

Module Design of S-curve Acceleration and Deceleration Control Based on FPGA for Numerical-control Systems

Mengfeng Shen¹, Ning Li^{2*}, and Shuang Liu³

¹ Hangzhou Polytechnic, Hangzhou 311402, P.R. China
shenmengfeng@163.com

² Hangzhou Dianzi University Information Engineering College, Hangzhou, 311305, P.R. China
lintlt515@126.com

³ Hangzhou Xiaoshan Technician College, Hangzhou 311201, P.R. China
534971532@qq.com

Received 1 January 2025; Revised 27 April 2025; Accepted 28 April 2025

Abstract. A high-frequency sampling S-curve acceleration and deceleration control algorithm utilizing a field-programmable gate array (FPGA) have been implemented to address the challenges associated with the complex S-curve acceleration and deceleration control algorithm in numerical control systems. It also includes the issues existing in hardware implementation and the discontinuity of transition points in the detection results. A discrete sampling iteration formula for the 7-segment S-curve acceleration and deceleration has been derived. Furthermore, an S-curve acceleration and deceleration control module, encompassing system clock processing, speed frequency calculation, transition point determination, and drive pulse generation, has been developed using hardware description language (HDL). The criteria for determining each transition point during the motion of both 7-segment and complete S-curves have been thoroughly analyzed. A simplified CNC system hardware platform based on a motion controller was constructed. Through extensive analysis of the frequency and position curves derived from motor encoder feedback pulses, it was observed that the motor's acceleration and speed transitions are smooth, devoid of any abrupt steps or discontinuities. The acceleration and deceleration periods are brief, with no instances of step loss, thereby enhancing motor operational efficiency and system reliability. Additionally, the S-curve acceleration and deceleration control module is characterized by its straightforward design, high sampling frequency, rapid computation speed, and the capability for reconfiguration and reuse.

Keywords: Field Programmable Logic Gate Array (FPGA), motion controller, CNC system, S-curve acceleration and deceleration, modular design

1 Introduction

A stepper motor is an electromagnetic mechanical device that converts discrete electrical pulse signals into angular displacement or linear displacement [1]. Acceleration and deceleration control of motor is always used in computer numerical control (CNC) systems for preventing impact, out-of-step and oscillation induced by velocity jump, thereby ensuring both stability and accuracy in motor operation is necessary [1-3]. Traditional CNC systems rely on microcontrollers (MCUs) or digital signal processors (DSPs) for motion control, but these solutions often struggle with real-time performance, jitter, and computational bottlenecks when handling complex trajectory planning. FPGAs, with their parallel processing capabilities, low-latency signal generation, and reconfigurability, offer significant advantages in high-precision motion control. However, if the input pulse frequency of the motor changes too quickly or there is excessive torque fluctuation [3], the rotor may fail to keep up with the changes in electrical signals due to its inertia and limited rotational resolution [4]. This will result in issues such as motor missing steps or overshooting. To reduce the missing steps or overshoot in the motion of stepper motors, various closed-loop control algorithms have been studied and applied. A series of control methods including trapezoidal, exponential and S-curve have been carried out for acceleration and deceleration control of motor. However, the two methods of trapezoidal and exponential are easily resulting in the oscillation of

* Corresponding Author

the machine tool, the acceleration easily suggested the inapplicability working scenarios with high velocity and high precision [4-6]. Qian [6] optimized the architecture of the “Economy Plus (ECP)” using high-performance Digital Signal Processing (DSP) slices on field-programmable gate arrays (FPGAs). S-curve acceleration and deceleration control can well solve above problems owing to the continuity of acceleration [7]. In addition, S-curve method should achieve the control of parameters in different phases and judge the transition points, with high theoretical research values. On that basis, Luo et al. [8] proposed a high-speed S-curve acceleration and deceleration control algorithms with smooth transition of acceleration derivative and divided S-curve into 15 segments; the operating process was smoother but the algorithm was quite complex, leading to the reduction of the motor’s operating efficiency. Zhu et al. [9] considering the flexibility of trigonometric motion profiles and the higher motion efficiency of S-curve motion profiles, the authors add the polynomial parts into the jerk profile of the cosine function acc/dec algorithm to hold the jerk when it reaches the maximum so that the motion efficiency can increase and decrease residual vibration at the same time. Yu et al. [10] Aiming at increasing the smoothness of toolpath and reducing machining vibration, a smooth S-curve acceleration/deceleration (AD) control algorithm based on the jounce limited profile is proposed, which achieved continuous derivability of velocity and acceleration. According to the experimental data, the machining efficiency can be enhanced. Nevertheless, the time node in various segments should be determined in advance and the calculation parameters should be calculated in real time, thereby setting high requirements on software programming system. The acceleration and deceleration control algorithms as described in Refs. [7-10] can hardly be achieved with hardware. Reference [11] adopted the concept of hardware reuse to design trapezoidal and exponential acceleration/deceleration control modules, subsequently validating their functionality. In Reference [12], the field-programmable gate array (FPGA) was utilized to implement trapezoidal acceleration/deceleration control, with a moving average filter employed to mitigate the adverse effects caused by abrupt acceleration changes. Reference [13] achieved both trapezoidal and S-curve acceleration/deceleration control in FPGA through digital convolution methods; however, the deceleration phase was prone to constant-velocity segments due to pulse compensation, thereby compromising deceleration efficiency. FPGA exhibits distinctive characteristics, notably its high parallel processing capability and reconfigurability. These features empower FPGA to implement S-curve acceleration/deceleration control, enabling real-time processing of complex control algorithms and facilitating high-precision, high-frequency control outputs. Furthermore, the reconfigurability of FPGA allows the system to be adjusted and optimized according to actual requirements, thereby enhancing its adaptability and scalability.

FPGAs excel in real-time trajectory generation due to their hardware-level parallelism. Unlike sequential processors, FPGAs can compute multiple motion axes simultaneously, ensuring smooth and precise acc/dec profiles. Alhomoud et al. [14] implemented an adaptive S-curve acceleration algorithm on an FPGA, achieving sub-microsecond latency, which is crucial for high-speed machining. Their work demonstrated that FPGA-based control reduces timing jitter by over 70% compared to DSP-based solutions. Similarly, Wang [15] proposed a dynamically reconfigurable FPGA architecture for multi-axis CNC systems. By leveraging partial reconfiguration, their approach allowed on-the-fly adjustment of acc/dec parameters without stopping the machine, improving flexibility in industrial applications. Xiong et al. [16] developed an FPGA-based pulse-width modulation (PWM) controller for servo motors, showing that hardware-accelerated interpolation reduces tracking errors by 30% in high-speed CNC operations. Moreover, FPGAs support advanced interpolation algorithms, such as non-uniform rational B-spline (NURBS) interpolation, which is computationally intensive for conventional processors. A recent study by Zhu et al. [17] demonstrated that an FPGA-accelerated NURBS interpolator achieved real-time performance at 1 MHz update rates, enabling ultra-precision machining.

This study successfully implemented the design of high-frequency S-curve acceleration/ deceleration control modules within an FPGA-based motion controller operating under constant-pulse driving mode, utilizing VHDL language. The FPGA’s specific applications in high-frequency S-curve control encompass critical functions such as high-frequency sampling, pulse generation, and real-time acceleration/velocity calculations. Additionally, a hardware experimental platform was established to validate the accuracy of the S-curve in acceleration/deceleration control. The results demonstrated a series of advantages of the designed module, including high-speed performance, favorable reliability, and smooth feed velocity.

