electronics

MDPI

Article

Digitizing Low-Frequency Analog Control Circuit Using Bilinear Function Algorithm

Hsiung-Cheng Lin * and Zong-You Chen

Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung City 411030, Taiwan; s4b113036@student.ncut.edu.tw

* Correspondence: hclin@ncut.edu.tw

Abstract: In the past, low-frequency analog control circuits were largely used in analog control systems. With the rapid development of modern digital electronic technology, the digitization of traditional low-frequency analog control circuits while retaining the same functions as the original analog control systems has become an important trend in the field of electronic design. For this reason, this study aimed to develop an analog signal digitization model for control signals at a low frequency below 10 Hz. In terms of signal receiving and transmission requirements, the proposed model is configured with analog-to-digital converter (ADC), digital-to-analog converter (DAC), and pulse width modulation (PWM) functions. In the development environment, the input and output signals are first normalized and processed with PWM, enabling the digital signal processor to deal with analog signal receiving, processing, and external transmission. To replace the existing compensator in analog circuits, a digital compensator is used in the digital signal processor. Based on the bilinear function in MATLAB software, the parameter values demanded for the digital compensator can thus be obtained, achieving the automatic calculation of bilinear transformation in the system. Multisim simulation is then used to simulate analog circuit systems for comparison with the digitized outcomes. The experimental results reveal that the performance of the designed digital control circuit matches the simulation outcomes in both Bode gain (dB) and phase responses when the signal frequency is below 10 Hz. The effectiveness of the digitized analog control circuit for low-frequency control signals is therefore confirmed.

Keywords: ADC; bilinear function; DAC; digitization; PWM

check for updates

Academic Editor: Padmanabhan Balasubramanian

Received: 14 November 2024
Revised: 30 December 2024
Accepted: 8 January 2025
Published: 10 January 2025

Citation: Lin, H.-C.; Chen, Z.-Y. Digitizing Low-Frequency Analog Control Circuit Using Bilinear Function Algorithm. *Electronics* **2025**, *14*, 273. https://doi.org/10.3390/electronics14020273

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past, analog control hardware circuits were widely used in mechanical or electronic facilities in industry [1]. Since the mid-20th century, however, micro-processors and digital signal processors have become major control cores due to their rapid computational and programming capabilities [2]. The control accuracy, flexibility, and expansion have therefore been significantly enhanced, covering a wide range of areas such as aviation, machines, auto industry, and robotics [3,4]. Traditional control systems rely on analog signal processing techniques such as analog baseband spectrum sensing circuits, analog front-end circuit design, and digital-to-analog converters [5–7]. With the rapid electronic technology enhancement, digital circuit systems are currently leading the major developments in system design, covering a wide range of applications in various areas. For example, a 6-bit binary current-controlled D/A converter was proposed in [8]. The digitally controlled oscillator was constructed as a combination of a voltage-controlled oscillator

and a digital-to-analog converter (DAC). All inputs need to be fed to differential switches consisting of a pair of NMOS transistors. This implies that a more complex hardware circuit should be erected in the system, meaning that a high cost to develop it is required. Another example is the design of a PCB hosting a digital-to-analog converter based on an FPGA development board [9]. External trigger and clock inputs are demanded to synchronize the DACs and multiple FPGAs. It was shown that this design is superior in terms of resistance to digital and analog noise by separating ground planes to prevent ground loops, achieving high-precision, low-noise power supplies and voltage reference circuits. Apart from software development, the hardware requirements, such as FPGAs, breakout boards, and DAC boards, make the system more complex in design and implementation. Alternatively, a low-power, high-resolution, small-area digitally controlled oscillator has been designed with CMOS technology [10]. The proposed ring oscillator architecture with four stages of delay cells can operate with a 6.10 kHz resolution, where the power consumption only varies between 0.6 mW and 1.6 mW for a supply voltage of 1.2 V. However, a 16-bit digitally controlled oscillator has been developed, being specially designed to work within 1.8 GHz to 2.2 GHz. As shown above, it is obvious that it is indispensable to work on analog–digital system conversion or transformation to meet various application requirements in industry [11–13].

In recent years, traditional control systems have usually relied on an analog circuit for design and operation. Although digital control systems are now at the forefront of modern electronic technology, the algorithms developed for the transformation of analog systems into digital systems still play a crucial role in digital system design. In a control system, a digital filter has higher accuracy and better programming capabilities compared with an analog filter. Accordingly, digital filters are regarded as leading signal processing structures in digital control system design [14]. To ensure the consistency of characteristics and responses between digital systems and analog systems in the analog digitization process, the determination of the digital filter structure, such as Direct Form I or Direct Form II, is extremely important for the system stability and reliability [15]. It has been verified that the Direct Form II filter structure is superior to Direct Form I in IIR filter design as it consumes less area and has a lower number of lines and circuit connections [16]. In circuit design, Multisim can provide an overall development environment simulation and test [17]. On the other hand, MATLAB, which provides packages such as Simulink and Control System Toolbox, is suitable for system mathematic modeling and analysis [18]. To verify the effectiveness of the system performance, the Bode response is a key criterion for evaluation [19]. In addition, a bilinear function is used in digital signal processing and discrete-time control theory to transform continuous-time system representations to discrete-time representations. As a result, its transformation process is highly effective in analog-to-digital transformation or digital control design [20–23]. In this study, the bilinear function supported by MATLAB is applied to provide the parameters of a digital compensator. Therefore, it can efficiently avoid aliasing distortion during the digitalization process from the s domain to the z domain [24,25]. The original system characteristics and response of the continuous-time filter in the transformed discrete-time filter can thus be retained [26].

As shown in the above literature, it is suggested that analog control systems be transformed into digital systems to directly connect to modern digital systems. With this motivation, in this study, a digitized circuit using a bilinear function algorithm based on a traditional low-frequency analog control circuit was developed. The merit is that this model can make the design of new digital control circuits simpler and more reliable without additional system design and testing. Section 2 introduces the system structure. Section 3 presents the digitization process for analog control circuit transformation. The

experimental results and analysis are provided in Section 4. The conclusions and future work are presented in Section 5.

2. System Structure

The proposed hardware structure for the digitization process is shown in Figure 1. Here, an oscilloscope is used to measure the signal waveforms and Bode plot response. The input command (CMD) is sent to a digital signal processor (DSP), which contains ADC, DAC, and PWM modules. Switches in the H-Bridge circuit are driven by modulated PWM signals. An electromagnetic interference (EMI) filter connected to the load is used to filter out the noise in the system circuit. The load current is sensed using a current sensor in series with the load, and it is then transmitted to the DSP for signal processing. More details are provided below.

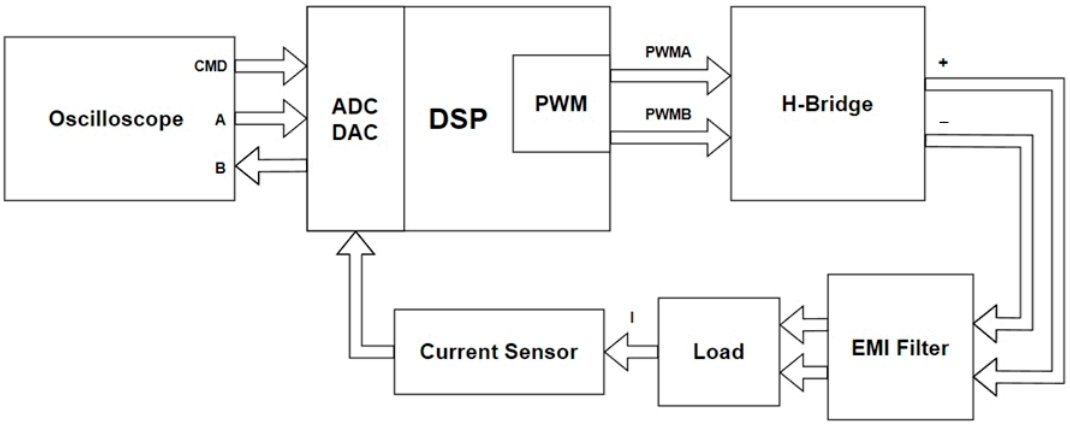


Figure 1. Hardware structure of the digitized system.

