International Journal of Advanced Culture Technology Vol.10 No.4 518-528 (2022)
DOI https://doi.org/10.17703/IJACT.2022.10.4.518

IJACT 22-12-61

Design and Implementation of Continuous Control for Household Electric Fan Speed for Virtual Reality Applications

[1]Jonas John Claud, [2]Dae-Young Na, [3]Daseong Han

[1]Graduate student, Dept. of Advanced Convergence, Handong Global Univ., Korea
[2]Prof., School of Global Leadership, Handong Global Univ., Korea
[3]Prof., School of Global Entrepreneurship and ICT, Handong Global Univ., Korea
claudezjohn@gmail.com, soriru@handong.edu, dshan@handong.edu

Abstract

Virtual Reality (VR) has been widely used in various applications to generate realistic virtual environments. A sense of immersion can be increased by providing additional stimuli such as tactile sensation to VR contents. However, it is still challenging to provide a realistic feel for the wind blowing over the whole body by smoothly controlling the airflow. To address this issue, we employ a household electric fan as a wind generating device to provide users with wind experience in VR environments. The wind generating device targets the whole body to mimic the wind we feel outside in our daily life. To do so, we present a low-cost method to smoothly control household fan speed using an Arduino microcontroller. Here, we use the Sinusoidal Pulse Width Modulation (SPWM) technique to generate the sinusoidal voltage required to drive the fan motor. Our experimental results show how Variable Voltage Variable Frequency (VVVF) is implemented at a low cost using our method for household fan speed control. The results can be applied to various VR applications to enhance the sense of immersion by providing users with realistic wind.

Keywords: *Virtual Reality, Wind, Household Electric Fan, SPWM, Arduino*

1. INTRODUCTION

Adding tactile sensations to Virtual Reality (VR) content is known to enhance enjoyment and immersion [1-3]. Vibration and wind have drawn much attention as stimuli that cause tactile sensation in VR experiences. Although vibration increases immersion, it requires wearing additional devices to stimulate the body, which is heavy and expensive, making it inconvenient [4]. On the other hand, natural wind generally causes pleasant feelings and can be produced from an unwearable device. The wind generation in virtual reality applications can be used to simulate sailing [5], sports training, such as archery [6] and football, and even weather conditions [7-8]. However, reproducing a smooth airflow in a simulated environment remains a challenge. Most research focuses on small-scale wind generating devices that allow wind to blow only to the head [9-11]. This is because the head is one of the most sensitive body parts, which can lead to a great sense of immersion when the wind is applied to it. Rapid airflow generation close to reality has been achieved using air nozzles [11] and mini fans [7,12].

Manuscript received: November 26, 2022 / revised: December 4, 2022 / accepted: December 9, 2022
Corresponding Author: dshan@handong.edu
Tel:*** - **** - ****
Professor, School of Global Entrepreneurship and ICT, Handong Global University, Korea

Despite the improvement in immersion by these small-scale systems, there is still a demand for large-scale systems that can give a feeling of natural wind to the whole body as in a natural environment, not just the head only. In one study, researchers developed a system that uses a silent hair drier to generate airflow while varying the temperature in a treadmill to let the user experience the movements also [13]. Household fans have been used to generate airflow for large-scale systems in virtual reality applications [14-16]. However, most large-scale systems suffer from latency in responding to quick changes in airflow, the cost, and complicated control mechanisms such as using light dimmers [16]. Unlike these existing systems, we aim to reduce the cost of building large-scale systems while providing a user with safety, awareness, and fewer input controls. We employ a household electric fan (EF-2080) since it is affordable and readily available in the domestic environment. It is also ergonomically designed to provide wind to humans. Moreover, we provide detailed design and implementation techniques to control the fan's speed to ensure the required airflow generation unlike most studies.