2 Principle and Algorithm of S-Curve Acceleration and Deceleration Control

2.1 Principle of S-curve Acceleration and Deceleration Control

The S-curve acceleration-deceleration (acc/dec) profile is a motion control algorithm that modifies the traditional trapezoidal acc/dec by introducing a time-dependent jerk (the derivative of acceleration). This results in a smooth, continuous transition between motion states, avoiding abrupt changes in acceleration that can cause mechanical vibrations and wear. S-curve acceleration and deceleration control is characterized by a continuous variation in the acceleration profile throughout the acceleration and deceleration phases. During the motion process, the derivative of acceleration (jerk) remains constant. By controlling jerk, the S-curve ensures continuous acceleration changes, eliminating sudden force variations that occur in trapezoidal profiles. The sign of the jerk can be altered across different phases to facilitate a smooth transition in acceleration, thereby mitigating motor vibration caused by abrupt acceleration changes [18]. The fundamental principle of S-curve acceleration and deceleration control is illustrated in Fig. 1.

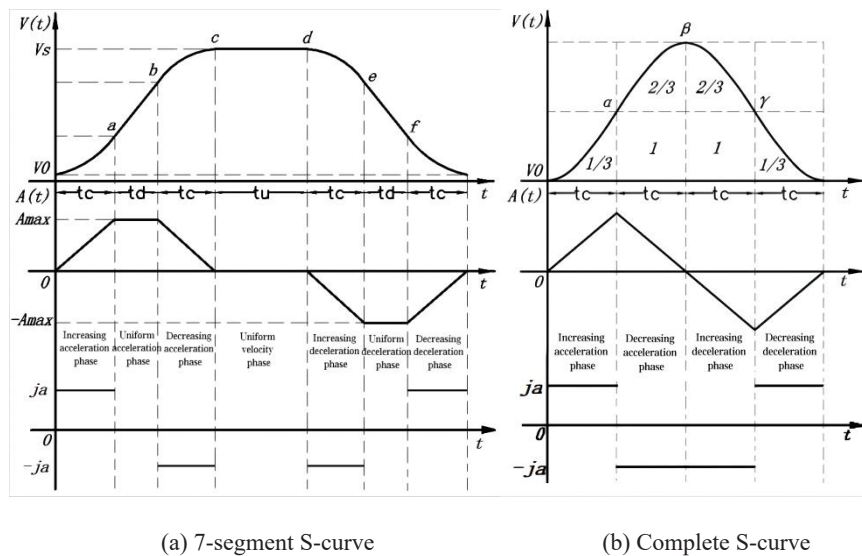


Fig. 1. Illustration of the principle of S-curve acceleration and deceleration control

As illustrated in Fig. 1(a), a typical S-curve profile generally consists of seven segments divided into three distinct phases. Specifically, the acceleration phase comprises increasing, uniform, and decreasing acceleration segments, while the deceleration phase includes increasing, uniform, and decreasing deceleration segments. The variations in key parameters differ across these phases. During the increasing acceleration phase, the derivative of acceleration (jerk, j) remains constant at j_a , indicating a linear increase in acceleration (A) from 0 to the maximum value A_{max} . Concurrently, the driving speed (V) increases in a parabolic manner. In the uniform acceleration phase, the driving speed increases linearly at the maximum acceleration A_{max} . In the decreasing acceleration phase, the sign of the jerk changes to $-j_a$, causing the acceleration (A) to decrease linearly to 0. During the uniform speed phase, both the acceleration and its derivative (jerk) remain at 0. The deceleration phases can be considered as a mirror image of the acceleration phases. As depicted in Fig. 1(b), if the predefined motor stroke is insufficient to achieve the maximum driving speed V_s and the acceleration cannot reach its maximum value, the uniform acceleration, uniform velocity, and uniform deceleration phases will not be fully realized. In such cases, the acceleration curve is referred to as a complete S-curve, whereas the former is termed a 7-segment S-curve.

When the acceleration reaches A_{max} , the system transitions from the increasing acceleration phase to the uniform acceleration phase. Ensure that the computed values adhere to physical constraints, such as maximum motor speed V_s and torque limits. If the target velocity (V) cannot be achieved within the given displacement, adjust the acceleration and j values accordingly. Utilize the reconfigurability of FPGA to dynamically adjust the motion profile based on real-time feedback. This includes modifying the jerk and acceleration values to compensate for external disturbances or changes in system parameters.

2.2 Derivation of the Curve Formula

Next, giving the sampling period T_s , the maximum driving velocity V_s and the initial velocity V_0 , the S-curve formula was derived. As shown in Fig. 1(a), 4 time periods when the derivative of acceleration j is not 0 all equal to t_c ($t_c = N_c \cdot T_s$, where N_c denotes the number of the pulses within t_c). Similarly, the uniform velocity time equals to t_u . When the system can reach the maximum acceleration A_{max} , both uniform acceleration and deceleration time equal to t_d ($t_d = N_d \cdot T_s$, where N_d denotes the number of the pulses within t_d). Additionally, according the linear relation between the maximum acceleration and the derivative of acceleration, it can be derived that $t_c = A_{max}/j_a$.

Based on the integral relation among V , A and j [19], the discrete sampling iterative formulas of the current acceleration $A(t)$ and the current velocity $V(t)$ using the 7-segment S-curve algorithm can be derived as the following formulars.

$$A(t) = \begin{cases} j_a i T_s & i = \lceil t / T_s \rceil & 0 < t \leq t_c \\ A_{max} & & t_c < t \leq t_c + t_d \\ A_{max} - j_a i T_s & i = \left\lceil \frac{t - t_c - t_d}{T_s} \right\rceil & t_c + t_d < t \leq 2t_c + t_d \\ 0 & & 2t_c + t_d < t \leq 2t_c + t_d + t_u \\ -j_a i T_s & i = \left\lceil \frac{t - 2t_c - t_d - t_u}{T_s} \right\rceil & 2t_c + t_d + t_u < t \leq 3t_c + t_d + t_u \\ -A_{max} & & 3t_c + t_d + t_u < t \leq 3t_c + 2t_d + t_u \\ -A_{max} + j_a i T_s & i = \left\lceil \frac{t - 3t_c - 2t_d - t_u}{T_s} \right\rceil & 3t_c + 2t_d + t_u < t \leq 4t_c + 2t_d + t_u \end{cases} \quad (1)$$

$$V(t) = \begin{cases} V_0 + j_a T_s^2 \sum_{i=0}^N i & N = \lceil t / T_s \rceil & 0 < t \leq t_c \\ V_0 + j_a T_s^2 \sum_{i=0}^{N_c} i + A_{max} T_s N_d & N = \left\lceil \frac{t - t_c}{T_s} \right\rceil & t_c < t \leq t_c + t_d \\ V_0 + j_a T_s^2 \left(\sum_{i=0}^{N_c} i - \sum_{i=0}^N i \right) + A_{max} T_s (N_d + N) & N = \left\lceil \frac{t - t_c - t_d}{T_s} \right\rceil & t_c + t_d < t \leq 2t_c + t_d \\ V_s & & 2t_c + t_d < t \leq 2t_c + t_d + t_u \\ V_s + j_a T_s^2 \sum_{i=0}^N i & N = \left\lceil \frac{t - 2t_c - t_d - t_u}{T_s} \right\rceil & 2t_c + t_d + t_u < t \leq 3t_c + t_d + t_u \\ V_s + j_a T_s^2 \sum_{i=0}^{N_c} i + A_{max} T_s N_d & N = \left\lceil \frac{t - 3t_c - t_d - t_u}{T_s} \right\rceil & 3t_c + t_d + t_u < t \leq 3t_c + 2t_d + t_u \\ V_s - j_a T_s^2 \left(\sum_{i=0}^{N_c} i - \sum_{i=0}^N i \right) - A_{max} T_s (N_d + N) & N = \left\lceil \frac{t - 3t_c - 2t_d - t_u}{T_s} \right\rceil & 3t_c + 2t_d + t_u < t \leq 4t_c + 2t_d + t_u \end{cases} \quad (2)$$

where t denotes the current moment, i and N denotes the number of the pulses within the current time period.