(1) Oscilloscope

The PICO SCOPE 2205A, as shown in Figure 2, is used as the oscilloscope, where terminal A is connected to the measured object input, terminal B is connected to the measured object output, and AWG can be used as either the input of the measured object or the output of the signal generator. Therefore, the measured waveforms and Bode response can be observed. In addition, various waveforms, e.g., square, sine, and triangular waveforms, under adjustable frequency, magnitude, output bias, etc., can be generated.



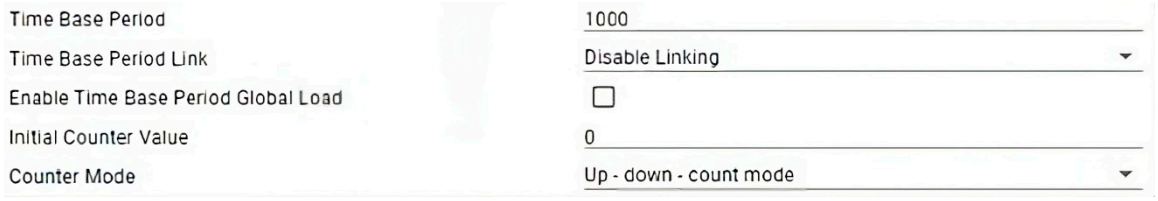
Figure 2. Entity diagram of the PICO SCOPE 2205A.

(2) Digital signal processor

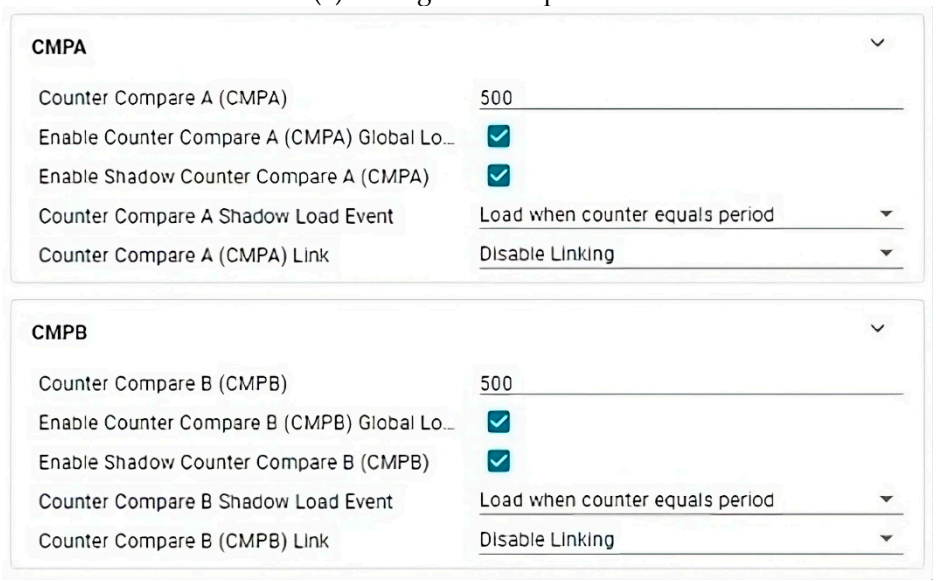
A dual-core real-time microcontroller, i.e., the TMS320F28379D produced by Texas Instruments, Dallas, TX, USA, is used as the DSP. It has three major modules, namely PWM, ADC, and DAC, meeting the fundamental requirements to operate the proposed model. Some key specific features include an analog subsystem, with up to four analog-to-digital converters (ADCs); 16-bit mode for up to 12 external channels; eight windowed comparators with 12-bit digital-to-analog converter (DAC) references; three 12-bit buffered DAC outputs; 24 pulse width modulation (PWM) channels; and high resolution in both the A and B channels of 8 PWM modules. More details for the settings are described below.

(a) Settings of PWM module

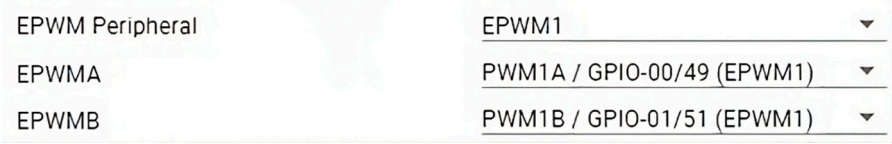
We select the ePWM1 module as the output of the PWM control signal in the system. The PWM time base period is set to 1000, producing a 50 kHz frequency PWM signal with a 50% duty ratio, as shown in Figure 3a. Counter Compare A (CMPA) and Counter Compare B (CMPB) are set to 500, generating positive and negative half-cycle PWM duty ratios, as shown in Figure 3b. PWM1A and PWM1B are set to output the PWM control signals, as shown in Figure 3c. Note that the PWM output frequency cannot be set beyond the system clock, i.e., 100 MHz. On the other hand, the duty cycle can be set from 0% to 100%.



(**a**) Setting of PWM period



(**b**) Setting of PWM counter compare register



(**c**) Setting of PWM1A and PWM1B

Figure 3. Setup of PWM.

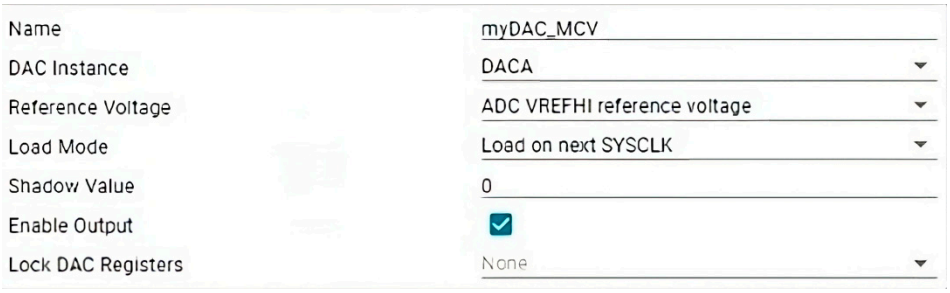
(b) ADC/DAC module and settings

DACA is selected as the system output channel, as shown in Figure 4a. The reference voltage (V_{ref}) of the DAC is set to 3.3 V. The analog output voltage (V_{out}) transformed from the DAC is calculated according to Equation (1), and it is connected to the scope for display. In the ADC module, the ADC-C2 and ADC-D0 channels are used to receive the control commands and current feedback signal, respectively, as shown in Figure 4b. The input analog voltage (V_{in}) of the ADC is normalized according to Equation (2). ePWM1 is set as the trigger source, and the ADC sampling time is calculated as shown in Equation (3).