2. METHOD AND DESIGN

To reproduce the smooth variable airflow, we modify the speed control mechanism of the household electric fan. Traditionally, the household electric fan uses an induction motor to rotate the blades to generate airflow. It is affordable in construction and has durability in general. To continuously adjust the airflow by changing the speed of the fan motor, therefore, the operating frequency should be varied. The relationship between the speed and the frequency is shown in Equation (1).

$$\boldsymbol{Ns} = \frac{\boldsymbol{120f}}{\boldsymbol{p}} = \boldsymbol{N - slip} \tag{1}$$

Here, $\boldsymbol{Ns}$ denotes the motor's synchronous speed, which is directly proportional to operating frequency $\boldsymbol{f}$ given number of poles $\boldsymbol{p}$. The synchronous speed is always a little less than the actual speed $\boldsymbol{N}$ at which the motor rotates due to $\boldsymbol{slip}$, which helps to keep the motor's rotation by the differential in magnetic flux between the rotor and stator parts of the motor.

The Sinusoidal Pulse Width Modulation (SPWM) technique is utilized to produce a variable frequency by varying the duty cycles of several high sampled frequency voltages in a sinusoidal sequence until it fits the operating frequency. A cheap microcontroller (Arduino Uno SMD) is used to produce SPWM signals at 16 kHz each. The SPWM is adopted considering the characteristics of the induction motor. Therefore, we select a phase and frequency correct mode of Arduino timer/counter interrupt (TIMER1) to ensure each signal generated has the same center, which enables the motor to receive the signal of the same phase, as shown in Figure 3. Reducing the duty cycle of all PWM signals by a certain percentage in a half period (of operating frequency) results in a reduction of the average voltage, whereas reducing the number of PWM signals in the half period (of the operating frequency) reduces the operating frequency. It should be noted that adding the number of PWM signals while fixing the operating frequency increases the resolution of the sine wave to be as close to the traditional sine wave as possible and vice versa. However, high-frequency harmonic components can be introduced into the system. To efficiently control the induction motor while changing the speed by varying the frequency, the ratio between voltage and frequency (v/f ratio), which is 3.667 (220 V/60 Hz), is always kept constant. We employ the H-bridge configuration and a half unipolar SPWM topology in which the one-side in-series switches receive a high or low DC signal, and the other side in-series switches receive the SPWM. This topology reduces the overall switching losses of the MOSFETs and saves the instruction execution time, as only two output pins of the controller are used to generate the SPWM.

The hardware components used are Arduino Uno R3 SMD as the system's main computing component and SPWM generator, the bench switch mode power supply (KEITHLEY, 2231A-30-3 Triple-channel DC Power Supply, 2015, Tektronix) as the power source, the dual optocoupler device (HCPL-2630, FAIRCHILD SEMICONDUCTOR) to provide isolation between Arduino and high power side of the system, the high and low side gate driver (IR2113(S)PBF, 14 Lead PDIP, INFINEON) used to provide a sufficient amount of current to quickly turn on and off the MOSFETs, the MOSFETs (IRF830) power switches to produce high AC voltage to the fan's motor, and the LC filter circuit to smooth the generated AC voltage. We use four N-channel MOSFETs because of their minimal power losses and the same properties. The MOSFETs are configured in an H-bridge configuration, allowing the motor to receive the DC signals in both directions, imitating AC voltage.

The Arduino software (Arduino IDE version 1.8.16, 2021.09.06) is used to write and upload the code to the ATmega328 chip. The programming language used is the industry standard C++ implemented by the GNU g++ compiler. In the design, no Prescaler mode is set to allow the whole 16 MHz (62.5 nanoseconds per instruction) internal clock of the Arduino to be utilized. This is because the SPWM signals are generated at every 16 kHz (62.5 μseconds), and other calculations, such as determining the input value (desired frequency from the airflow input speed), dead bands (to prevent short circuits of the MOSFETs), v/f ratio, and output value, are necessary for a short period. Arduino PWM pin 9, pin 10, pin 11, and digital pin 12, defined as PORTB, are used as the outputs. Instead of pin numbers, the registers are employed to reduce the execution time (translating the pin to the PORT register) and allow the code to be implemented in any ATmega328 chip-based microcontroller. Figure 1 shows the workflow of control software, including the input frequency (desired frequency) representing the desired airflow, the process from waveform generation, Arduino interrupt initializations for producing the SPWM signals, a soft start to the frequency and voltage generation process, and the variable frequency corresponding to the output frequency resulting in fan speed.