The S-curve smooth transitions minimize vibrations, extending machine lifespan and improving positioning accuracy. Abrupt acceleration changes in trapezoidal profiles induce tool chatter, negatively affecting surface quality. The S-curve's gradual acceleration reduces cutting force fluctuations, ensuring smoother finishes in high-precision machining. CNC machines operating at high speeds require precise acceleration control to avoid overshoot and resonance. The S-curve jerk-limiting feature allows stable high-speed motion without sacrificing accuracy. In variable-load conditions, the S-curve can dynamically adjust jerk and acceleration to maintain smooth motion. This adaptability is crucial for multi-axis CNC systems, where different axes may experience varying inertial loads. Sudden acceleration changes in trapezoidal profiles cause current spikes in servo motors, increasing energy consumption and heat generation. The S-curve gradual transitions reduce peak current demands, improving energy efficiency and motor longevity. When implemented in FPGA-based control systems, it

enables real-time adaptability, making it ideal for high-speed, high-precision machining applications.

3 Design of the System Hardware Platform

The current system is primarily composed of a human-machine interface, a motor controller, a servo motor, and a driver. As illustrated in Fig. 2, a series of parameters including the number of constant-length pulses P , the derivative of acceleration j_a , the maximum acceleration A_{max} , the maximum driving velocity V_s , and the initial velocity V_0 can be configured via the human-machine interface and transmitted to the FPGA-based motor controller through a serial bus. The motion controller integrates several functional modules, such as the acceleration/deceleration control module, the interpolation control module, the electronic handwheel control module, and the auto-reset control module. Based on the input parameters received via the bus, the motion controller performs velocity matching and generates driving pulses with a constant number of pulses. These pulses are then transmitted to the driver through a differential channel after interpolation control. Key parameters, including phase detection, pulse signals, and filters, were determined based on established literature [20-23].

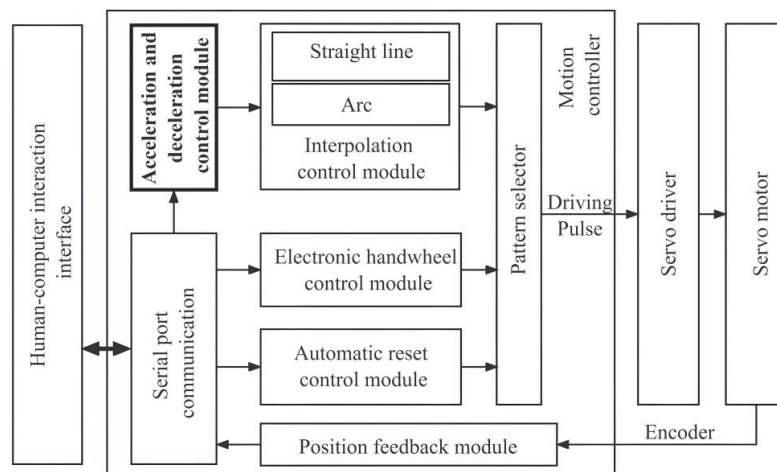


Fig. 2. Design flow of the hardware platform

The position pulse control mode is widely employed in CNC systems for servo control applications. In this mode, a series of continuous pulses is utilized to regulate the motion stroke of the servo motor, while the pulse frequency determines the motor velocity [24]. In this study, the constant-length pulse-driven output mode was adopted, and a pulse generator based on the S-curve algorithm was implemented using gate arrays. The output driving pulse frequency is achieved through an adjustable frequency divider, which offers a wide range of adjustability.

4 S Implementation of S-curve Acceleration and Deceleration Control in FPGA

The design concept and internal functional architecture of the S-curve acceleration and deceleration control module are comprehensively described as illustrated in Fig. 3. The module incorporates a range of functionalities, including system clock processing, velocity frequency calculation, transition point determination, and driving pulse generation. Input data for calculations are supplied via an external bus. Data exchange among various functional units is facilitated by the parallel bus within the FPGA. The parallel operation of these functional units demonstrates high temporal efficiency and supports functional reconfiguration [25].