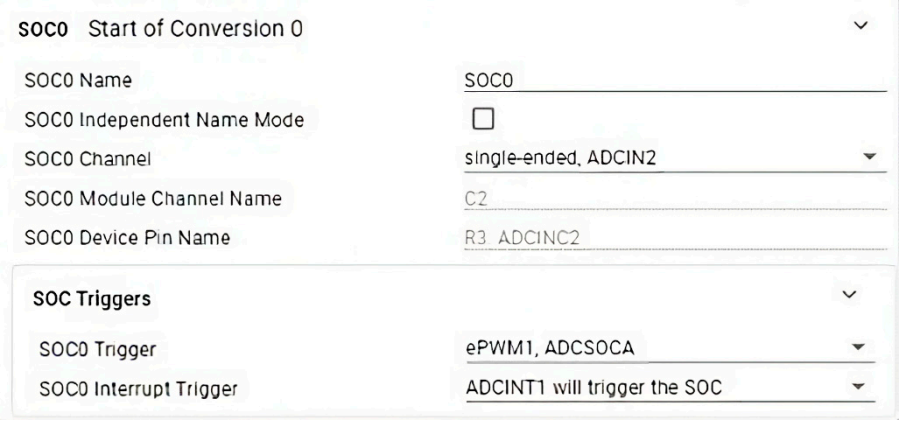
$$V_{out} = \frac{DAC \times 4095}{V_{ref}} + V_{offset} \quad (1)$$

$$V_{in} = \frac{(ADC - V_{offset})}{4095} \times V_{ref} \quad (2)$$

$$\text{Sampling time} = \frac{\text{PWM counter period}}{\text{systemclock}} \quad (3)$$



(**a**) Setting of DAC

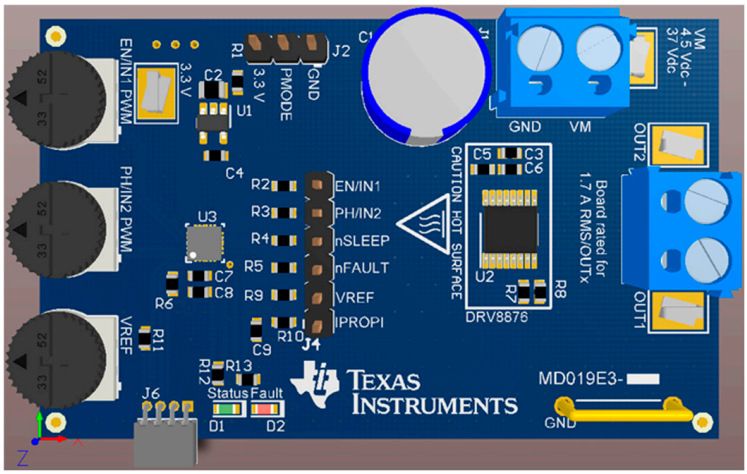


(**b**) Setting of ADC

Figure 4. ADC/DAC modules and settings.

(3) H-Bridge circuit

In this study, the DRV8874EVM Evaluation Module (Texas Instruments) is used to drive the H-Bridge circuit, as shown in Figure 5a. The control mode, PWM mode, and switch status of the H-Bridge circuit are shown in Figure 5b–d, respectively. PMODE is set to high voltage to operate the PWM control mode. The pins of EN and PH are used to receive the control signals from PWM1A and PWM1B, respectively. The switch, i.e., MOSFET, status is determined by the control signals.



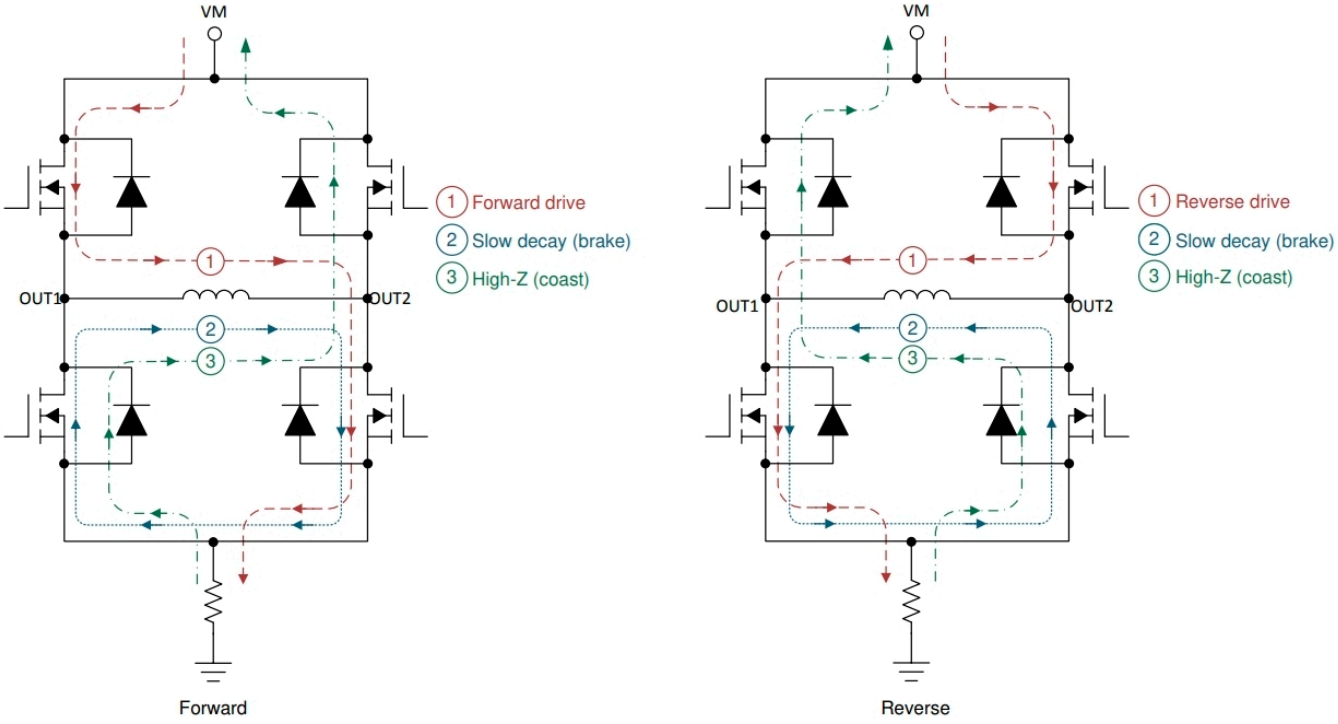
(**a**) DRV8874EVM module

PMODE STATE	CONTROL MODE
PMODE = Logic Low	PH/EN
PMODE = Logic High	PWM
PMODE = Hi-Z	Independent Half-Bridge

(**b**) Control mode

nSLEEP	IN1	IN2	OUT1	OUT2	DESCRIPTION
0	X	X	Hi-Z	Hi-Z	Sleep, (H-Bridge Hi-Z)
1	0	0	Hi-Z	Hi-Z	Coast, (H-Bridge Hi-Z)
1	0	1	L	H	Reverse (OUT2 → OUT1)
1	1	0	H	L	Forward (OUT1 → OUT2)
1	1	1	L	L	Brake, (Low-Side Slow Decay)

(**c**) PWM mode



(**d**) Switch status of H-Bridge circuit

Figure 5. H-Bridge module.

(4) EMI filter

The EMI filter is used to filter out high-frequency noise from the input signal, and its circuit is a standard circuit, as shown in Figure 6. OUT1 and OUT2 of the DRV8874EVM module are connected to the EMI filter.