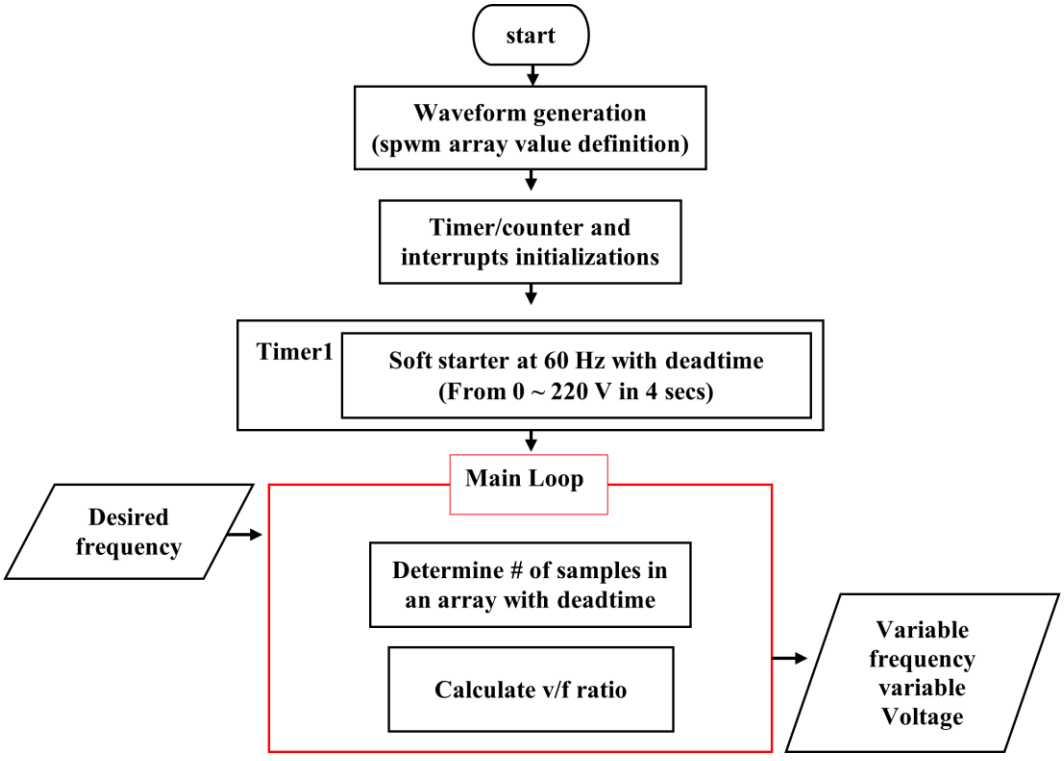

Figure 1. Software design flowchart

The critical design features for the software are the waveform generation utilizing the ATmega328 registers, safety features such as deadtime and soft-start, and the v/f ratio. Although the discussion (Section 4) shows direct approaches in programming, few optimizations are considered due to the limitations of the actual hardware devices.

