

# Frequency response analysis of microcontroller-based discrete-time lead-lag compensators

GUNAWAN DEWANTORO<sup>ORCID</sup>, IRWIN SHAUMA RIZKY, BUDIHardJA MURTIANTA

*Department of Electronic and Computer Engineering*

*Satya Wacana Christian University*

*Jl. Diponegoro 52-60 Salatiga, Central Java, Indonesia 50711*

*e-mail: gunawan.dewantoro@staff.uksw.edu, irwine2424@gmail.com, budihardja.murtianta@uksw.edu*

(Received: 15.02.2020, revised: 20.07.2020)

**Abstract:** With the rapid advancement of digital processors, filters have been commonly implemented using microcomputers. In this study, a low-cost and compact Arduino Uno development board was used to realize digital lead and lag compensators. Arduino boards are very affordable. Consequently, they were investigated to see if they were capable of preserving the frequency response of continuous-time compensators. The experiments required a set of equipment including a function generator, an Arduino Uno development board, a PC-based oscilloscope, and a laptop. The signal frequency was varied from 0 to 500 Hz. Two discretization methods were employed, namely bilinear transformation and matched pole-zero mapping. The results showed that an Arduino Uno board can be utilized to implement lead and lag compensators to some extent. The discrete-time compensator preserved the capability of filtering out certain frequencies. The change in DC gain was negligible, however, there was a significant difference in the cut-off frequency and transient slope. For both discretization methods, the frequency responses at high frequency experienced a rippling profile.

**Key words:** Arduino Uno, discretization, frequency response, lag compensator, lead compensator

## 1. Introduction

Lead and lag compensators, among the many kinds of compensators, are the most employed control architectures in various process engineering. Therefore they have always attracted a lot of attention [1]. As is known, the lead/lag compensators are characterized by a first-order system as defined by [2]:

$$C(s) = k \frac{1 + \alpha \tau s}{1 + \tau s} . \quad (1)$$



© 2020. The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits use, distribution, and reproduction in any medium, provided that the Article is properly cited, the use is non-commercial, and no modifications or adaptations are made.

As shown in (1), the DC gain,  $k$ , is typically determined to meet the steady-state specifications. Meanwhile, the other two accompanying parameters, i.e.  $\alpha\tau$  and  $\tau$ , are the time constants of the finite zero and pole, respectively [3]. Recent literature shows that there have been numerous attempts to directly design lead and lag compensators [4–9] and fractional lead and lag compensators [10–14].

The employment of lead and lag compensators has been presented using various techniques through simulations [15–17]. Nowadays, these kinds of compensators have been embedded into many kinds of digital computers for various applications, such as: grid inverter control [18–19], single-phase power corrector [20], axes control of CNC machines [21], permanent magnet synchronous motor control [22], buck converter designing [23], power systems [24], etc. The realization of digital compensators can be done by means of programming. The Arduino is a development board that consists of several parts including a microcontroller, communication peripherals, input/output pins, crystal oscillator, and power pins. The Arduino Integrated Development Environment (IDE) is an open-source software that provides a compact way to write codes and also compiles these codes to become a binary .hex file [25]. Reference [26] demonstrates that PI controllers can be discretized and realized using an Arduino Uno board. The experiment results exhibit an adequate agreement regarding the transient response, as well as the stability, which validates the feasibility of employing Arduino in industries or factories. Bauer in [27] showed that the employment of a non-integer order filter using an Arduino Uno development board was possible. Biomedical data, particularly electroencephalography signals, were applied and then compared to the simulated performance using Matlab/Simulink. Likewise, Arduino Uno was applied to generate signals, in particular heart sounds, and murmurs, for research purposes [28]. Different types of heart sounds, as well as murmurs, were identified by an Arduino-based signal generator. For face tracking applications, separate lead/lag compensators, as well as a PID control, were applied using an Arduino Uno board [29]. The face images were captured optically by a camera and tracked accordingly. The position of the camera was organized using an Arduino-based controller and two servo motors to adjust the orientation pan and tilt angles.

The research mentioned above successfully realized digital controllers using the Arduino Uno. However, the assessments of performance were conducted in terms of the time response. This means that all parametric keys were in the realm of time domains, i.e. disturbance rejection, closeness to set point, rise time, etc. In control fields, a frequency response analysis provides useful insights into stability, suppression to disturbance, and other dynamic performances of the control system [30]. Several exhaustive studies of comparisons between different discretization strategies were investigated. It has been demonstrated that the selection of the discretization technique is the key factor for constructing resonant controllers [31]. In many simulations, the discretized controllers exhibited nearly identical frequency responses to those of the continuous-time controller [32–34].

In this paper, lead and lag compensators were implemented using the Arduino Uno board to see the practicality with regards to the frequency response. The discretization methods included a numerical method, namely the bilinear transformation or Tustin approximation, and a mapping method, i.e. the matched pole-zero mapping method. The outline of this article is according to the following order: section 1 describes the introductory background and gaps. The methods used are explained in section 2. The experiment results are shown in section 3 and discussed in section 4. Finally, section V concludes the completed study and sets up future works.