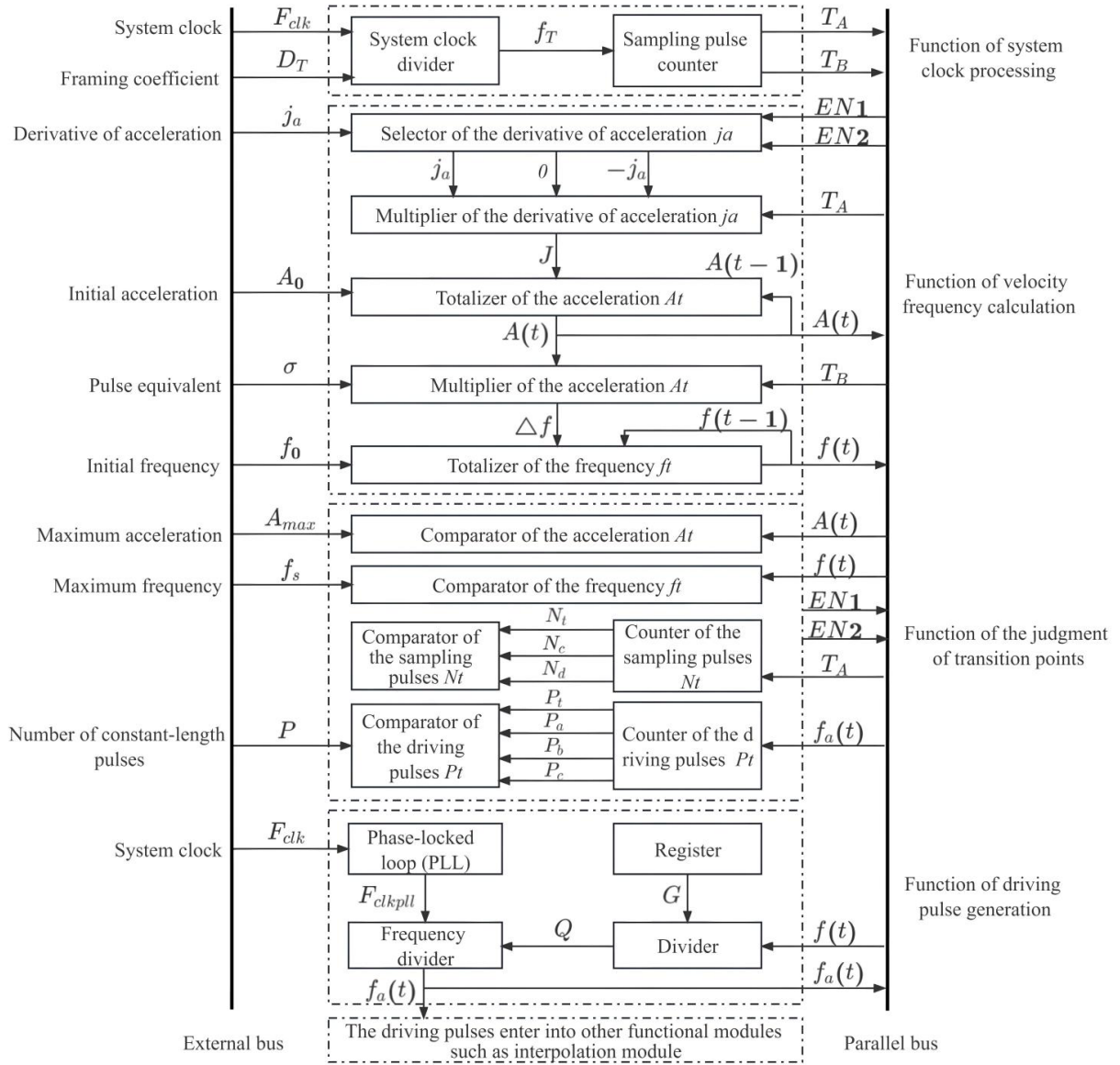


Fig. 3. Working principle of the designed S-curve acceleration and deceleration control module

4.1 Function of System Clock Processing

The system clock processing functional unit comprises a system clock divider and a sampling pulse counter. In this experiment, the FPGA system clock operates at 50 MHz. The frequency division coefficient D_T is input via the human-machine interface to divide the system clock frequency F_{clk} , thereby generating the sampling pulse frequency f_T and enabling pulse counting. Consequently, the desired sampling period can be accurately determined.

$$f_T = F_{clk} / D_T \quad (3)$$

$$T_s = D_T / F_{clk} \quad (4)$$

The frequency division coefficient D_T is user-configurable, allowing for precise control over the system's timing characteristics. D_T can provide flexibility in tuning the system's response to different operational conditions, and balance the computational efficiency and control precision. Within a single sampling period T_s , enhancing real-time performance, a complete acceleration/deceleration control cycle is executed, which directly influences the motion performance of the servo motor. Specifically, the sampling frequency f_T plays a critical role in determining the smoothness and responsiveness of the motor's motion. A higher f_T improving motion accuracy and reducing mechanical vibrations results in finer control resolution but increases computational load, whereas a lower f_T reduces processing demands but may compromise motion accuracy. The sampling pulse counter generates two asynchronous triggering pulses, T_A and T_B , facilitating precise control, which share the same frequency. These pulses serve as timing references for the velocity control module's multiplier, ensuring synchronized operation across the system. Total motion time was minimized while respecting jerk/acceleration constraints. And transition points are calculated using Pontryagin's maximum principle.

4.2 Function of Velocity Frequency Calculation

Pulse synchronization control and S-curve motion profiles exhibit a tightly coupled relationship in precision motion systems. The S-curve defines the ideal kinematic trajectory while pulse synchronization ensures this trajectory is accurately executed by multiple axes. According to the thought of integral, the displacement increment within the known time period can be regarded as the accumulation in this period, and the velocity increment can be regarded as the accumulation of acceleration. Each pulse's rising/falling edge must align precisely with S-curve phase transitions, and pulse frequency directly encodes the S-curve's velocity profile. On that basis, the selector and the multiplier of the derivative of the acceleration j_a , the accumulator and the multiplier of the acceleration A_i , and the accumulator of the frequency f_i . As regard to the selector of the derivative of the acceleration j_a , the derivative of the acceleration j_a can be selected by decoding the external control signal EN1 and EN2, and the derivative of the acceleration j_a is also input via the human-computer interface. The selector can select three parameters j_a , 0 and $-j_a$. Table 1 lists the detailed selections and coding/decoding values. Assuming that J denotes the acceleration increment, J equals to the product of the value output from the selector of j_a and T_s in the multiplier, and then enters into the accumulator of A_i . By taking the increasing acceleration segment as the example, the following Eq. (5) can be derived as the formular.

$$J = j_a \cdot T_s \quad (5)$$

The rotation angle of a single pulse driven motor is fixed, which is also referred to as the motor's pulse equivalent pulse (σ). Therefore, the current motor velocity can be calculated as:

$$V(t) = \sigma \cdot f(t) \quad (6)$$

where $f(t)$ denotes the current predicted pulse frequency. According to Eqs. (1), (2) and (5), the following expressions can be derived:

$$A(t) = \begin{cases} A(t-T_s) + J \\ N \cdot J \end{cases} \quad N = [t/T_s] \quad (7)$$

$$V(t) = \begin{cases} V(t-T_s) + A(t) \cdot T_s \\ V_0 + T_s \cdot J \sum_{i=0}^N i \end{cases} \quad N = [t/T_s] \quad (8)$$

According to Eqs. (6), (7) and (8), the following expression can be derived:

$$f(t) = \begin{cases} f(t-T_s) + A(t) \cdot T_s / \sigma \\ f_0 + T_s \cdot J \sum_{i=0}^N i / \sigma \end{cases} \quad N = \lceil t / T_s \rceil \quad (9)$$

$$\Delta f = A(t) \cdot T_s / \sigma \quad (10)$$

According to Eq. (7), the former acceleration value $A(t-1)$ can be adopted as the input of the A_i accumulator and added to the acceleration increment J . Afterwards, the A_i multiplier can multiply $A(t)$, T_s with the reciprocal of σ according to Eq. (10), and then the pulse frequency increment Δf is output and input to the accumulator of f_i . Afterwards, the accumulator of f_i can output the frequency of the predicted pulse $f(t)$. Overall, the velocity frequency calculation device can following the idea of integral and transmit the frequency of the predicted pulse $f(t)$ to the driving pulse generation device so as to generate the driving pulse at the corresponding frequency. Measures machine vibration at different jerk settings to identify resonant frequencies. And transition points are adjusted to avoid excitation of harmful frequencies. Reinforcement learning (RL) optimizes transition points in real-time based on cutting forces and positioning errors. This intricate relationship between pulse synchronization and S-curve profiles continues to drive innovations in motion control, particularly in applications requiring both high speed and ultra-high precision.

4.3 Function of Driving Pulse Generation

The drive pulse generation module is a critical component in motion control systems, responsible for converting digital motion commands into precise electrical pulses that drive servo motors. Phase accumulator was used to ensure smooth frequency transitions during S-curve acceleration/deceleration. The real-time encoder feedback was used to correct pulse timing errors. And adaptive algorithms compensate for mechanical backlash and inertia. The functional device of driving pulse generation adopts the phase-locked loop (PLL) to perform frequency multiplication on the system clock F_{clk} , in which the frequency multiplication coefficient k ranges from $1 \leq k \leq 8$. Accordingly, high-frequency clock can be obtained:

$$F_{clkpll} = k \cdot F_{clk} \quad (11)$$

By storing the frequency of F_{clkpll} , denoted as G , into a 32-bit integer register, and inputting G and $f(t)$ into the divider, the following expression can be obtained:

$$Q = \left\lfloor \frac{G}{f(t)} \right\rfloor \quad (12)$$

where Q denotes the operation result by the divider and equals to the quotient after rounding down. Q is a 32-bit integer and can be used as the frequency-dividing coefficient of the adjustable frequency divider. According to Eq. (12), the frequency of the driving pulse can be written as:

$$f_a(t) = \frac{F_{clkpll}}{Q} = \frac{k \cdot F_{clk}}{\lfloor G / f(t) \rfloor} \quad (13)$$

The driving pulse is the final output from the S-curve acceleration and deceleration control. The pulse then passes through the interpolation control module and pattern selector, and then can be used in the driving motor after phase discrimination and filter in the difference channel.