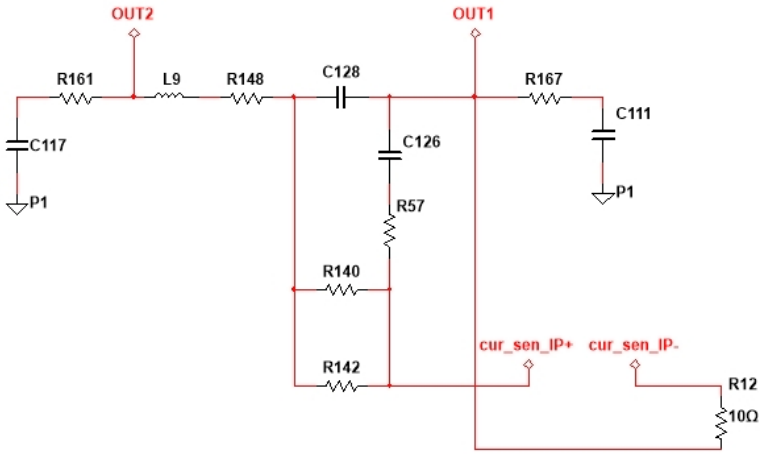
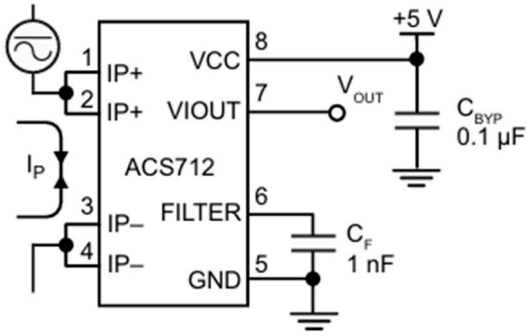


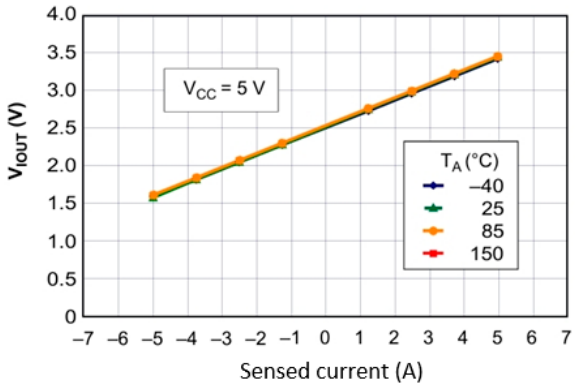
Figure 6. EMI filter circuit.

(5) Current sensor

The current sensor (ACS712-x05B) is used to detect the feedback load current. The entity diagram of ACS712-x05B is shown in Figure 7a, and its output voltage versus sensed current is shown in Figure 7b. IP+ is connected to the EMI filter, and IP- is connected to the load so that the load current can be detected. VOUT is the output of the voltage signal. When no current exists, the output voltage is set to 2.5 V.



(**a**) Diagram of ACS712-x05B



(**b**) Output voltage versus sensed current

Figure 7. Current sensor.

3. Digitization Process

The analog system digitization process is shown in Figure 8. Firstly, the analog controller is simulated using the Multisim 10 software package to obtain the transfer function of the analog controller. Secondly, the bilinear function transformation is implemented in MATLAB to extract the parameters of the digital controller. Thirdly, the parameters obtained are input to the Direct Form 2 compensator of the digital signal processor. The output of the digital signal processor is obtained and sent to the scope. Finally, the Bode response of the digital controller is displayed in the scope. Note that the digitization process directly uses a standard formula so that no additional decision loop is involved in the proposed model. More details are described in Section 3.2.

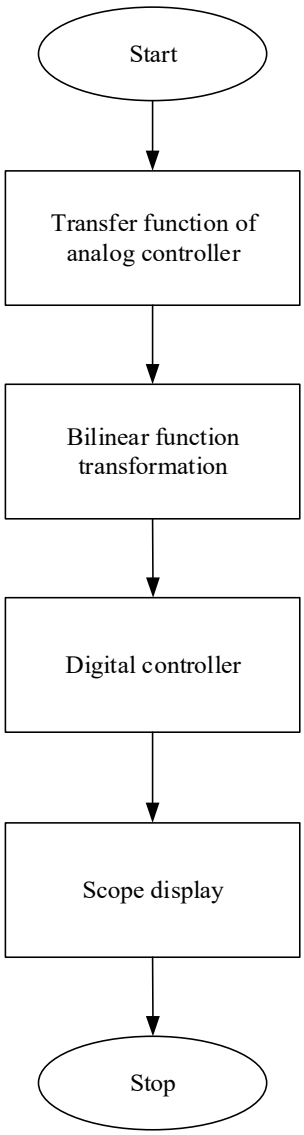
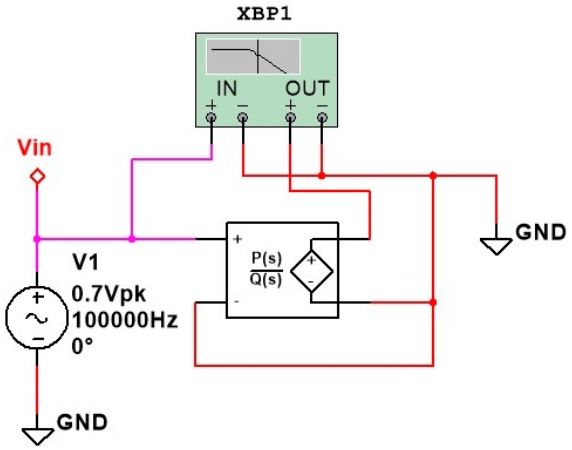


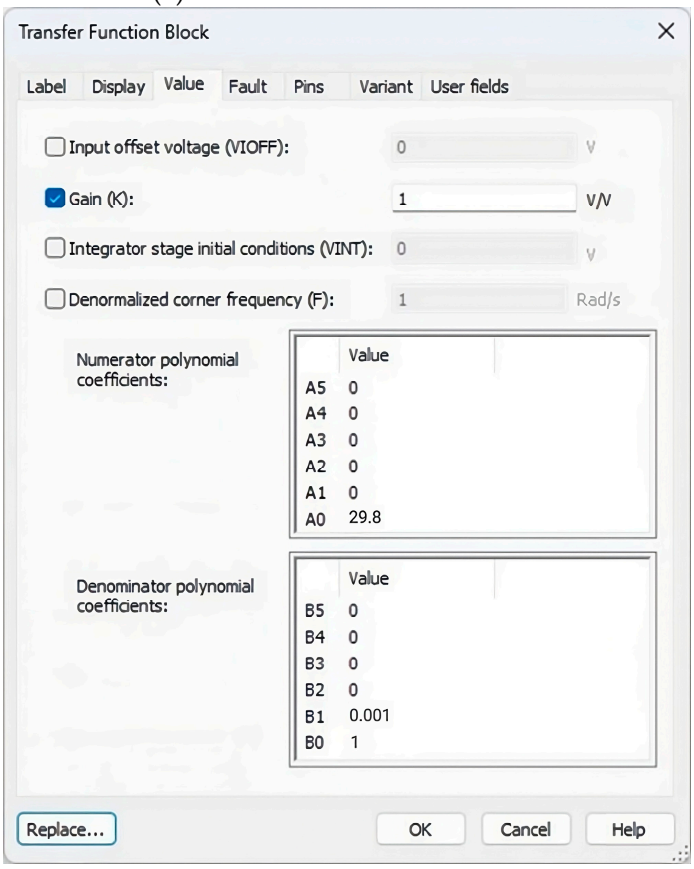
Figure 8. Flowchart of the digitization process.