## 2. Proposed methods

### 2.1. Arduino-based filter discretization

In a general sense, lag and lead compensators are simply classes of filters. With the massive developments in digital computers, including microcontrollers, it is a simple matter to realize a comparable digital controller for a given analog controller. In this study, a low-cost Arduino Uno board was programmed in such a way that the digital controllers might behave the same as the analog controllers. In most practical situations, when discretizing a continuous-time filter, one might desire to maintain one or more of the following properties: impulse response characteristics, step response characteristics, dc gain, phase and gain margins, and pole or zero locations. Nevertheless, not all of these properties will be preserved once an analog filter is discretized. In particular, it is always difficult to resemble the frequency response characteristics of the original continuous-time filter to those of the discretized filter.

Regarding the characteristics of frequency response, it is important to pay attention as to whether or not the analog filter is discretized. If it is, the resulting filter may exhibit an undesirable phenomenon, e.g. frequency folding that is not present in the original continuous-time filter. This might occur if the sampling frequency is too low to fulfill the sampling theorem. To reconstruct the original signal from a previously sampled signal, the lowest frequency during the sampling process must be satisfied [35]. Such a minimum frequency is specified by the following Nyquist rate:

$$\omega_s > 2\omega_1, \quad (2)$$

where  $\omega_1$  is the maximum frequency component that composes the continuous-time signal and  $\omega_s$  is the sampling frequency. Although the requirement of the minimum sampling frequency is shown in (2), practical considerations may make it necessary to sample at a frequency much higher than this theoretical minimum.

An Arduino Uno board, as shown in Fig. 1, is a development board which uses ATmega328P as the controller. This board is equipped with 14 digital input and output pins, various interface protocols, and 6 analog pins. The board is clocked by a 16 MHz crystal.



Fig. 1. Arduino Uno board

Since there is no single approximation method of general use, the expected performance from the digital algorithms must be known in advance. In this study, an Arduino Uno board was programmed to be a digital lag and lead compensator using the bilinear transformation

and matched the pole-zero mapping method to preserve the frequency response. The direct  $z$  transformation approaches were not of interest because the resulting frequency spectrum was likely to be distorted [36].

## 2.2. Bilinear transformation

The bilinear transformation is a numerical method based on the trapezoidal integration rule, by approximating the area

$$\int_{(k-1)T}^{kT} y(t) dt \quad \text{and} \quad \int_{(k-1)T}^{kT} x(t) dt,$$

by

$$\frac{1}{2} [y(kT) + y((k-1)T)] \quad \text{and} \quad \frac{1}{2} [x(kT) + x((k-1)T)],$$

respectively [37]. The variable  $y(t)$  denotes the output of the continuous-time filter,  $x(t)$  is the input of the continuous-time filter,  $k$  is the sampling instant, and  $T$  is the sampling period. The discrete-time filters can be obtained by mapping the  $s$ -plane to the  $z$ -plane by using the following equation:

$$G_D(z) = G(s) \Big|_{s=\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}}, \quad (3)$$

where:  $G_D(z)$  is the discretized filter,  $G(s)$  is the continuous-time filter, and  $T$  is the sampling period. The left half of the  $s$ -plane is mapped into the region  $\text{Re}((z-1)/(z+1)) < 0$ , which is equivalent to  $\sigma^2 + \omega^2 < 1$ . Hence the bilinear transformation maps every part of the left half-plane in the  $s$  domain into the unit circle of the  $z$  plane and always guarantees the stability of the discrete-time filter as long as the continuous-time filter is also stable.

## 2.3. Matched pole-zero mapping

In this method, the numerator and denominator of the continuous-time filter  $H(s)$  are considered separately. The poles of  $H(s)$  are mapped to the poles of the discrete-time filter, and the zeros of  $H(s)$  are mapped to the zeros of the discrete-time filter. The matched pole-zero mapping method follows the following procedure:

1. The finite poles and finite zeros of  $H(s)$  are mapped to the  $z$  plane poles and zeros according to the relation of  $z = e^{sT}$ .
2. The infinite poles and infinite zeros of  $H(s)$  are mapped to the point  $z = -1$ . Thus, for each infinite pole and zero, there is a factor of  $z+1$  in the denominator and numerator of  $H_D(z)$ , respectively.
3. The gain of the discrete-time filter is modified to match the gain of the continuous-time filter. For the case of low-pass filters, the gain of the continuous-time filter at  $s = 0$  should be identical to the gain of the discrete-time filter at  $z = 1$ . Similarly, for high-pass filters, the gains should be matched at  $s = \infty$  and  $z = -1$ .

### 3. Results

#### 3.1. Experimental setup

The experimental setup is shown in Fig. 2. The signal processing was carried out by an Arduino Uno board. The function generator used was GWInstek GFG-8216A, by which sine waves with varied frequencies were produced. A computer-based oscilloscope Hantek 6022BL was utilized to record the sine waves processed by the Arduino board, and measure the amplitude of each sine wave. It also provided information as to whether or not the signal exceeded the permitted voltage range (0–5 V). The experiment was conducted according to the procedure as shown in Fig. 3. The discretized lead and lag compensator were first coded in the Arduino Integrated Development Environment (IDE). The code itself contained some lines describing the difference equations of the filter. In this study, the analog lead and lag compensators are defined by (4) and (5), respectively.