3. EXPERIMENTAL RESULTS

In experiments, one of our implementation targets is an SPWM inverter that uses a 220 VAC power source from the socket wall. Thus, we perform a simulation for the design of the rectifier and Power Factor Correction (PFC). However, we only implemented the simple rectifier circuit, which successfully converts 220 VAC to 305 VDC. This shows that the inverter can receive enough power to generate about 210 VAC to power the household electric fan, considering the inverter power losses. The rectifier consists of a fuse, a bridge rectifier (GBJ5006-BP, 600 V), a bulk power inductor (546-159ZG, 15 mH, 4 A) to reduce the inrush current during capacitor charging, and an electrolytic capacitor (2200 µF, 400 VDC) as shown in Figure 2. In the future study, the rectifier will be modified by adding PFC and safety features (safety capacitors) to power the inverter.

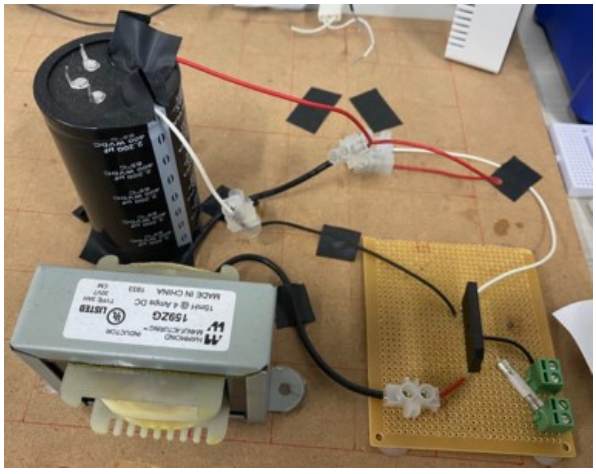

Figure 2. The rectifier circuit

16 kHz,
6 Vp-p

Figure 3. The SPWM generated signals in phase and frequency correct mode

Due to this, the SMPS was used to provide the power source to the system. The signals produced by the SPWM generator were visualized using the oscilloscope (Tektronix, TBS1102B-EDU). The signals generated through Arduino output pin 9 and pin 10, and pin 11 and pin 12 are for SPWM signals and for high and low

levels, respectively. As discussed in Section 2, the phase and frequency correct was implemented in pin 9 and pin 10. As shown in Figure 3, the yellow signal represents the SPWM signal in increasing the duty cycle (non-inverting mode) while the blue signal represents the SPWM signal in decreasing the duty cycle (inverting mode). These signals were measured from the Arduino pins 9 and 10, respectively. The switching frequency is 16 kHz at each signal.

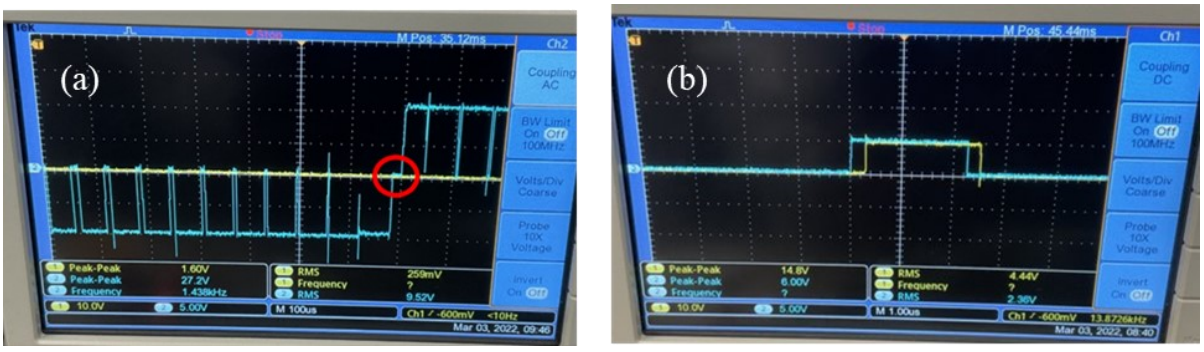

Figure 4. (a) Deadtime at the half period (b) Delay from Arduino output (blue) to gate driver output (yellow)

A dead time of 1 µs between PWM signals was added to prevent cross-conduction. Moreover, the dead time that was also introduced during the half period is visualized as shown in Figure 4(a) and is circled in red. We measured the signals that were outputted by the gate driver and compared them to those that the Arduino outputted to see whether they were the same. The overall turn-on latency from Arduino to the gate driver's output (IR2113) is approximately 250 ns, and the turn-off delay is nearly the same amount of time. Neither signal will be missed even at a duty cycle of 1%, which corresponds to 625 nanoseconds. The comparison of the output signals from the Arduino and gate driver is shown in Figure 4(b) where each division in the x-axis is at the scale of 1 µs.