3.1. Transfer Function of Analog Circuit Controller

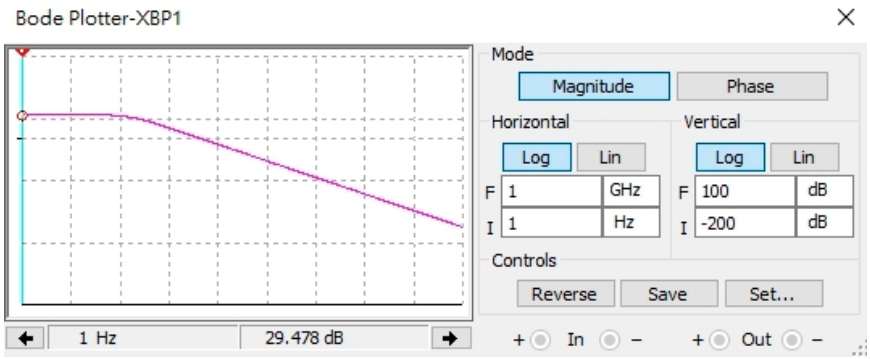
The transfer function of the analog circuit controller is input to the Multisim transfer function for simulation, as shown in Figure 9a. The simulation settings are shown in Figure 9, and the Bode response using the scope is shown in Figure 9c.



(**a**) Multisim controller simulation



(**b**) The simulation settings



(**c**) Bode response

Figure 9. Multisim simulation.

3.2. Bilinear Transformation

Bilinear transformation is used to transform an analog system to a digital system. Through $s = 2F_s\frac{(z-1)}{(z+1)}$, the z-domain function can be obtained by replacing the s variable in the analog type [15], expressed as

$$H(z) = H(s)|_{s=2F_s\frac{(z-1)}{(z+1)}} \quad (4)$$

where

$H(z)$: discrete-time transfer function;

$H(s)$: continuous-time transfer function;

F_s: sampling rate.

By introducing the transfer function $\left(H(s) = \frac{29.8}{0.001s+1}\right)$ of the analog controller into Equation (4), $H(z)$ can be expressed as shown in Equation (5). Equation (6) can thus be obtained using a simplification process.

$$H(z) = \frac{29.8}{0.001s+1}|_{s=2F_s\frac{(z-1)}{(z+1)}} \quad (5)$$

According to Equation (5), the program codes of the bilinear transformation are carried out using MATLAB R2022a software programming, as shown in Figure 10a. The outcome of all coefficients is obtained, as shown in Figure 10b, where $a_0 = -0.99005$, $b_0 = 0.14826$, and $b_1 = 0.14826$ in the digital controller.

```
Fs = 1 / 0.00001;


numerator = [0, 29.8];
denominator = [0.001, 1];


[num_discrete, den_discrete] = bilinear(numerator, denominator, Fs);
```

(**a**) Program codes of bilinear transformation

```
Command Window
>> untitled
b0: 0.14826
b1: 0.14826
a1: -0.99005
```

(**b**) Extracted coefficients of digital controller

Figure 10. Digital controller processing program codes.

As shown above, the bilinear transformation to transform an analog system to a digital system using Equation (4) is a standard formula. All coefficients demanded by the transformation process can be accurately achieved up to the fifth decimal place. Therefore, the computational error is sufficiently low to be ignored.

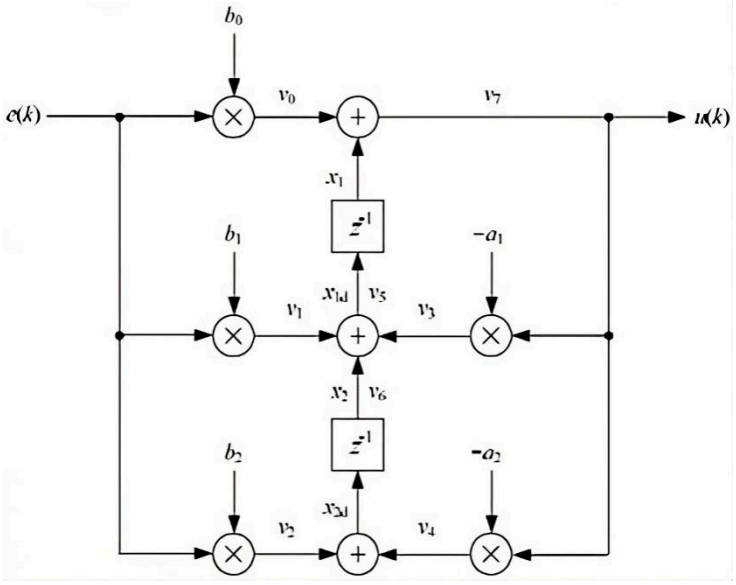
Based on the standard formula of the digital controller, as expressed in Equation (6), all coefficients, i.e., a_0, a_1, b_0, b_1, and b_2, demanded by the digital controller can be achieved, as shown in Equation (7).

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_0z^{-1} + a_1z^{-2}} \quad (6)$$

$$H(z) = \frac{0.14826 + 0.14826z^{-1}}{1 + 0.99005z^{-1}} \quad (7)$$

3.3. Digital Controller

The Direct Form 2 compensator, i.e., the digital filter structure shown in Figure 11a, is used to realize finite impulse response and infinite impulse response filters. It consists of two series of one-order filters, where the first filter receives the input signal, and its output is sent to the second filter. The coefficients, i.e., b0, b1, and b2, are the weight values of the feed-forward control path, and a1 and a2 are the weight values of the feedback control path. The settings of the Direct Form 2 compensator in the digital controller are shown in Figure 11b.



(**a**) Structure of Direct Form 2 compensator

Name	DCL_2P2Z_MCV_demodulator
Controller/Filter	Direct-Form 2, 2nd Order
Precision	Floating-Point (32-bit)
Controller Structure Chosen	DF22
Detailed Description	Click Question Mark for Details
a1	-0.99005f
a2	0.0f
b0	0.14826f
b1	0.14826f

(**b**) Settings of Direct Form 2 compensator in digital controller

Figure 11. Configuration of the digital controller.

3.4. Parameter Configuration and Realization of Digital System

(a) Configuration of system parameters

The configuration of the system parameters to realize the proposed system is shown in Table 1.

Table 1. Parameter configuration.

System Parameters	Value	Unit
Stimulus offset	1.65	V
Sampling time	10	μs
H-Bridge driver (VM)	28	V
Current sensor (VCC)	5	V
Current sensor without current (VOUT)	2.5	V

There are two ADC pins used for receiving the system control signal and current feedback signal in the TMS320F28379D, where their respective transfer codes are shown in Figure 12a and 12b, respectively.

```
adcResult_A0 = ADC_readResult(ADC0_PIN28_RESULT_BASE, ADC0_PIN28_SOC0);


rk = ((float32_t)adcResult_A0-2047.0f)/4095.0f*3.3f;



Interrupt_clearACKGroup(INT_ADC0_PIN28_1_INTERRUPT_ACK_GROUP);
ADC_clearInterruptStatus(ADC0_PIN28_BASE, ADC_INT_NUMBER1);
```

(**a**) System control signal

```
adcResult_cursen = ADC_readResult(ADC1_PIN12_RESULT_BASE, ADC1_PIN12_SOC0);

rk5 = ((float32_t)adcResult_cursen-3412.0f)/4095.0f*3.0f/0.185f;


Interrupt_clearACKGroup(INT_ADC1_PIN12_1_INTERRUPT_ACK_GROUP);
ADC_clearInterruptStatus(ADC1_PIN12_BASE, ADC_INT_NUMBER1);
```

(**b**) Current sensor signal

Figure 12. Transfer codes.

