ISSN 1454 - 8003
Proceedings of 2022 International Conference on Hydraulics and Pneumatics - HERVEX
November 9-10, Băile Govora, Romania

S-CURVE MOTION CONTROL IMPLEMENTATION USING 32-BIT MICROCONTROLLER

Andrei DRUMEA[1], Cristina Ioana MARGHESCU[1], Mihaela PANTAZICĂ[1], Gheorghe JITIANU[1], Andrei VLAD[2]

[1] Politehnica University of Bucharest, andrei.drumea@upb.ro

[2] S.C. HESPER S.A.

Abstract: *The paper presents the design, implementation and testing of a stepper motor controller with soft-start function based on 32-bit microcontroller. The soft-start is implemented using the S-curve due to its simplicity and its ability to control the maximum values of speed and acceleration, key parameters for many mechatronic systems. Theoretical aspects are briefly presented, as well as software and hardware solution and its testing and measurement on a specific test bench.*

Keywords: *S-curve, stepper motor, microcontroller*

1. Introduction

Many mechatronic applications require fast and precise movement of actuators from one point to another. Shocks in the actuator mechanism should be avoided and a smooth operation is preferred; these requirements are harder to meet at the start of the motion, especially in the case of actuators implemented with stepper motors that have movement divided in discrete parts. Soft-start operation is a good solution for such systems, and this technique involves slower movement at the beginning, with controlled (limited) speed and acceleration.
A good implementation of soft-start operation is the S-curve, which offers limited speed, acceleration and jerk ([1]). At the start of the motion, the speed is slow but gradually increases up to a maximal value; near the end of the movement, the speed starts to decrease in the same way. In typical mechanical movements, a 0.5 seconds timespan for the S-curve profile is a good value. Considering this parameter, the S-curve can be obtained from trigonometric functions, as follows in formula (1):

$$y(t) = \int_0^t sin^2(x)dx \tag{1}$$

This particular function offers S-type curve for a 0.5 seconds timespan. Calculating the integral, we can obtain the simplified equation for the S-curve, as formula (2):

$$y(t) = \frac{2}{\pi}\left(\pi t - \frac{1}{4} sin(4\pi t)\right) \tag{2}$$

This formula can be used by a microcontroller to generate a pulse train with variable period for driving a stepper motor; an integrated timer module simplifies this task. The period of the pulse can be calculated on the fly (for each pulse the microcontroller calculates its period, performing 4 multiplications and one difference) or, for improved speed but with larger memory consumption, this function can be tabulated (calculated before using a mathematical software and stored as a vector in microcontroller program memory). Because modern microcontrollers offer large program memory space, the second method is preferred due to its speed and easier software implementation (the program parses the pulse period vector and sets the timer to generate the pulse accordingly).

The S-curve described by equation (2) for a 0.5 seconds timespan is shown in figure 1.