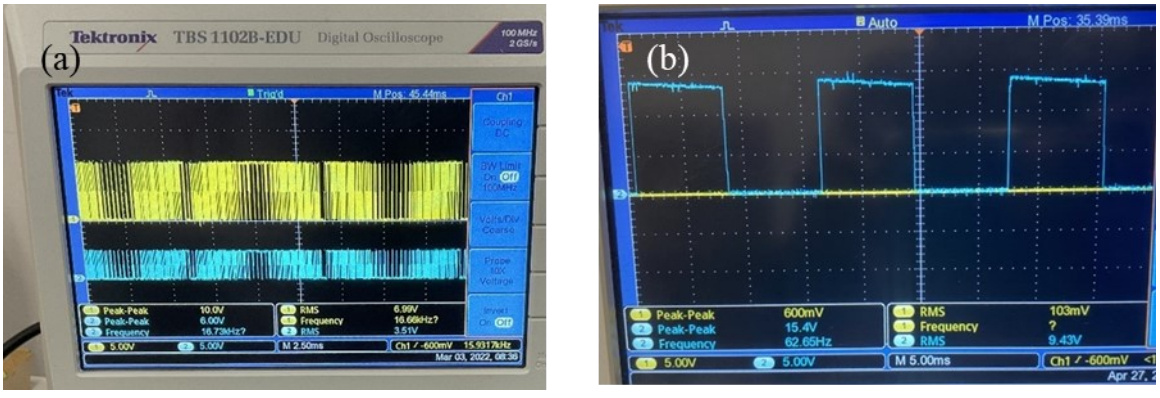

Figure 5. (a) High-side (yellow line) and low-side (blue line) gate driver's outputs (b) High-side gate driver PWM signal from Arduino pin 11

In addition, due to the bootstrap mechanism of the gate driver, the high-side driver output has a higher voltage peak than the low-side driver's output. The mechanism makes it possible to drive a load that is situated beneath the N-Channel MOSFET (in an H-bridge configuration) by boosting the gate-source voltage to be higher than the load voltage while maintaining the gate-source threshold for the N-channel MOSFET to operate. The high-side and low-side gate drivers were visualized by providing a 10 V to a 10 kΩ load resistor.

This resulted in 20 V being displayed at the output of the high-side gate driver as we applied 10 V as the gate voltage. The output at the low-side gate driver remained at 10 V as the gate voltage. The comparison of the high-side and low-side outputs of the gate driver is shown in Figure 5(a). Figure 5(b), however, demonstrates a fall in voltage with time. This indicated that a bootstrap capacitor is discharged over time to boost high-side gating. The bootstrap capacitor loses charge because of a prolonged-ON duration. A high capacitance capacitor replacement, nevertheless, can improve this.