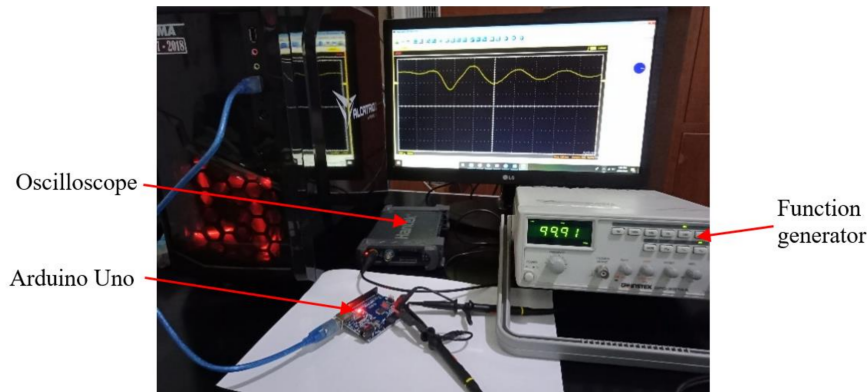


Fig. 2. The experiment setup

Based on Eq. (1), the value of  $\alpha$  is less than 1 for the lead compensator, meaning that the finite pole is located to the left of the finite zero. While for the lag compensator, the value of  $\alpha$  is greater than 1 which means that the finite zero is located to the left of the finite pole.

$$H_{\text{lead}}(s) = \frac{s + 10}{s + 100}, \quad (4)$$

$$H_{\text{lag}}(s) = \frac{s + 1000}{s + 100}. \quad (5)$$

Using the bilinear transformation, the equivalent digital filters in the  $z$ -domain with  $T = 1$  ms are shown by (6) and (7):

$$H_{\text{lead}_1}(z) = \frac{0.9571 - 0.9476z^{-1}}{1 - 0.9048z^{-1}}, \quad (6)$$

$$H_{\text{lag}_1}(z) = \frac{1.429 - 0.4762z^{-1}}{1 - 0.9048z^{-1}}. \quad (7)$$

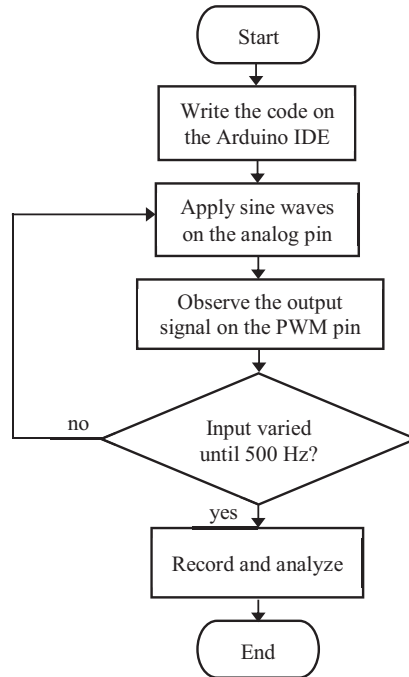


Fig. 3. Flowchart of the experimentation

Meanwhile, the matched pole-zero mapping with  $T = 1$  ms yielded digital filters as formulated by (8) and (9):

$$H_{\text{lead}_2}(z) = \frac{0.9546 - 0.9469z^{-1}}{1 - 0.9048z^{-1}}, \quad (8)$$

$$H_{\text{lag}_2}(z) = \frac{1.505 - 0.5538z^{-1}}{1 - 0.9048z^{-1}}. \quad (9)$$

All the resulting calculations in Eqs. (6)–(9) were rounded to four decimal places. The input signal that originated from the function generator was sampled with a sampling frequency of 125 kHz. This was done by setting the ADC prescaler to 128 as the Arduino Uno board employs a 16 MHz internal oscillator. The sine wave that was put into the Arduino Uno board started from 0 Hz to 500 Hz, meaning that the sampling frequency fulfilled the Nyquist rate requirement. In the range of 0–9 Hz, the input signal was sampled with an increment of 1 Hz, whereas in the range of 10 Hz–500 Hz, the input signal was sampled with an increment of 10 Hz, resulting in a total of 60 different frequencies of the input signal. Fig. 4 shows an example of a 100 Hz sine wave processed by the Arduino board. For all the experiments that employed a sinusoid wave input, the signal peak-to-peak amplitude was kept at 2 V and then a +1 V offset was applied such that the wave trough was equal to 0 V. It was performed this way because the Arduino ADC works within the 0–5 V range.

