions were employed for the variable-feedrate NURBS curve interpolator, sampling periods of 1 ms and 2 ms, respectively, were sufficient to achieve a real-time implementation. Meanwhile, the experimental evaluation results have confirmed that the proposed approach, in which ACC/DEC planning on the feedrate command is executed prior to feedrate interpolation, reduces the discrepancy between the command and actual feedrate and position tracking responses and therefore minimizes errors in the motion path.

**REFERENCES**