Major codes for processing ADCs are described as follows:

1. Read ADC:
   - ADC_readResult(base, socNumber): This is the function used to read the ADC transfer results from the assigned system on a chip (SOC).
   - ADCX_PINXX_RESULT_BASE: ADCX_PINXX is the base register address for storing the ADC results.
   - ADCX_PINXX_SOCX: This is the number of SOCs required to read the results via SOCX data transfer.
   - adcResult_A0: This is a variable set up by the user to be used to store the transfer results from the ADC reading.
2. Transfer ADC to real voltage values:
   - Transfer the digital data of adcResult_A0 to a real voltage value.
   - Real voltage transformation: In the normalization process, the real voltage is located between 0 and 3.3 V.
3. ADC interrupt:
   - Clear interrupt confirmation signal: Interrupt_clearACKGroup is used to clear the assigned interrupt confirmation signal, informing the interrupt controller for the interrupt function to be activated. This allows the next interrupt request to become available, such as INT_ADCX_PINXX_X_INTERRUPT_ACK_GROUP.
   - Clear ADC interrupt status: ADC_clearInterruptStatus is used to clear the assigned ADC interrupt sign. For example, ADC_INT_NUMBERX is the ADC interrupt index number, where X is the index to be cleared.

The transfer code for the DAC outputting the system signal to the scope is shown in Figure 13. The stages for processing the DAC codes are as follows:

1. System output results transferred to DAC output values.
   - Take the normalization process and locate the output range between 0 and 1.
   - The middle value of the DAC maximum voltage is adjusted to 1.65 V.

2. Output restriction: The output is restricted between 0 and 4095 to prevent it from exceeding the maximum range.
3. dacResult: This is used to store the DAC output variable, as shown in Figure 13.

```
dacResult=(((float)uk2*4095.0f)+2047.0f);
    if(dacResult>=4096.0f)
        dacResult=4095.0f;

          DAC_setShadowValue(myDAC_1_BASE, dacResult);
```

Figure 13. The transfer code for outputting the DAC signal.

(b) Realization of digital controller

The digital control codes for implementing the digital controller are shown in Figure 14 and are described as follows:

- DCL_runDF22_C1: This is the function for the digital control library to implement the digital compensator of second-order Direct Form 2.
- &DCL_2P2Z_MCV_demodulator: This is the name of the controller, including the allocation of the controller.
- Input and output: rk is used to store the voltage value obtained from the ADC, and uk2 is the result of the digital controller output.

```
uk2 = DCL_runDF22_C1(&DCL_2P2Z_MCV_demodulator,rk);
```

Figure 14. Digital control codes.

(c) PWM control signal

The major stages of processing the PWM codes are shown below, and the PWM control code is shown in Figure 15.

```
if(ek3>=1.6f)
    {
    ek3=1.6f;
    }
else if(ek3<=-1.6f)
{
    ek3=-1.6f;
}

if(ek3>=0.0f)
{
    duty=((float)ek3/1.6f)*500.0f;
    if(duty>500.0f)
    {
        duty = 500.0f;
    }

    EPWM_setCounterCompareValue(myEPWM1_BASE, EPWM_COUNTER_COMPARE_A, (uint16_t)duty);

}else{
    duty=((float)(-ek3)/1.6f)*500.0f;
    if(duty>500.0f)
    {
        duty = 500.0f;
    }

    EPWM_setCounterCompareValue(myEPWM1_BASE, EPWM_COUNTER_COMPARE_B, (uint16_t)duty);

}
```

Figure 15. PWM control code.

1. Limitation of output amplitude:

The system signal variable amount (ek3) is restricted between −1.6 and +1.6. This means that the output is set to 1.6 when ek3 is bigger than 1.6. On the other hand, the output is set to −1.6 if ek3 is smaller than −1.6.

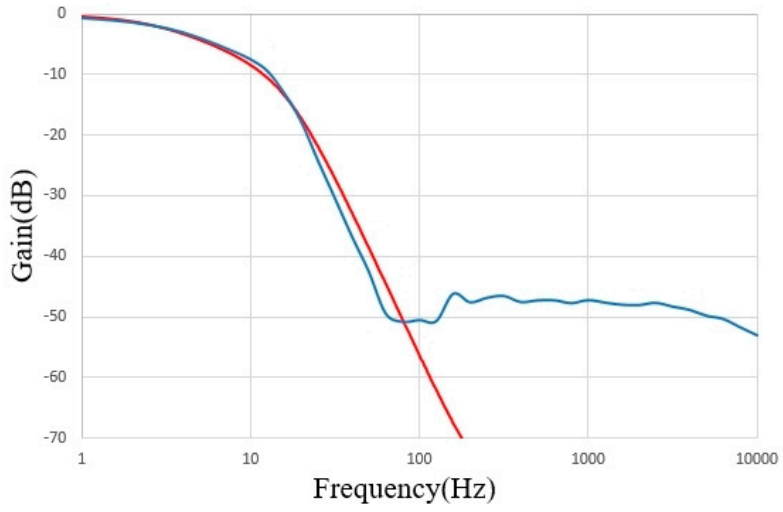
2. Duty ratio setting:
   - EPWM_setCounterCompareValue: This is a function used to adjust the PWM period and duty ratio.
   - myEPWMX_BASE: This is the base address of ePWM.
   - EPWM_COUNTER_COMPARE_X: This is the comparison register (X) of ePWM used to adjust the PWM period and duty ratio.
3. ek3 $\geq$ 0:
   - Normalization process: ek3 is located between 0 and 1.6 so that ek3 is divided by 1.6 for normalization between 0 and 1.
   - Range transformation: The comparison register (X) is set to max. 500 so that ek3 is thus multiplied by 500 to make its range between 0 and 500.
4. ek3 < 0:
   - Normalization process: ek3 is located between 0 and -1.6 so that ek3 is divided by -1.6 for normalization between 0 and 1.
   - Range transformation: The comparison register (X) is set to max. 500 so that ek3 is thus multiplied by 500 to make its range between 0 and 500.

4. Experimental Results and Analysis

The simulation results based on the Multisim 10 software package are compared with the experimental results for the Bode response and waveform response.

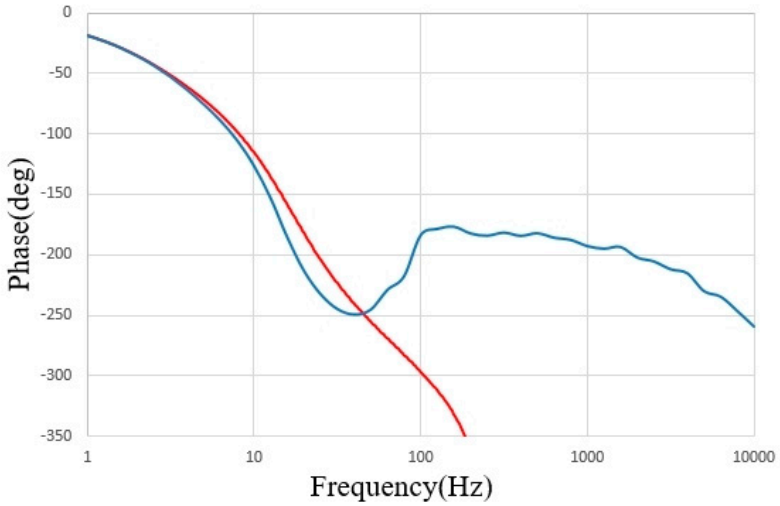
(a) Bode response

The comparison between the simulation and experimental results for the Bode response is shown in Figure 16. The comparisons of the Bode gain and phase are shown in Figure 16a and 16b, respectively, where the blue line represents the input signal, and the red line represents the output signal. It can be seen that both lines are almost identical when the frequency is located below 10 Hz. Beyond 10 Hz, the two lines are separated gradually since 100 Hz in gain and 60 Hz in phase as the frequency increases. For the Bode gain (dB), the breaking point appears at -50 dB, as shown in Figure 16a. For the Bode phase, the breaking phase is located at 250 deg, as shown in Figure 16b.