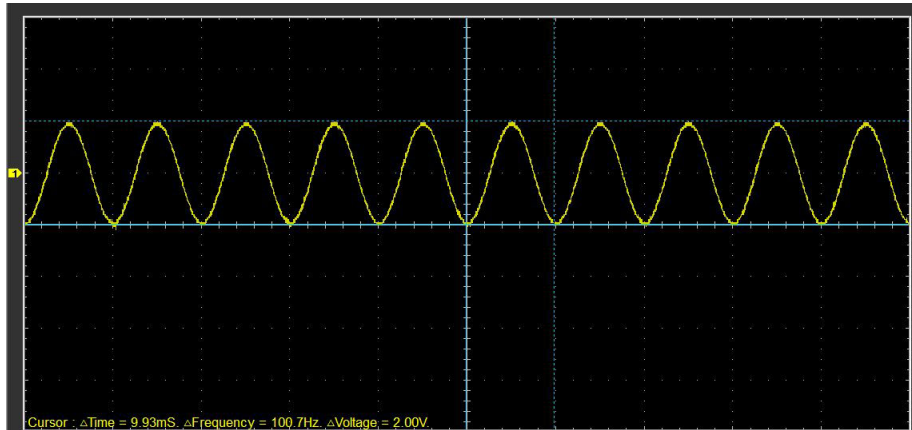


Fig. 4. A 2-Vpp sine wave as the compensator input

### 3.2. The spectrum of a lead compensator

By means of inverse  $z$  transformations [38], the difference equations of the digital lead compensators using a bilinear transformation and matched pole-zero mapping can be found, as formulated by (10) and (11), respectively.

$$y_1(k) = 0.9571x(k) - 0.9476x(k - 1) + 0.9048y(k - 1), \quad (10)$$

$$y_2(k) = 0.9546x(k) - 0.9469x(k - 1) + 0.9048y(k - 1). \quad (11)$$

The plots of the frequency response of the lead compensator are shown in Fig. 5 and Fig. 6.

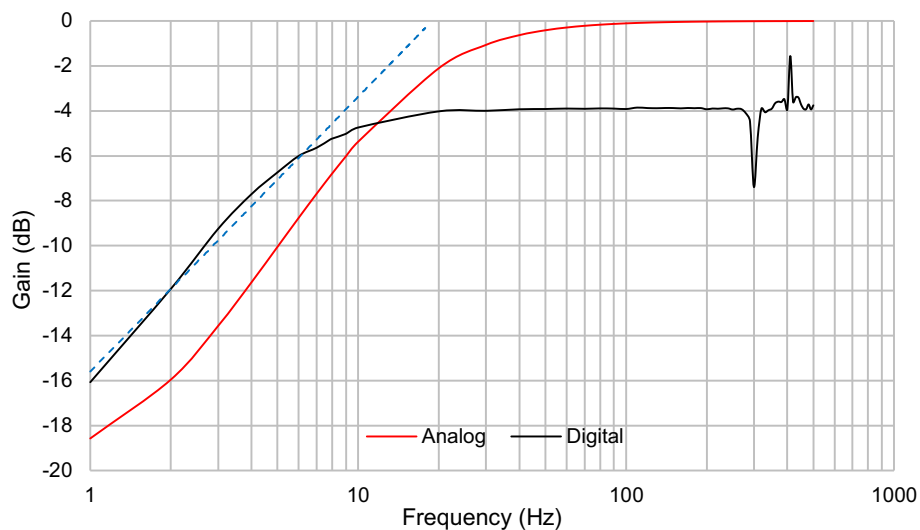


Fig. 5. The frequency response of the digital lead compensator using a bilinear transformation

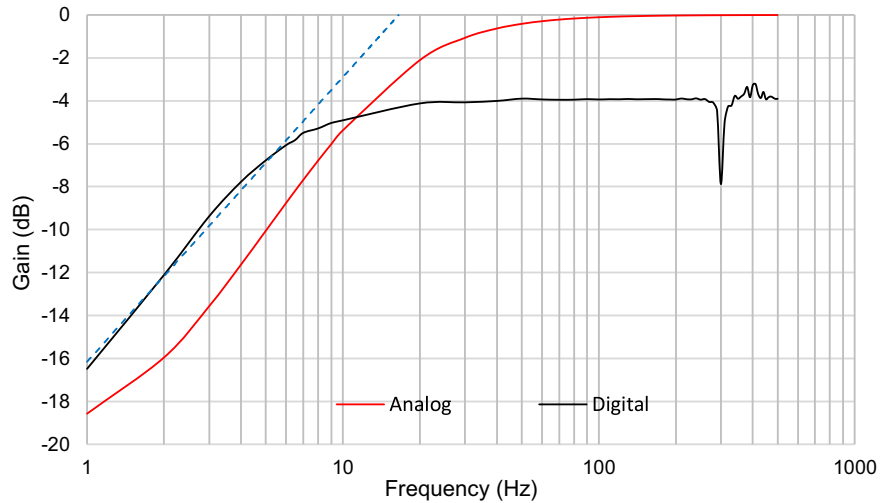


Fig. 6. The frequency response of the digital lead compensator using the matched pole-zero technique

### 3.3. The spectrum of a lag compensator

Similarly, the difference equations of digital lag compensators were derived by means of inverse z transformations. Eqs. (12) and (13) show the corresponding difference equations using bilinear transformation and matched pole-zero mapping, respectively.

$$y_3(k) = 1.429x(k) - 0.4762x(k - 1) + 0.9048y(k - 1), \quad (12)$$

$$y_4(k) = 1.505x(k) - 0.5538x(k - 1) + 0.9048y(k - 1). \quad (13)$$

The plots of the frequency response for the lag compensator are shown in Fig. 7 and Fig. 8.