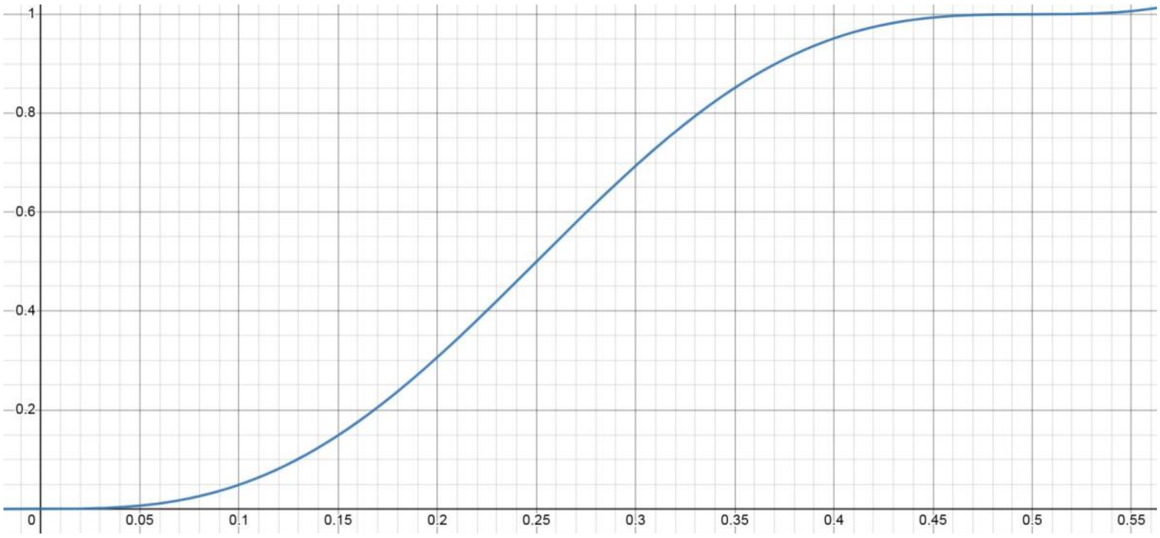

Fig. 1. Proposed S-curve with 0.5 seconds timespan

This curve will be used to generate variable period/frequency pulses for a stepper motor driver from lower frequencies (start frequency is in 100Hz...1kHz range) up to 30kHz.

2. Software Implementation

An Object-Pascal based Windows application was developed to generate the vector values for the presented S-curve. The application, shown in figure 2, generates the vector of configuration values for a microcontroller timer programmed for a 0.25microsecond ticks.

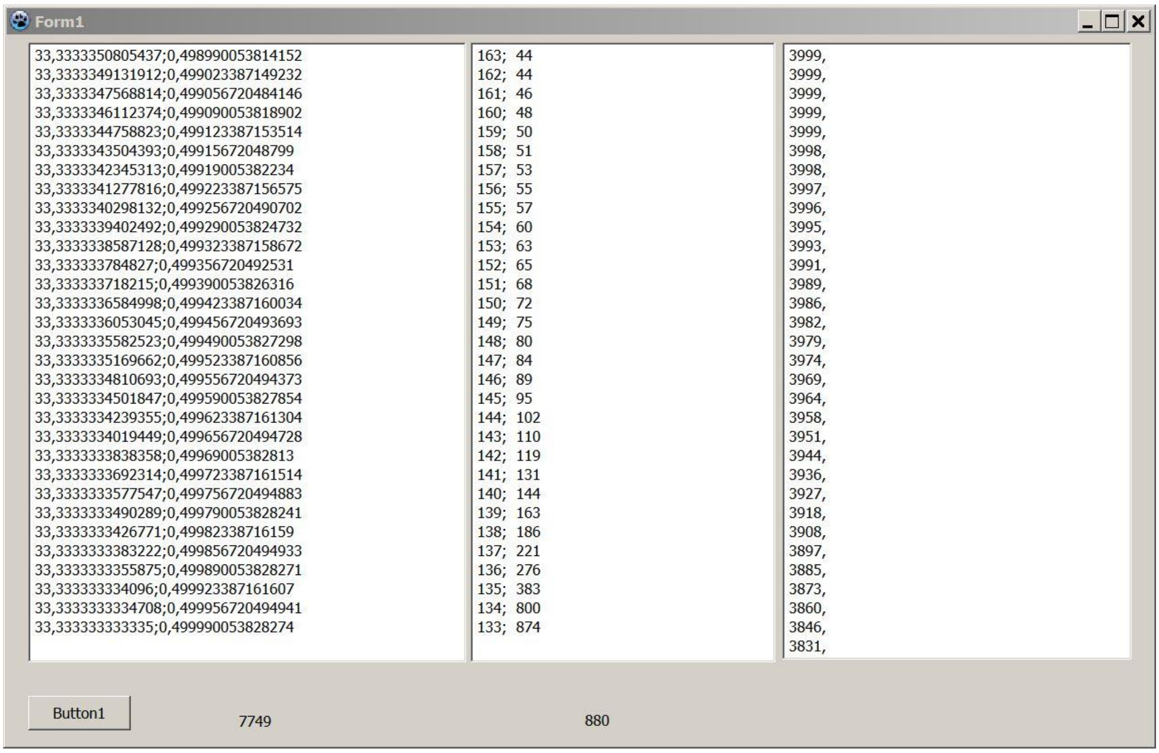

Fig. 2. Main window of Windows application for generating vector values for S-curve

This 0.25 microsecond value can be precisely generated using a standard 16bit timer of the STM32F205 microcontroller. The microcontroller is clocked at 120MHz so it has enough power for industrial process control and enough granularity for timer.