(**a**) Comparison of gain

Figure 16. *Cont.*

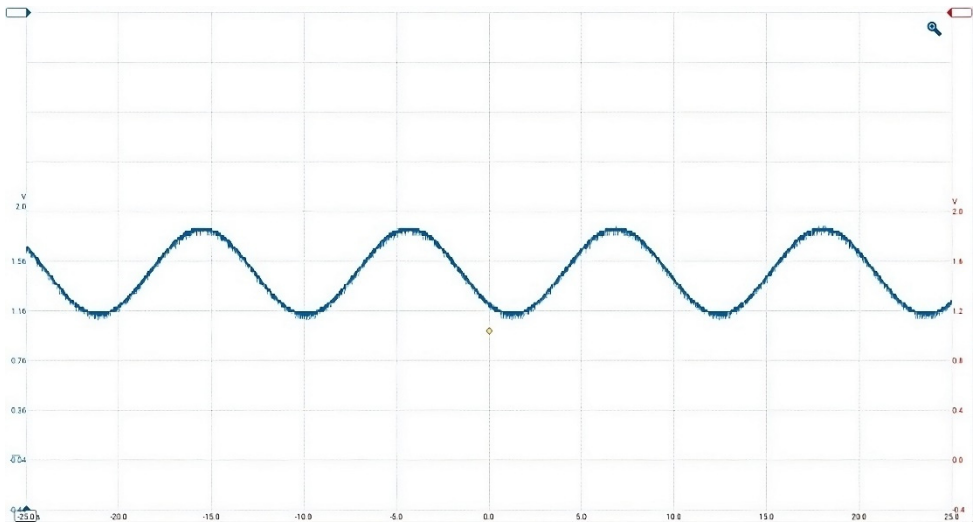


(**b**) Comparison of phase

Figure 16. Comparison of Bode responses.

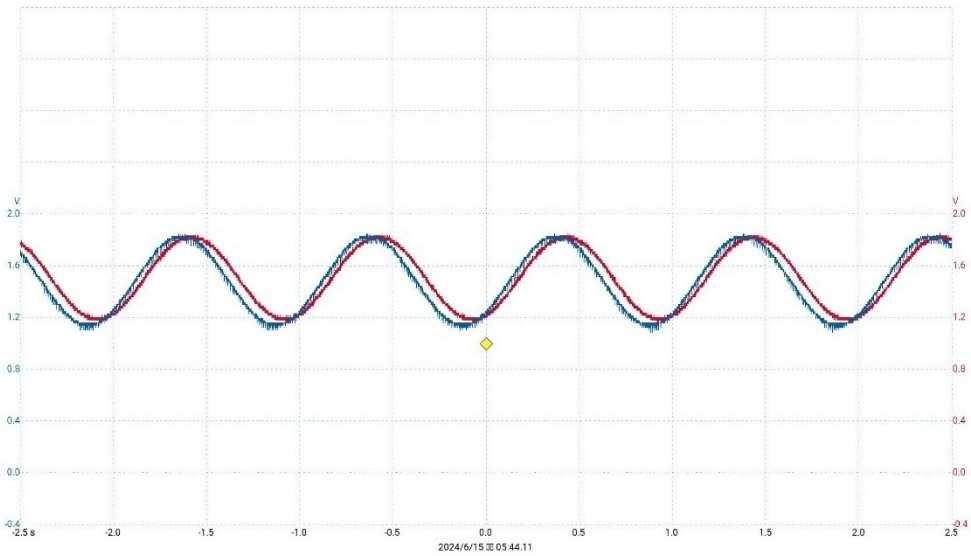
(b) Waveform tracking response

The tests for the waveform tracking response were carried out at 100 mHz, 1 Hz, and 1000 Hz, as shown in Figure 17, where the blue line represents the input signal, and the red line represents the output signal. In Figure 17a, the tracking waveform perfectly matches the input signal at 100 mHz. In Figure 17b, at 1 Hz, both the tracking waveform and the input signal are almost identical, with only a slight difference in the magnitude and phase of the output waveform. However, at 1000 Hz, the output signal cannot be identified due to signal attenuation toward almost zero, as shown in Figure 17c.

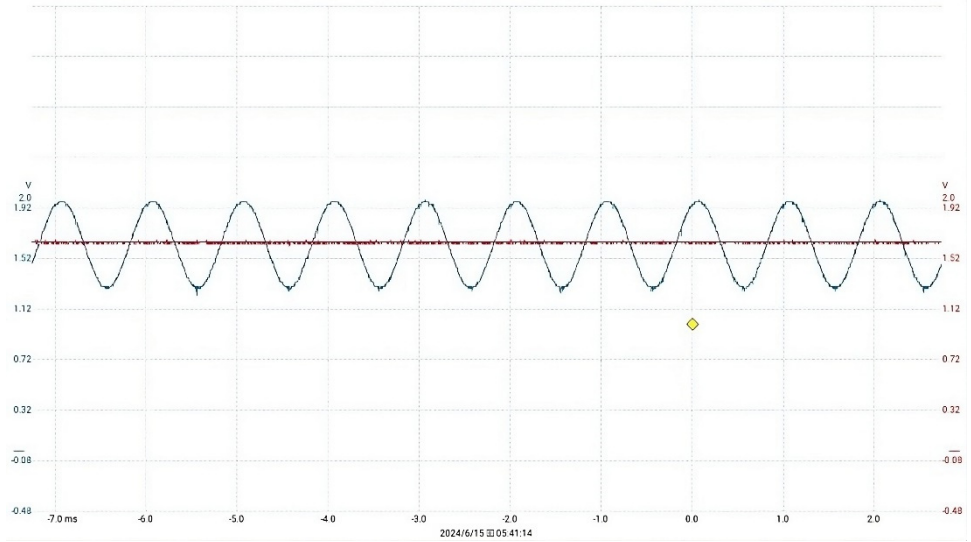


(**a**) Waveform response at 100 mHz

Figure 17. *Cont.*



(**b**) Waveform response at 1 Hz



(**c**) Waveform response at 1000 Hz

Figure 17. Waveform response under different frequency signals.

5. Discussion

An analog circuit was simulated using Multisim 10 simulation software, and the Bode plot was generated based on the input signals and actuator feedback signals in the simulated system. Both the simulation and experimental results, i.e., the Bode plots for gain and phase, are shown in Figure 16. The figure shows that the simulation results (red line) are very close to those from the experiments (blue line) at a low frequency below 10 Hz. This implies that the system is suitable for processing extremely low frequency signals, processed, e.g., in the wing control of a flight control system (FCS). However, from medium to high frequency, an error arises due to the lower gain. When the gain decreases to the −50 dB point, the distortion becomes obvious with sufficient accuracy of the DAC.

Additionally, Figure 17 shows that the responses in the actuator feedback sine wave signal (red) and input sine wave signal (blue) at 100 mHz and 1 Hz are consistent with each other, similar to the Bode plot response. When the signal frequency exceeds 1000 Hz, the output signal attenuates to almost 0 so that its gain and phase are unable to be identified.

Overall, this development in digitizing a low-frequency analog control circuit mainly uses Multisim and MATLAB R2022a software packages to provide an effective analog–digital control circuit transformation process. As can be seen, no additional requirements for

hardware circuit design and testing in advance are required for the proposed model. Also, all parameters required by the Direct Form 2 compensator of the digital signal processor can be directly extracted without an additional decision loop. Accordingly, this represents a direct and simple computational process with the economic benefit of being able to find the solution at a low cost.