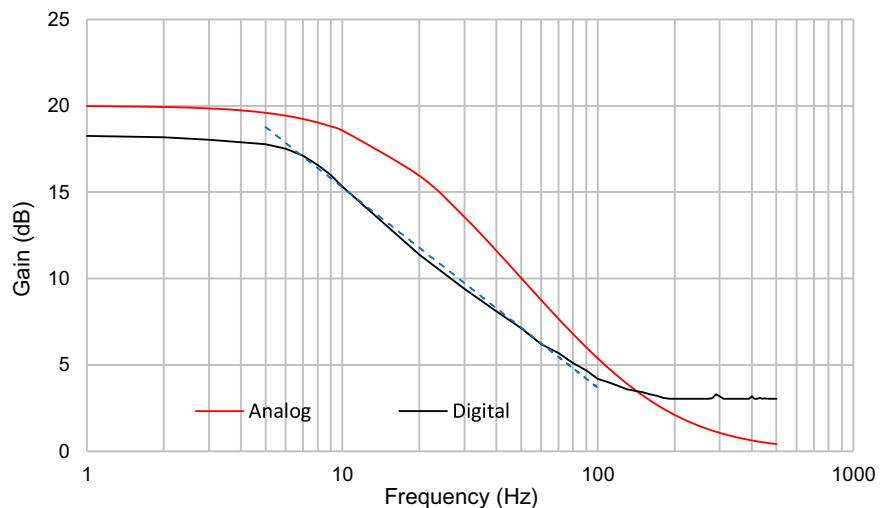


Fig. 7. The frequency response of the digital lag compensator using a bilinear transformation



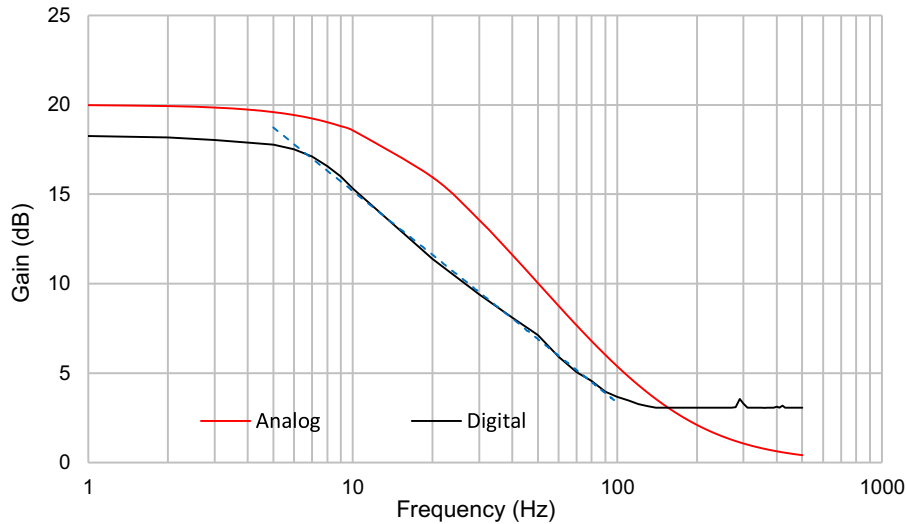


Fig. 8. The frequency response of the digital lag compensator using the matched pole-zero technique

## 4. Discussions

### 4.1. Analysis of a lead compensator

In general, the analog and digital lead compensators exhibited the profile of a high-pass filter which filters out low-frequency signals. This spectrum profile is in agreement with the findings in [39]. A comparison of the frequency response characteristics is shown in Table 1.

Table 1. Comparisons of lead compensators

	DC gain	Cut-off freq. (Hz)	Slope (dB/dec)
Analog	0.1	15	14.34
Bilinear transformation	0.1044	4.8	12.23
Matched pole-zero	0.1028	4.7	13.26

From Table 1, it can be inferred that all the parametric keys of the frequency response for each lead compensator are not the same. For the lead compensator discretized using a bilinear transformation, the DC gain was measured at a frequency of 0 Hz, with  $V_o = 0.1608$  V and  $V_{in} = 1.5396$  V, resulting in a DC gain of 0.1044 V/V. The largest gain was approximated by averaging the gains from  $f = 200$  Hz to  $f = 500$  Hz, which gives  $-3.9175$  dB. This value was used to find out the gain at the cut-off frequency by subtracting 3 dB, yielding the cut-off frequency of 4.8 Hz. To measure the steepness of the slope, a regression equation was constructed to approximate the gradient from 1 Hz to 8 Hz, as follows:

$$y = 5.3077 \ln(x) - 15.597, \quad (14)$$

which gives a slope of 12.23 dB/decade. Compared to the original analog lead compensator, this discretized lead compensator produced a slightly higher DC gain. The cut-off frequency shifted to the left by as much as 11.2 Hz, which resulted in a lower gradient slope.