4.4 Function of the Judgment of Transition Points

The primary challenge in S-curve acceleration and deceleration control lies in accurately determining the transition points within the S-curve profile. To address this, the functional modules incorporate several key components, including an acceleration comparator, a frequency comparator, counters and comparators for both sampling pulses and driving pulses. These components collectively enable precise identification of transition points by evaluating real-time data against predefined thresholds. The output signals $EN1$ and $EN2$ are encoded based on the logical evaluation of various conditions, such as acceleration thresholds, frequency limits, and pulse counts. Specifically, $EN1$ is activated when the acceleration exceeds a predefined upper threshold or when the frequency of sampling pulses reaches a specified limit, and $EN2$ is triggered when the driving pulse count surpasses a designated value or when the system enters a critical phase, such as the transition from acceleration to constant velocity.

(1) Judgment of the transition points in complete S-curve. The complete S-curve maintains continuous jerk (3rd-order continuity) throughout the entire motion and eliminates all abrupt changes in acceleration derivatives. Unlike piecewise S-curves, transition points in a complete S-curve are implicitly defined by critical points in the jerk/acceleration functions. When the number of constant-length pulses within the specified motor stroke is insufficient to accelerate the motor to its maximum driving velocity V_s , the acceleration cannot reach its maximum value, resulting in a triangular profile. In this scenario, the S-curve can be segmented into four distinct phases, comprising three transition points. As illustrated in Fig. 1(b), the velocity profile across different acceleration phases, starting from an initial velocity of zero ($V_0 = 0$), can be described for the four curve segments connected by points α , β and γ as follows.

$$V(t) = \begin{cases} j_a T_s^2 \sum_{i=0}^N i & N = \lceil t / T_s \rceil & 0 < t \leq t_c \\ j_a T_s^2 (\sum_{i=0}^{N_c} i - \sum_{i=0}^N i) + j_a T_s^2 N_c N & N = \lceil \frac{t-t_c}{T_s} \rceil & t_c < t \leq 2t_c \end{cases} \quad (14)$$

Therefore, the output pulse $P(t)$ from the initial moment to t moment equals to the integral with respect to the velocity $V(t)$:

$$P(t) = \begin{cases} j_a T_s^2 \sum_{n=0}^N \sum_{i=0}^N i & N = \lceil t / T_s \rceil & 0 < t \leq t_c \\ j_a T_s^2 \sum_{n=0}^N \sum_{i=0}^{N_c} i + j_a T_s^2 N_c \sum_{n=0}^N N & N = \lceil \frac{t-t_c}{T_s} \rceil & t_c < t \leq 2t_c \end{cases} \quad (15)$$

According to Eq. (15), it can be derived that:

$$P(t_c) / P(t_{2c}) = 1/6 \quad (16)$$

The number of the pulses $P(t_c)$ in the increasing acceleration phase can be written as:

$$P(t_c) = P/12 \quad (17)$$

Therefore, a P_i counter can be used for counting the driving pulses. When Eq. (17) is satisfied and the current acceleration is below the maximum (i.e., $A(t) < A_{max}$), the selector of j_a should select the derivative of acceleration $-j_a$ so as to reduce the current acceleration. Next, the motor enters into the increasing deceleration phase via the comparator of Pt when $P(t) = P/2$. The derivative of acceleration j_a is selected so that motor velocity slowly drops to 0 until $P(t) = 11P/12$. The velocity in the whole phase shows complete S-curve.

(2) Judgment of the transition points in complete S-curve. When the number of constant-length pulses within the specified motor stroke is sufficient to accelerate the motor to its maximum driving velocity V_s , the S-curve can be segmented into seven distinct phases, comprising six transition points. As depicted in Fig. 1(a), the veloc-

ity profile across these phases, connected by points a, b, c, d, e and f , is illustrated. The judgment conditions for these transition points are described in detail in Table 1.

The starting point of the uniform acceleration phase, denoted as a , can be judgment by comparing the current acceleration with A_{max} in the acceleration comparator A_t . When the current acceleration $A(t)$ equals to A_{max} ($A(t) = A_{max}$), the point can be judged as the starting point a . Meanwhile, the pulse at the starting point a is recorded by the P_t counter. The starting point of the decreasing acceleration phase, denoted as b , can be judged by the sampling periods in the increasing and the uniform acceleration phases as calculated in Eq. (2).

$$(N_c + N_d) = (V_s - V_0) / A_{max} T_s \quad (18)$$

Therefore, the number of the trigger pulses T_A is counted with the counter of the sampling pulses N_t , and then transited to the comparator. When $N_t = N_c + N_d$, the point can be judged as the starting point b . Meanwhile, the pulse at the point b , denoted as P_b , is also recorded. In addition, the starting point of the uniform velocity phase can be easily judged. When the $f(t)$ in the frequency f_t comparator reaches f_s , the point can be judged as the point c . Meanwhile, the pulse at the point c , denoted as P_c , is recorded. The starting points of the increasing deceleration, the uniform deceleration and the decreasing deceleration phases, denoted as d, e and f , can be directly judged with the P_t comparator. When $P(t) = P - P_c$, the point can be judged as the starting point of the increasing deceleration phase d . When $P(t) = P - P_b$, the point can be judged as the starting point of the uniform deceleration phase e . When $P(t) = P - P_a$, the point can be judged as the starting point of the decreasing phase f .

According to above judgment conditions, the electrical levels of $EN1$ and $EN2$ are encoded. Next, different values of j_a can be output by decoding the selector of the derivative of acceleration j_a in the functional device of velocity frequency calculation. Table 1 lists the coding and decoding values of $EN1$ and $EN2$.

Table 1. Coding and decoding values of $EN1$ and $EN2$

| Type | Segment | Point | Condition | $EN1$ | $EN2$ | Output |
|-------------------|-------------------------------|----------|------------------------------------|-------|-------|--------|
| Complete S-curve | Increasing acceleration phase | V_0 | Initial value | 0 | 0 | j_a |
| | Decreasing acceleration phase | α | $P(t) = P/12$ and $A(t) < A_{max}$ | 0 | 1 | $-j_a$ |
| | Increasing deceleration phase | β | $P(t) = P/2$ | 0 | 1 | $-j_a$ |
| | Decreasing deceleration phase | γ | $P(t) = 11P/12$ | 1 | 0 | j_a |
| 7-segment S-curve | Increasing acceleration phase | V_0 | Initial value | 0 | 0 | j_a |
| | Uniform acceleration phase | a | $P(t) < P/12$ and $A(t) = A_{max}$ | 1 | 1 | 0 |
| | Decreasing acceleration phase | b | $N_t = N_c + N_d$ | 0 | 1 | $-j_a$ |
| | Uniform velocity phase | c | $V(t) = V_s$ i.e., $f(t) = f_s$ | 1 | 1 | 0 |
| | Increasing deceleration phase | d | $P(t) = P - P_c$ | 0 | 1 | $-j_a$ |
| | Uniform deceleration phase | e | $P(t) = P - P_b$ | 1 | 1 | 0 |
| | Decreasing deceleration phase | f | $P(t) = P - P_a$ | 1 | 0 | j_a |