In the application, current pulse period (in microseconds) and time (in seconds) are stored on the left memo box; in the middle memo box is shown a statistics of values for the timer configuration (the value and the number of times it repeats), and the right memo box is the vector itself, that can be pasted into C source code of the microcontroller program. The bottom labels contain the total number of vector values and the number of different values. This information is useful when defining vector size or for other optimizations.

The generated S-curve values for frequencies between 100Hz and 30kHz are shown in figure 3.

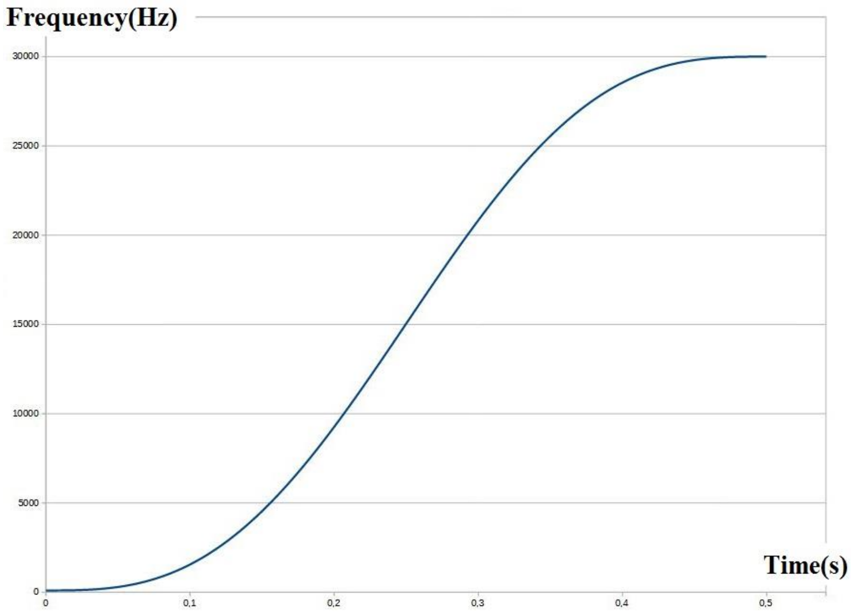

Fig. 3. Calculated S-curve for frequency sweep from 100Hz to 30kHz in 0.5 seconds

Figure 4 presents pulse width, which decreases from 10 miliseconds to 33.3 microseconds.

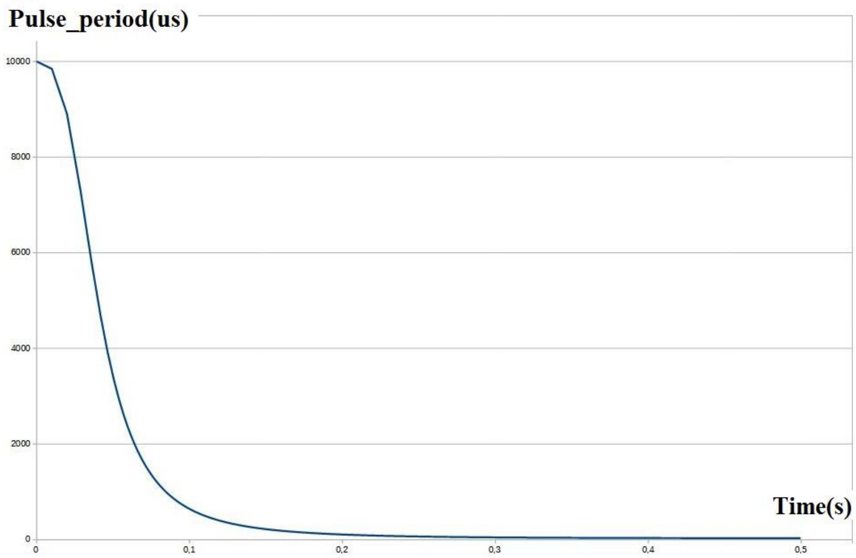

Fig. 4. Calculated period of pulses for frequency sweep from 100Hz to 30kHz in 0.5 seconds

Different vectors can be generated for other start frequencies (200Hz, 333Hz, 500Hz or 1kHz) and can be grouped in a matrix for more flexibility; each vector element requires 2 bytes of flash memory, about 16 kBytes for the entire vector ([4]). This memory requirement is not a problem for STM32F205 microcontroller that has a flash memory of 256kBytes...1MByte, depending on chosen chip version.
In our application we use an array of 8 S-curve vectors, but the microcontroller has enough memory for more vectors (up to 32).