For the lead compensator discretized using matched pole-zero mapping, the DC gain was obtained with  $V_o = 0.158$  V and  $V_{in} = 1.5371$  V, resulting in a DC gain of 0.1028 V/V. The largest gain was approximated by averaging gains from  $f = 120$  Hz to  $f = 500$  Hz, which gives  $-3.9506$  dB. By subtracting 3 dB from  $-4.0019$  dB, the gain at the cut-off frequency was obtained with a cut-off frequency of 4.7 Hz. To measure the steepness of the slope, a regression equation was constructed to approximate the gradient from 1 Hz to 8 Hz, as follows:

$$y = 5.7602 \ln(x) - 16.155, \quad (15)$$

which gives a slope of 13.26 dB/decade. Compared to the theoretical analog lead compensator, this discretized lead compensator possessed a closer DC gain than that of the filter discretized using a bilinear transformation. However, the cut-off frequency shifted to the left by as much as 11.3 Hz, making the transient slope steeper than that of the filter discretized using a bilinear transformation.

For both discretization methods, the frequency responses at a high frequency were deteriorated by the presence of a notch at about  $f = 300$  Hz followed by a rippling profile. The gain at a low frequency was higher than that of the analog lead compensator, while the gain at a high frequency was lower. These discrepancies are due to the limitations of Arduino Uno data types. The floating-point numbers are represented in 32 bits which allow for up to 7 decimal places for the mantissa part. Filters with fractional coefficients are prone to rounding errors when implemented using the Arduino Uno [27, 40].

#### 4.2. Analysis of a lag compensator

In general, the analog and digital lag compensators exhibited the profile of a low-pass filter which filters out high-frequency signals. A comparison of the frequency response characteristics is shown in Table 2.

Table 2. Comparisons of lag compensators

	DC gain	Cut-off freq. (Hz)	Slope (dB/dec)
Analog	10	10	-13.98
Bilinear	9.9548	7.4	-11.6
Matched	9.9462	7.5	-11.9

From Table 2, it can be concluded that all the parametric keys of the frequency response for each lag compensator changed. For the lag compensator discretized using a bilinear transformation, the DC gain was measured at a frequency of 0 Hz, with  $V_o = 15.288$  V and  $V_{in} = 1.5371$  V, resulting in a DC gain of 9.9548 V/V. This value was used to find out the gain at the cut-off frequency by multiplying it by 0.707. The corresponding cut-off frequency was found at  $f = 7.4$  Hz.

To measure the steepness of the slope, a regression equation was constructed to approximate the gradient from 7 Hz to 80 Hz, as follows:

$$y = -5.04 \ln(x) + 26.882, \quad (16)$$

which gives a slope of  $-11.6$  dB/decade. Compared to the original analog lag compensator, this discretized lead compensator produced a slightly lower DC gain. The cut-off frequency shifted to the left by as much as 2.6 Hz which resulted in a lower gradient slope.

For the lag compensator discretized using matched pole-zero mapping, the DC gain was obtained with  $V_o = 11.8372$  V and  $V_{in} = 1.1901$  V, resulting in a DC gain of 9.9462 V/V or 19.953 dB. This value was used to find out the gain at the cut-off frequency by multiplying it by 0.707. The corresponding cut-off frequency was found at  $f = 7.4$  Hz. To measure the steepness of the slope, a regression equation was constructed to approximate the gradient from 7 Hz to 80 Hz, as follows.

$$y = -5.204 \ln(x) + 27.211, \quad (17)$$

which gives a slope of  $-11.9$  dB/decade. Compared to the bilinear transformation, this method resulted in a lower DC gain and steeper roll-off slope. Meanwhile, the cut-off frequency shifted to the left by as much as 2.5 Hz.

For both discretization methods, the frequency responses at high frequencies experienced a rippling profile after a sudden increase at  $f = 290$  Hz. The gain at a low frequency was lower than that of the analog lead compensator, while the gain at a high frequency was higher.

#### 4.3. Effects of Arduino-based lead/lag compensators

In control systems, the lead and lag compensators are utilized to meet some specific requirements, including stability, transient response, and steady-state response. In the experiment, the magnitude response was distorted and thus led to some consequences. Compared to the analog lead compensator, the Arduino board-based lead compensators showed a lower bandwidth. One should be cautious about the decrease in bandwidth because lower overall bandwidth of closed-loop systems will lead to slower response. At high frequency, the digital lead compensator gain was lower than that of the analog lead compensator which would attenuate high-frequency noise problems. At low frequency, the digital lead compensator had higher DC gains than the analog lead compensator. Therefore, the DC gains of the forward path employing digital lead compensator will also increase which is beneficial for the steady-state accuracy. To investigate the relative stability, the phase response should be taken into account. By assuming the phase response of Arduino-based lead compensators are nearly identical to that of the analog type, it is likely to maintain or increase the relative stability of the compensated systems [41].

For the Arduino board-based lag compensators, the experiment also showed a lower bandwidth. This finding indicates a contradictory result compared to [42] in which the DSP-based PI controller has higher bandwidth. At high frequency, the digital lag compensator gain was higher than that of the analog lag compensator; therefore any high-frequency noises might still remain. At low frequencies, the DC gain of the closed-loop systems employing the digital lag compensator will be slightly lower than that employing the analog lag compensator. Like the digital lead compensator, the phase response should be taken into account to investigate the relative stability.

By assuming the phase response of Arduino based-lag compensators as being nearly identical to that of the analog type, it is also likely that the relative stability of the compensated systems is maintained or increased.