5 Experimental Validations

In the current experiment, the integrated machine control system is equipped with a touch screen and a human-machine interface (MD304L, Kinco). The system also incorporates motors with a rated power of 750 W and a rated rotational speed of 3000 rpm, paired with a compatible driver (Minas A4, Panasonic). The FPGA utilized in this setup is the EP4CE30F23C6 chip from ALTERA, featuring 28,848 logic units, 328 pins, and 4 PLL IP cores, which adequately meets the motion controller's requirements for multi-axis motor control. The established experimental platform is illustrated in Fig. 4 [26-28]. A simplified numerical control system was subsequently developed to experimentally validate the aforementioned S-curve acceleration and deceleration control module [29]. FPGAs are ideal for implementing S-curve control due to their parallel processing, deterministic latency, and hardware-level pulse generation. FPGAs calculate jerk, acceleration, and velocity transitions in nanosecond-scale latency. If vibration sensors detect instability, the FPGA can dynamically reconfigure the jerk profile

without stopping the machine. FPGAs synchronize transition points across axes using hardware-timed triggers, eliminating skew errors in contouring applications.

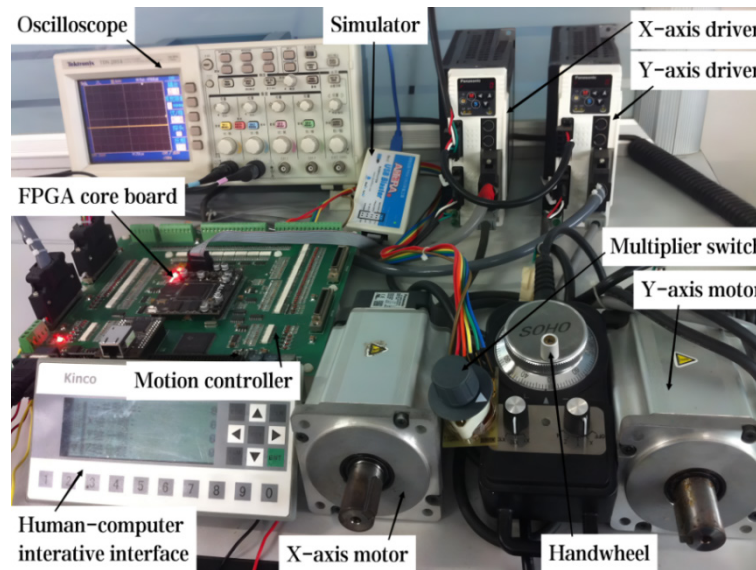


Fig. 4. Established numerical-control experimental platform

The system clock of the FPGA core board in the motion controller (F_{clk}) was configured at 50 MHz, with the PLL multiplier coefficient (k) set to 8. The system parameters were defined via the MD304L interface as follows: the frequency dividing coefficient (D_f) was set to $5k$. Based on Eq. (4), the sampling period (T_s) was calculated to be 0.1 ms; the derivative of acceleration (j_a) was set to 10. The maximum acceleration (A_{max}) was set to $1k$, corresponding to the maximum pulse frequency increment (Δf) in Hz. The initial velocity (V_0) was interpreted as the value at an initial frequency (f_0) of 1 kHz, and the maximum driving velocity (V_s) interpreted as the value at a maximum frequency (f_s) of 20 kHz.

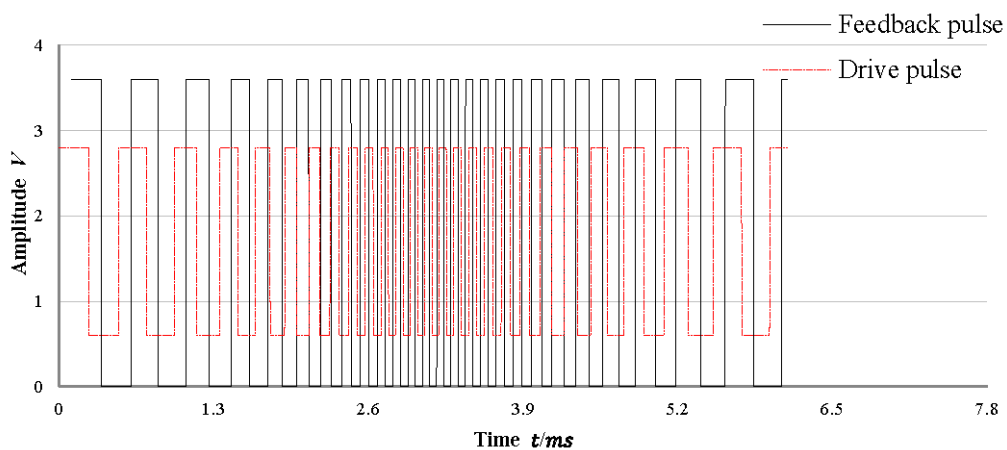


Fig. 5. Illustration of the driving pulse and the feedback signal

(1) The results when the number of the constant-length pulses with the input motor stroke was 25. The experiment started, and the output driving pulse was compared with the modulated signal of Position feedback mod-

ule, as shown in Fig. 5. Overall, actual feedback pulse (marked in solid line) and the driving pulse (marked in dash line) show coincident variation tendencies but lagged by 0.1 ms. In a sampling period, the numbers of two pulses were 25. Fig. 6(a) shows the frequency of the modulated signal of Position feedback module. Apparently, ladder-pattern waveform can be observed. After fitting, it can be found that the fitted curve conformed to complete S-curve characteristics. To be specific, the acceleration process and the deceleration process lasted 2.63 ms and 2.84 ms, respectively. The difference equaled to 0.21 ms, which was subjected to the initial velocity V_0 . In addition, the motor was accelerated to the maximum frequency 8.34 kHz. By observing the position curve of the modulated signal of Position feedback module, 25 pulses lasted 5.47 ms. As shown in Fig. 6(b), the positive curve showed S-type variation. The number of the pulses when the motor travelled was identical to the number of the set constant-length pulses P . Conclusively, when the input number of the driving pulses P cannot accelerate the motor to the maximum driving velocity V_s , the acceleration and deceleration control module can perform motion control according to complete S-curve acceleration and deceleration control algorithm.

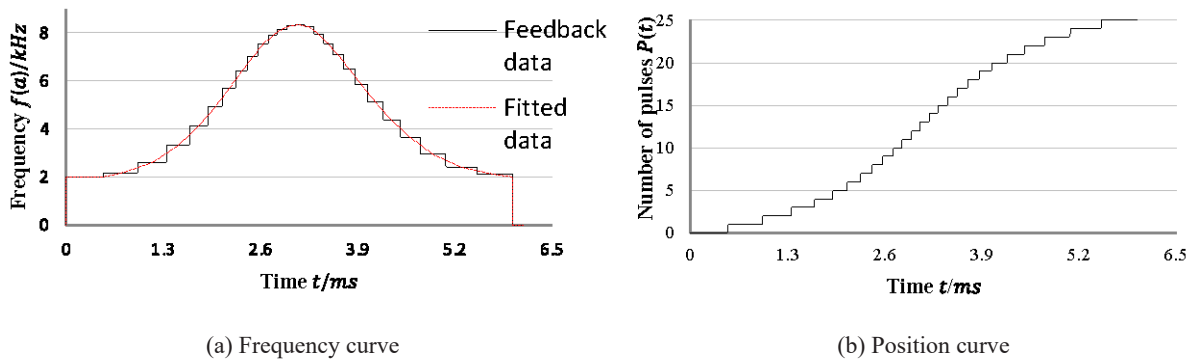


Fig. 6. Analysis of the feedback pulse data in the first experiment

(2) When the number of the constant-length pulses P changed to 100, the frequency curve and the position curve of the modulated signal of Position feedback module were plotted for analysis, as shown in Fig. 7. When the number of the driving pulses can enhance the velocity to the maximum driving velocity (i.e., V_s at a maximum frequency of 20kHz), the plotted frequency curve showed 7-segment S-curve characteristics. The acceleration process and the deceleration process lasted 4.12 ms and 4.18 ms, respectively, with a difference of only 1.2 ms. At that moment, the velocity was slightly affected by the initial velocity. In addition, it can be observed from the position curve that the number of the pulses that the motor operated equals to then umber of the constant-length driving pulses, showing no out-of-step phenomenon.