3. Test and Measurements

The S-curve software was written in C language for ARM Cortex-M processors (STM32F205 belongs to this category) and implemented in an electromechanical system for electric discharge machining. This system moves the electrode for electric discharge using an actuator with stepper motor and rotary encoder, as shown in figure 5.

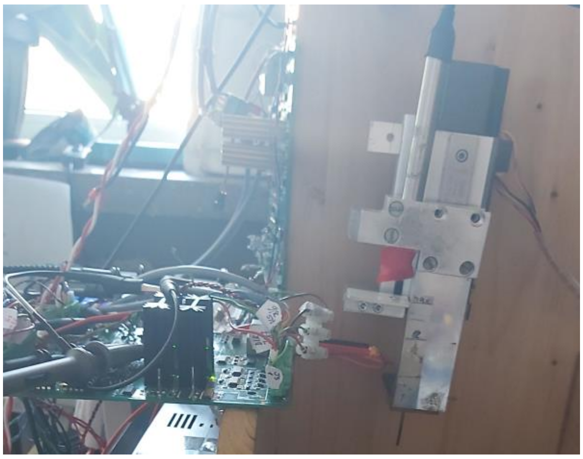

Fig. 5. System view, microcontroller board and actuator with stepper motor, rotary encoder and LVDT sensor

A LVDT displacement sensor ([3]) is fixed to the actuator to measure the real displacement and to observe if any motor steps are missing. The microcontroller board, shown on the left side, is detailed in figure 6.

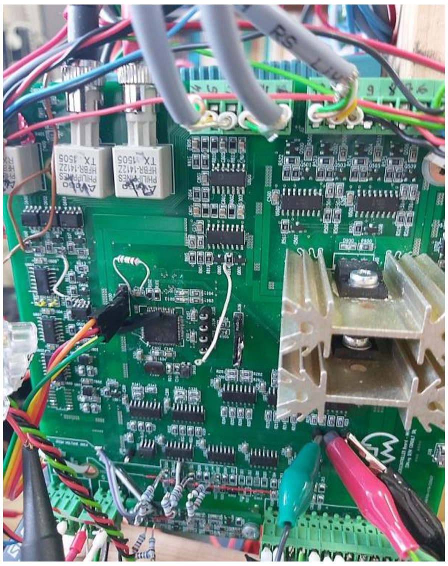

Fig. 6. Microcontroller board, detailed

The microcontroller board uses many optocouplers to isolate the digital command signals from the power signals ([2]); there is also a serial connection to a programmable logic controller (PLC) that offers human to machine interface using a touch panel.
The following pictures present the generated pulses for stepper motor driver (blue traces), measured displacement from LVDT sensor (green traces) and logic commands from master programmable logic controller – stop command (magenta traces) and slow command (orange traces). These signals were observed during an up-down periodic movement with 0.5 seconds pause at the end. Noise can be observed at the LVDT sensor output, due to the electric discharge process running in the same time.