6. Conclusions

In this study, analog control signals below 10 Hz were successfully digitized using a bilinear function algorithm. This verifies that digitized signals have a consistent response with analog signals. During the digitization process, in addition to the realization of the DSP, the conversion of the input and output signals, including the control signal modulation with the PWM mechanism, makes the process applicable to controlling the load in a control system. Based on the Bode response, it is obvious that the digitized signal is almost perfect in both gain and phase below 10 Hz, satisfying the actual control signal frequency range in practice. For example, at 100 mHz, the system output signal fully tracks the input signal in time. At 1 Hz, the output signal is almost identical to the input signal in both amplitude and phase. Similar results appear at 10 Hz. Due to the excess loop delay and the limitation of DAC precision, however, the proposed model is only suitable for processing lower signal frequencies. Higher signal frequencies beyond 10 Hz will cause increasing output signal attenuation and delay. Nevertheless, the experimental results verify that the proposed model works well in a low frequency-based control system. This means that it presents a good guideline for transforming a traditional analog control system into a digital control model in cases where a fast control switch is not demanded by the system. In future work considering higher-frequency signal processing, proportional integral (PI) control combined with the adaptive excess loop delay (ELD) compensation technique may be involved in extended and deeper research into this issue.

Author Contributions: Methodology, Z.-Y.C.; Software, Z.-Y.C.; Formal analysis, H.-C.L. and Z.-Y.C. Data curation, Z.-Y.C.; Writing—original draft, H.-C.L.; Writing—review & editing, H.-C.L.; Supervision, H.-C.L.; Funding acquisition, H.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
PWM	Pulse width modulation
FPGA	Field programmable gate array
D/A	Digital-to-analog converter
NMOS	n-Type complementary metal–oxide–semiconductor
DSP	Digital signal processor
EMI	Electromagnetic interference
FCS	Flight control system
PI	Proportional integral control
ELD	Excess loop delay

References

1. Suits, C.G. New Applications of Non-Linear Circuits To Relay and Control Problems. *Trans. Am. Inst. Electr. Eng.* **1932**, *51*, 914–921. [CrossRef]
2. Dastres, R.; Soori, M. A Review in Advanced Digital Signal Processing Systems. *Int. J. Electr. Comput. Eng.* **2021**.
3. Vanathy, B.; Dikshit, V.G.; Hotkar, V.G.; Reddy, S.K.; Jacob, A. Design and Development of Digital Controller for Airborne Electro Mechanical Actuators. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018.
4. Luo, Y.-L.; Chen, B.-J. The Design of UAV Control System Based on DSP. In Proceedings of the 2012 International Conference on Control Engineering and Communication Technology, Shenyang, China, 7–9 December 2012; pp. 527–530.
5. Jeong, S.; Kim, D.; Lee, I.-Y.; Im, D. An Analog Baseband Spectrum Sensing Circuit Employing Voltage Follower-Based Multiorder Channel Selection Filters. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 4714–4718. [CrossRef]
6. Chi, Y.; Zhang, H.; Zheng, Z.; Liu, R.; Qiao, L.; Cui, W. Analog Front-end Circuit Design for Wireless Sensor System-on-Chip. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 38–42. [CrossRef]
7. Crovetti, P.S.; Rubino, R.; Musolino, F. Relaxation Digital-to-Analog Converter with Foreground Digital Self-Calibration. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 10–21 October 2020; pp. 1–5. [CrossRef]
8. Bhandari, R.H.; Kotabagi, S.; Nayak, A. A 6-bit Low Power Digitally Controlled Oscillator. In Proceedings of the 2021 IEEE 18th India Council International for (INDICON), Guwahati, India, 19–21 December 2021; pp. 1–5. [CrossRef]
9. Pahl, D.; Pahl, L.; Mustafa, E.; Liu, Z.; Fabritius, P.; Mohan, J.; Clements, P.; Akin, A.; Esslinger, T. A low-noise and scalable FPGA-based analog signal generator for quantum gas experiments. In Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 18–22 October 2021; pp. 450–451. [CrossRef]
10. Selvaraj, S.; Bayram, E.; Negra, R. Low phase noise, high resolution digitally-controlled ring oscillator operating at 2.2 GHz. In Proceedings of the 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 7–9 September 2020; pp. 1–4. [CrossRef]
11. Jiang, J.; Li, J.; Zhang, D.; Hu, C.; Peng, X.; Sun, N.; Wang, S.; Zhang, Z.; Cui, W. Research on Analog-to-Digital Converter (ADC) Dynamic Parameter Method Based on the Sinusoidal Test Signal. *Electronics* **2022**, *11*, 2125. [CrossRef]
12. Wu, W.; Ji, H. Research on Conversion between Analog Signal and Digital Signal of Single Chip Microcomputer Based on Improved Genetic Algorithm. In Proceedings of the 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), Bristol, UK, 29–31 July 2023; pp. 545–549. [CrossRef]
13. Wang, L.; Chen, W.; Chen, K.; He, R.; Zhou, W. The Research on the Signal Generation Method and Digital Pre-Processing Based on Time-Interleaved Digital-to-Analog Converter for Analog-to-Digital Converter Testing. *Appl. Sci.* **2022**, *12*, 1704. [CrossRef]
14. Kumar, V.; Arya, M.; Kumar, A.; Jhariya, D.K. Design and Comparison Between IIR Butterworth and Chebyshev Digital Filters using MATLAB. In Proceedings of the 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 11–12 January 2024; pp. 1–7.
15. Getu, B.N. Digital IIR Filter Design using Bilinear Transformation in MATLAB. In Proceedings of the 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Sharjah, United Arab Emirates, 3–5 November 2020; pp. 1–6.
16. Samsudin, N.N. Implementation of Optimized Low Pass Filter for ECG filtering using Verilog. *J. Phys. Conf. Ser.* **2022**, *2312*, 012049. [CrossRef]
17. Fan, Y.; Zhang, M.; Li, J. Improvement of Power Factor and its Multisim Simulation. In Proceedings of the 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 28–30 July 2020; pp. 766–769.
18. Mohammed, Y.R.; Basil, N.; Bayat, O.; Mohammed, A.H. A New Novel Optimization Techniques Implemented on the AVR Control System using MATLAB-SIMULINK. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 4515–4521.
19. Buccella, C.; Cecati, C.; Latafat, H. Digital Control of Power Converters—A Survey. *IEEE Trans. Ind. Inform.* **2012**, *8*, 437–447. [CrossRef]
20. Dyer, S.A.; Dyer, J.S. The bilinear transformation. *IEEE Instrum. Meas. Mag.* **2000**, *3*, 30–34. [CrossRef]
21. Goswami, O.P.; Upadhyay, D.K.; Rawat, T.K. Extended bilinear transform and multirate technique based approach for analog-to-digital transform. *Int. J. Electron.* **2021**, 1–15. [CrossRef]
22. Xu, Z.; Bai, Y. A transfer function based on bilinear transformation for narrow-band seismic record. *PLoS ONE* **2024**, *19*, e0297999. [CrossRef]
23. Zhang, G.; Chen, X.; Chen, T. Digital redesign via the generalised bilinear transformation. *Int. J. Control* **2009**, *82*, 741–754. [CrossRef]

24. Kwaha, B.J.; Kolawole, E.A.; Batu, A.M. The Design and Implementation of a Digital Infinite Impulse Response (IIR) Lowpass Butterworth Filter-a Comparison of Matlab and Bilinear Transformation Methods. *Indian J. Sci. Technol.* **2011**, *4*, 451–455. [CrossRef]
25. Patanavijit, V. Computer Modelling and Simulation of Bilinear Transformation on the Digital Infinite Impulse Response Butterworth Filter. *Transformation* **2019**, *1*, 1–2. [CrossRef]
26. Kim, S.; Na, S.-I.; Yang, Y.; Kim, S. A 2-MHz BW 82-dB DR Continuous-Time Delta–Sigma Modulator With a Capacitor-Based Voltage DAC for ELD Compensation. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 1999–2006. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.