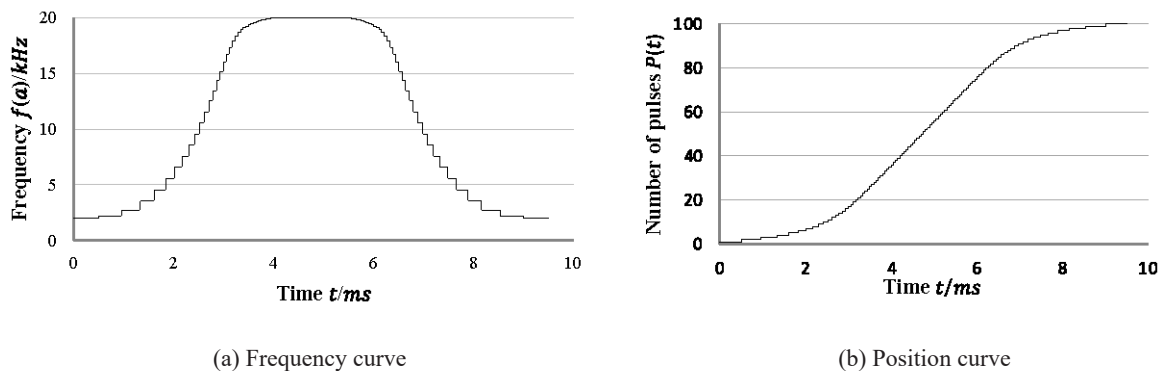


Fig. 7. Analysis of the feedback pulse data in the second experiment

(3) By changing the number of the constant-length driving pulses to P , the feedback data of the motor encoder was stored to the address bit 0X0090C0 in DSP. The data were directly red, and the corresponding positive curve

was plotted and displayed in Code Composer Studio (CCStudio). Fig. 8(a) and Fig. 8(b) display the position curves when the number of the constant-length driving pulses P were 1000 and 10000, respectively. Both two positive curves showed S-type pattern. The number of the pulses that the motor operated was identical to the number of the set constant-length driving pulses P .

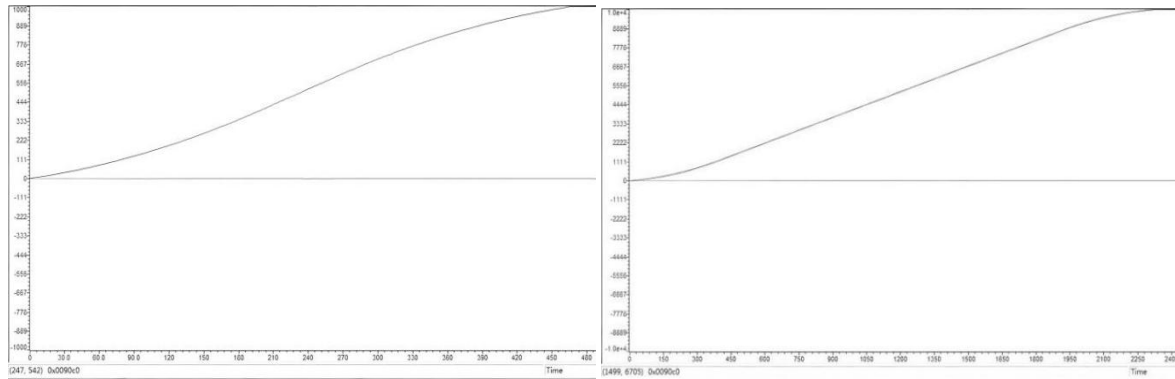
(a) $P=1000$ (b) $P=10000$

Fig. 8. Position data of the feedback pulses displayed in CCStudio

6 Conclusions

1) A control algorithm based on FPGA with S-curve acceleration and deceleration was developed, enabling high-precision and high-frequency output. Experimental results confirmed the effectiveness of the proposed method, particularly in enhancing motor operating efficiency and system reliability.

2) The S-curve stages and transition points were accurately determined using functional devices such as acceleration comparators, frequency comparators, and counters, which monitor real-time changes in motor feedback speed and acceleration. Leveraging the FPGA's high parallel processing capability and reconfigurability, high-frequency sampling and rapid computation were achieved, meeting the real-time and accuracy requirements of CNC systems.

3) Experimental findings demonstrate that the FPGA-based control algorithm significantly improves motor efficiency and reliability. The acceleration and deceleration phases were notably shortened without any instances of motor step loss. Additionally, the feedback pulse frequency and position curve exhibited smooth profiles without inflection points, further validating the robustness of the proposed approach.

Acknowledgement

This work was financially supported by the scientific research project from the National Industry and Information Technology Vocational Education and Teaching Steering Committee (GXHZWZ11931) and the scientific research project from the Education Department of Zhejiang Province, China (Y202043150).

References

- [1] J.C. Wu, K. Xu, G.G. Ren, D.P. Fan, Research on the S-shaped time-rounding series feedrate scheduling based on NURBS curve, *Advances in Mechanical Engineering* 14(9)(2022) 1-14.
<https://doi.org/10.1177/16878132221121479>
- [2] X. Yang, Y. Ran, G.B. Zhang, H.W. Wang, Z.Y. Mu, S.G. Zhi, A digital twin-driven hybrid approach for the prediction of performance degradation in transmission unit of CNC machine tool, *Robotics and Computer-Integrated Manufacturing* 73(2022) 102230.