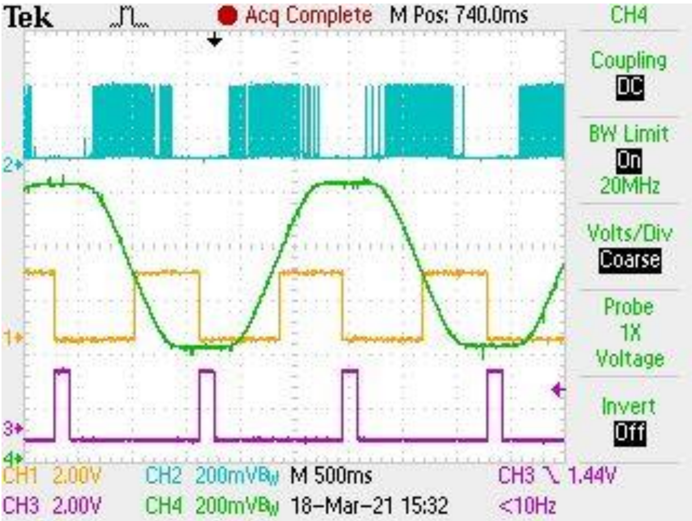

Fig. 7. Pulses from stepper motor controller for an up-down movement

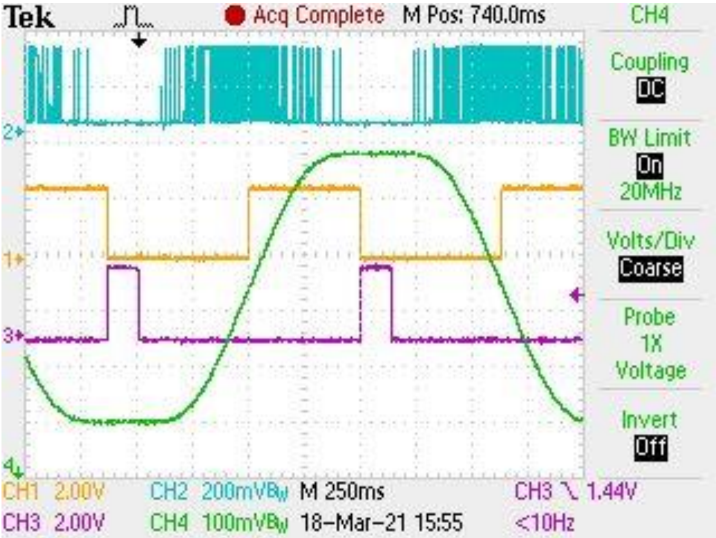

Fig. 8. Pulses from stepper motor controller for an up-down movement, detailed

Presented traces for stepper motor traces have lower pulse density at the start and the end of the motion and higher density in the middle, but the limited number of memory points of the digital oscilloscope as well as aliasing affects the display of the trace when trying to show the pulses on a larger timespan.
Nevertheless, the system works as expected and each pulse has the correct timing.

4. Conclusions

Soft-start option is an important feature in mechatronic systems such as stepper motor based actuators because it reduces the number of lost steps at the beginning of the motion. Soft-start is usually implemented as a variable speed with an S-curve profile that can be easy implemented using a 32-bit microcontroller with enough memory resources. Our implementation, based on STM32 microcontroller from ST Microelectronics, was tested and fulfilled its requirements.

Acknowledgments

The research presented in this paper has been developed under Financial Agreement no. 272/24.06.2020, signed by the Ministry of European Funds / Ministry of Education and Research and S.C. HESPER S.A. Bucharest for the Innovative Technological Project titled "Digital mechatronic systems for generating pressure of 1000 bar, using hydraulic pressure intensifiers" (SMGP), project under implementation from 01.07.2020 to 30.06.2023. Financial support has also been granted under a project funded by the Ministry of Research, Innovation and Digitalization through Programme 1- Development of the national research & development system, Sub-programme 1.2 – Institutional performance – Projects financing the R&D&I excellence, Financial Agreement no. 18PFE/30.12.2021.

References

[1] Blejan, Marian, and Robert Blejan. "Mathematics for Real-Time S-Curve Profile Generator." *Hidraulica Magazine*, no. 4 (October 2020): 7-25.
[2] Drumea, Andrei, and Robert Dobre. "Analysis of power supply circuits for electroluminescent panels." Paper presented at the Advanced Topics in Optoelectronics Microelectronics and Nanotechnologies VIII ATOM-N2016, Constanta, Romania, August 25-28, 2016.
[3] Drumea, Andrei, Marian Blejan, and Ciprian Ionescu. "Differential inductive displacement sensor with integrated electronics and infrared communication capabilities." Paper presented at the Advanced Topics in Optoelectronics Microelectronics and Nanotechnologies VI ATOM-N2012, Constanta, Romania, August 23-26, 2012.
[4] Drumea, Andrei. "Low power aspects of a microcontroller-based module with wireless communication." Paper presented at the 23rd International Symposium for Design and Technology in Electronic Packaging SIITME2017, Constanta, Romania, October 26-29, 2017.