## 5. Conclusions

The spectrum profiles of the Arduino board based digital lead and lag compensators were investigated. The discretization methods, in particular the bilinear transformation and matched pole-zero mapping methods, affected all the parametric keys with regards to the frequency responses. The DC gains produced by the digital lead compensators were slightly higher than the gains produced by the continuous-time compensators. The cut-off frequencies became lower and the transient slopes turned out to be less steep. Meanwhile, for the digital lag compensators, the DC gains were slightly lower, and the cut-off frequencies became lower. Also, the roll-off slope became less steep. The low-cost Arduino board served as lead and lag compensators to some extent due to the spectrum profile distortion. A trade-off among parameters of the compensators must be made to gain the benefits while sacrificing the objectives that are lower priority. In the future, a comparative study of discretization algorithms implemented on more sophisticated and specific-purpose hardware may be considered.

### Acknowledgements

The authors are very thankful to Satya Wacana Christian University for financially supporting this research under grant No. 121/Penel./Rek/4/V/2019.

### References

- [1] Ogata K., *Modern control engineering*, 5th ed., Prentice Hall, New Jersey, pp. 309–330 (2010).
- [2] Tavazoei M.S., Kakhki M.T., *Compensation by fractional-order phase-lead/lag compensators*, IET Control Theory and Applications, vol. 8, no. 5, pp. 319–329 (2014).
- [3] Zanasi R., Cuoghi S., Ntogramatzidis L., *Analytical and graphical design of lead–lag compensators*, International Journal of Control, vol. 84, no. 11, pp. 1830–1846 (2011).
- [4] Liu J.J.R., Lam J., Xie X., Shu Z., *Generalized lead-lag  $H_\infty$  compensators for MIMO linear systems*, IEEE Transactions on Systems, Man and Cybernetics (2020), DOI: 10.1109/TSMC.2019.2960583.
- [5] Dong Y., *Extreme value condition of stabilizing lead/lag region*, in Proc. of 2nd International Conference on Informatics, Control and Automation, Hangzhou, pp. 371–376 (2019).
- [6] Zanasi R., Cuoghi S., *Direct method for digital lead-lag design: analytical and graphical solutions*, in Proc. of IEEE International Conference on Automation Science and Engineering, Trieste, pp. 804–809 (2011).
- [7] Abo-Hammour Z.S., Saaideh M.I., Alkayyali M., Khasawneh H.J., *Optimal design of lead compensator using nature-inspired algorithms*, in Proc. of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, Amman, pp. 40–45 (2019).
- [8] Monjengue D., *Design and simulation of model-free lag-lead controller for DC motor speed control*, International Journal of Science and Technology, vol. 8, no. 1, pp. 83–89 (2019).

- [9] Kennedy H., *Recursive digital filters with tunable lag and lead characteristics for proportional-differential control*, IEEE Transactions on Control Systems Technology, vol. 23, no. 6, pp. 2369–2374 (2015).
- [10] Ates A., Alagoz B.B., Kavuran G., Yeroglu C., *Implementation of fractional order filters discretized by modified fractional order Darwinian Particle Swarm Optimization*, Measurement, vol. 107, pp. 153–164 (2017).
- [11] Abdulkhader H.K., Jacob J., Matthew A.T., *Fractional-order lead-lag compensator-based multi-band power system stabiliser design using a hybrid dynamic GA-PSO algorithm*, IET Generation, Transmission and Distribution, vol. 12, no. 13, pp. 3248–3260 (2018).
- [12] Montero C.M., Gaspariano L.A.S., Lopez C.S., Gonzalez-Diaz V.R., *On the electronic realizations of fractional-order phase-lead-lag compensators with OpAmps and FPAs*, in Fractional Order Control and Synchronization of Chaotic Systems, 2nd ed., Springer, Switzerland, vol. 688, pp. 131–164 (2017).
- [13] Kapoulea S., Tsirimokou G., Psychalinos C., Elwakil A.S., *Employment of the padé approximation for implementing fractional-order lead/lag compensators*, AEU – International Journal of Electronics and Communications, vol. 120 (2020).
- [14] Dogruer T., Tan N., *Lead and lag controller design in fractional-order control systems*, Measurement and Control, vol. 52, no. 7–8, pp. 1–12 (2019).
- [15] Khasawneh H.J., Abdelaal O., Al-Saaideh M.I., Abo-Hammour Z.S., *Optimal lead compensator for two-loop control system of linear DC motor*, in Proc. of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, Amman, pp. 634–639 (2019).
- [16] Kapoulea S., Tsirimokou G., Psychalinos C., Elwakil A.S., *OTA-C implementation of fractional-order lead/lag compensators*, in Proc. of Novel Intelligent and Leading Emerging Sciences Conference, Giza, pp. 38–41 (2019).
- [17] Qader M.R., *Identifying the optimal controller strategy for DC motors*, Archives of Electrical Engineering, vol. 68, no. 1, pp. 101–114 (2019).
- [18] Miao Z., Lu Z., *A single-cycle-lag compensator based active damping for digitally controlled LCL/LLCL-type grid-connected inverters*, IEEE Transactions of Industrial Electronics, vol. 67, no. 3, pp. 1980–1990 (2020).
- [19] Pan D., Ruan X., Wang X., *Direct realization of digital differentiators in discrete domain for active damping of LCL-Type grid-connected inverter*, IEEE Transactions on Power Electronics, vol. 33, no. 10, pp. 8461–8473 (2018).
- [20] Osornio-Rios R.A., *FPGA lead-lag compensator design for industrial control systems*, Journal of Scientific and Industrial Research, vol. 76, pp. 733–736 (2017).
- [21] Barra H.M., Marcillo K.E.L., Rocha E.M., Nunes M.V.A., Araujo R.B., Junior W.B., *A digital damping controller to increase the safety in hydraulic pumping systems*, in Proc. of IEEE International Conference on Industry Applications (INDUSCON), Sao Paulo, pp. 176–181 (2018).
- [22] Chen S., Wu X., Chen J., Tan G., Wang Y., *Second-order lead compensator-based quadrature PLL for sensorless interior permanent magnet synchronous motor control*, IET Power Electronics, vol. 13, no. 3, pp. 568–575 (2020).
- [23] Garg M.M., Hote Y.V., Pathak M.K., *Design and performance analysis of a pwm dc–dc buck converter using pi-lead compensator*, Arabian Journal for Science and Engineering, vol. 40, no. 12, pp. 3607–3626 (2015).
- [24] Ahmad F., Hameed Z., Rehman S.U., *Efficiency improvement of a hybrid power system using lead-lag Compensator (LLC)*, in Proc. of International Conference on Engineering and Emerging Technologies, Lahore, pp. 1–6 (2020).