- <https://doi.org/10.1016/j.rcim.2021.102230>
- [3] H.M. Chang, S. Rezvani, J. Lee, S.S. Park, H.W. Park, J. Lee, Indirect measurement of cutting forces during robotic milling using multiple sensors and a machine learning-based system identifier, *Journal of Manufacturing Processes*, 85(2023) 963-976.
<https://doi.org/10.1016/j.jmapro.2022.12.019>
 - [4] W.S. Xie, J. Zhou, T. Liu, Blind Fault Extraction of Rolling-Bearing Compound Fault Based on Improved Morphological Filtering and Sparse Component Analysis, *Sensors (Basel, Switzerland)* 22(18)(2022) 7093.
<https://doi.org/10.3390/s22187093>
 - [5] K. Ishizaki, E. Shamoto, A new real-time trajectory generation method modifying trajectory based on trajectory error and angular speed for high accuracy and short machining time, *Precision Engineering* 76(2022) 173-189.
<https://doi.org/10.1016/j.precisioneng.2022.02.012>
 - [6] N.H. Qian, N.R. Zhou, Precision Machining Technology of Jewelry on CNC Machine Tool Based on Mathematical Modeling, *Applied Mathematics and Nonlinear Sciences* 8(1)(2023) 611-620.
<https://doi.org/10.2478/amns.2022.2.0046>
 - [7] L.J. Yu, T.Y. Zhang, H.L. Tian, Z.J. Yang, A. Liu, F.Y. Zhang, FMECA of CNC machine tool design stage based on CBWM and DEA, *Quality and Reliability Engineering International* 40(1)(2024) 154-169.
<https://doi.org/10.1002/qre.3312>
 - [8] H.M. Luo, D.B. Zhao, W.Q. Fu, Speed Planning Algorithm Based on Improved S-Type Acceleration and Deceleration Model, *Journal of Shanghai Jiaotong University (Science)* 26(6)(2021) 786-793.
<https://doi.org/10.1007/s12204-021-2322-4>
 - [9] Q.X. Zhu, Y.S. Jin, Y.H. Zhu, An optimized cosine jerk motion profile with higher efficiency and flexibility, *Assembly Automation* 42(3)(2022) 350-360.
<https://doi.org/10.1108/AA-11-2021-0165>
 - [10] D.Y. Yu, Z. Ding, X.Q. Tian, Incomplete smooth S-curve acceleration and deceleration feedrate planning modeling and analysis, *International Journal of Advanced Manufacturing Technology* 120(11-12)(2022) 7171-7185.
<https://doi.org/10.1007/s00170-022-09236-7>
 - [11] G.R. Wang, Q. Wang, X.H. Cui, S. Xu, S.W. Zhang, An interpolation algorithm of B-spline curve based on S-curve acceleration/deceleration with interference pre-treatment, *International Journal of Computer Applications in Technology* 66(1)(2021) 51-62.
<https://doi.org/10.1504/IJCAT.2021.119607>
 - [12] X. Han, X.K. Zhang, H.G. Zhang, Trajectory Planning of USV: On-Line Computation of the Double S Trajectory Based on Multi-Scale A* Algorithm with Reeds–Shepp Curves, *Journal of Marine Science and Engineering* 11(1)(2023) 153.
<https://doi.org/10.3390/jmse11010153>
 - [13] M. Yessenov, A.F. Abouraddy, Accelerating and Decelerating Space-Time Optical Wave Packets in Free Space, *Physical review letters* 125(23)(2020) 233901.
<https://doi.org/10.1103/PhysRevLett.125.233901>
 - [14] A. Alhomoud, S.S. Jamal, S.M. Altowaijri, M. Ayari, A.R. Alharbi, A. Aljaedi, Large field-size throughput/area accelerator for elliptic-curve point multiplication on FPGA, *Applied Sciences* 13(2) (2023) 869.
<https://doi.org/10.3390/app13020869>
 - [15] B.K. Wang, S.H. Wang, Y.B. Peng, Y.G. Pi, Y. Luo, Design and high-order precision numerical implementation of fractional-order PI controller for PMSM speed system based on FPGA, *Fractal & Fractional* 6(4)(2022) 218.
<https://doi.org/10.3390/fractalfract6040218>
 - [16] Y.H. Xiong, K. Li, Z.T. Liu, J.H. She, Design and implementation of the universal servo control algorithm verification system based on high-speed communication fieldbus, *Journal of Advanced Computational Intelligence & Intelligent Informatics* 25(2)(2021) 248-257.
<https://doi.org/10.20965/jaciii.2021.p0248>
 - [17] H. Zhu, M.L. Wang, K. Liu, W.Y. Xu, NURBS modeling and curve interpolation optimization of 3D graphics, *Computers, Materials & Continua* 66(2)(2021) 1799-1811.
<https://doi.org/10.32604/cmc.2020.012706>
 - [18] S. Yaramyshev, W. Barth, S. Lauber, M. Miski-Oglu, A. Rubin, U. Scheeler, H. Vormann, M. Vossberg, DYNAMION—A Powerful Beam Dynamics Software Package for the Development of Ion Linear Accelerators and Decelerators, *Applied Sciences* 13(14)(2023) 8422.
<https://doi.org/10.3390/app13148422>
 - [19] L. Gong, Y.X. Wu, B.S. Gao, Y.F. Sun, X.Y. Le, C.L. Liu, Real-Time Dynamic Planning and Tracking Control of Auto-Docking for Efficient Wireless Charging, *IEEE Transactions on Intelligent Vehicles* 8(3)(2023) 2123-2134.
<https://doi.org/10.1109/TIV.2022.3189511>
 - [20] Y.F. Zhang, F.Z. Fang, W. Huang, W. Fan, Dwell Time Algorithm Based on Bounded Constrained Least Squares Under Dynamic Performance Constraints of Machine Tool in Deterministic Optical Finishing, *International Journal of Precision Engineering and Manufacturing-Green Technology* 8(5)(2021) 1415-1427.
<https://doi.org/10.1007/s40684-020-00306-3>

- [21] F.G. Zacchigna, Methodology for CNN Implementation in FPGA-Based Embedded Systems, *IEEE Embedded Systems Letters* 15(2)(2023) 85-88.
<https://doi.org/10.1109/LES.2022.3187382>
- [22] L. Cambuim, E. Barros, FPGA-Based Pedestrian Detection for Collision Prediction System, *Sensors* 22(12)(2022) 4421.
<https://doi.org/10.3390/s22124421>
- [23] D.Q. Phan, H.Q. Thinh Ngo, Enhanced performance of the mechanical respiratory system by FPGA-digital based on PID Controller, *Cogent Engineering* 10(1)(2023) 2247865.
<https://doi.org/10.1080/23311916.2023.2247865>
- [24] A. Przybył, FPGA-Based Optimization of Industrial Numerical Machine Tool Servo Drives, *Electronics* 12(17)(2023) 3585.
<https://doi.org/10.3390/electronics12173585>
- [25] P.K. Pathak, A.K. Yadav, A. Shastri, BWOA based metaheuristic approach for uncertain nonlinear milling CNC machine system, *Expert Systems* 40(6)(2023) e13233.
<https://doi.org/10.1111/exsy.13233>
- [26] B.Q. Liu, H.M. Zhang, Y. Liu, M.M. Lu, A Feedrate Planning Method in CNC System Based on Servo Response Error Model, *Electronics* 12(14)(2023) 3150.
<https://doi.org/10.3390/electronics12143150>
- [27] Y.F. Wu, N. Yue, K.K. Qian, Performance Optimization of CNC Machine Tool System Based on Sensor Data, *Scientific Programming* 2022(2022) 5663824.
<https://doi.org/10.1155/2022/5663824>
- [28] J.G. Li, C.G. Qi, Y.N. Li, Z.H. Wu, Prediction and Compensation of Contour Error of CNC Systems Based on LSTM Neural-Network, *IEEE/ASME Transactions on Mechatronics* 27(1)(2022) 572-581.
<https://doi.org/10.1109/TMECH.2021.3068354>
- [29] M.-A. Cabrera-Rufino, J.-M. Ramos-Arreguín, J. Rodríguez-Reséndiz, E. Gorrostieta-Hurtado, M.-A. Aceves-Fernandez, Implementation of ANN-Based Auto-Adjustable for a Pneumatic Servo System Embedded on FPGA, *Micromachines* 13(6)(2022) 890.
<https://doi.org/10.3390/mi13060890>