- [25] Blum J., *Exploring Arduino: tools and techniques for engineering wizardry*, 2nd ed., John Wiley and Son, Indianapolis (2013).
- [26] Chaouch S. *et al.*, *DC-motor control using Arduino-Uno board for wire-feed system*, in Proc. of International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Algiers (2018).
- [27] Bauer W., Janik A.K., *Implementation of bi-fractional filtering on the Arduino Uno hardware platform*, Theory and Applications of Non-integer Order Systems, Lecture Notes in Electrical Engineering, vol. 407, pp. 419–428 (2017).
- [28] Karar M.E., El-Brawany M., *Embedded heart sounds and murmurs generator based on discrete wavelet transform*, in Proc. of International Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC), Cairo, pp. 34–37 (2016).
- [29] Yosafat S.R., Machbub C., Hidayat E.M.I., *Design and implementation of pan-tilt control for face tracking*, in Proc. of IEEE International Conference on System Engineering and Technology (ICSET), Shah Alam, pp. 217–222 (2017).
- [30] Phillips C.L., Parr J., *Feedback control systems*, 5th ed., Pearson, New Jersey (2011).
- [31] Yepes A.G., Gandoy J.D., Malvar J., *Effects of discretization methods on the performance of resonant controllers*, IEEE Transactions on Power Electronics, vol. 25, no. 7, pp. 1692–1711 (2010).
- [32] Swarnakar J., Sarkar P., Singh L.J., *Realization of fractional-order proportional derivative controller for a class of fractional-order system in delta domain*, in: Choudhury S., Mishra R., Mishra R., Kumar A., (Ed.), Intelligent Communication, Control and Devices, Advances in Intelligent Systems and Computing, vol. 989, Springer, Singapore (2020).
- [33] Jarzebowicz L., Opalinski A., *Frequency and time domain characteristics of digital control of electric vehicle in-wheel drives*, Archives of Electrical Engineering, vol. 66, no. 4, pp. 829–842 (2017).
- [34] Magdy G. *et al.*, *Tustin's technique based digital decentralized load frequency control in a realistic multi power system considering wind farms and communications delays*, Ain Shams Engineering Journal, vol. 10, no. 2, pp. 327–241 (2019).
- [35] Oppenheim A.V., Willsky A.S., *Signals and systems*, 2nd ed., Pearson Education, New Jersey (1996).
- [36] Kowalczyk Z., *Discrete approximation of continuous-time systems: a survey*, IEE Proceedings G (Circuits, Devices and Systems), vol. 140, no. 4, pp. 264–278 (1993).
- [37] Ogata K., *Discrete-time control systems*, 2nd ed., Prentice Hall, New Jersey (1995).
- [38] Smith S.W., *Digital signal processing*, Elsevier Science, Massachusetts (2003).
- [39] Dastjerdi A.A., Vinagre B.M., Chen Y., HosseinNia S.H., *Linear fractional order controllers; A survey in the frequency domain*, Annual Reviews in Control, vol. 47, pp. 51–70 (2019).
- [40] Pérez E.J., Cid A.J.D., Rodríguez P.H., *Analog-based digital low-pass filter realization for mains noise using Arduino Mega 2560 default parameters*, Universidad Tecnológica Centroamericana (UNITEC), Honduras (2016).
- [41] Fekih S., Sfaihi B., Benrejeb M., *On the sampling and the performance comparison of controlled LTI systems*, International Journal of Advanced Computer Science and Applications, vol. 9, no. 3, pp. 218–225 (2018).
- [42] Rashmi R., Jagtap S., *A comparative study of phase margin based digital and analog controlled synchronous buck with optimum LC filter*, COMPEL – The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 38, no. 6, pp. 2020–2039 (2019).