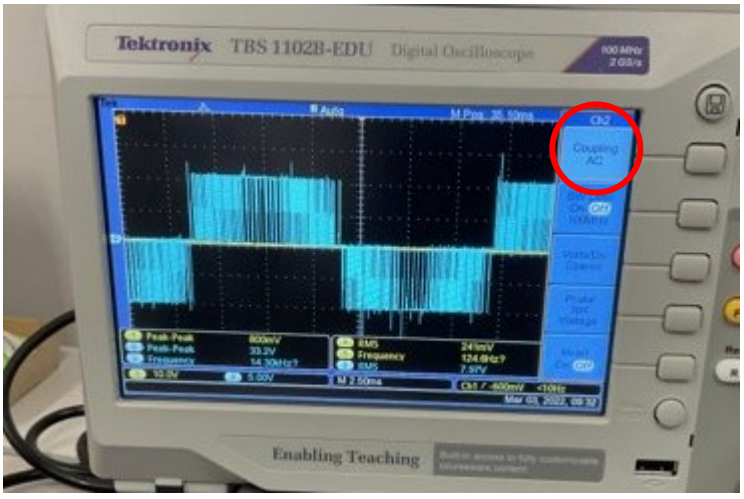

Figure 6. The SPWM signal output in an H-bridge configuration

Finally, the MOSFETs were arranged in an H-bridge configuration, and the SPWM signals were measured. The measurement was achieved by attaching the oscilloscope probes to the H-bridge output and by setting the received signal in the oscilloscope as an AC signal. The result is shown in Figure 6. The red circle indicates that the probe input coupling was set as AC.

3.1 LC Filter

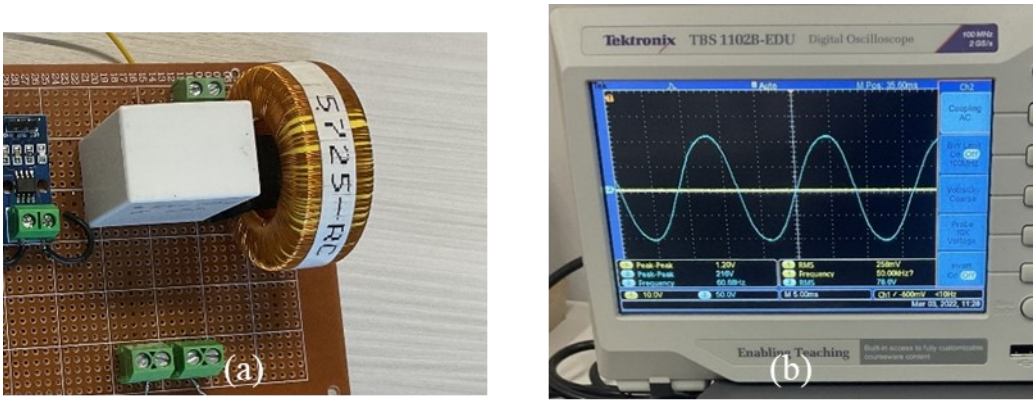

Figure 7. (a) The LC filter circuit (b) The pure sine wave signal at 60 Hz measured at the output of the LC filter

An in-depth exploration of the SPWM signals shown in Figure 6 demonstrates a stepwise concentration of the signals; this concentration is simply a reflection of the sinusoidal wave. To reveal the sinusoidal waveform, we added an LC filter at the H-bridge output that consisted of a capacitor (2.5 µF, C4AF3BU4250T12K) and

an inductor (8 mH, 5725-RC) used as a low-pass filter to filter out the higher frequency components and remain with the lower frequencies to produce a pure sine wave signal. The inductor is connected in series with the load, whereas the capacitor is connected parallel to the output load. This configuration is shown in Figure 7(a). We chose a cut-off frequency of 1125 Hz after considering that a 3 dB attenuation would occur at 563 Hz, which is still quite far from our operating frequency (60 Hz). Since the switching frequency is high, there is a reduced need for components that act as filters. The phase matching of the voltage and current that results from using an inductor and a capacitor is another significant advantage of this configuration. The measurement at the LC filter output shows the sinusoidal waveform as the one at the socket wall in a domestic environment, as shown in Figure 7(b).

Figure 8. (a) Soft start; 40% voltage at 58 Hz (b) 100% voltage at 60 Hz

As a result, a pure sine wave identical to that of the household was successfully regenerated. However, voltage and frequency are controllable with the sine wave generated, unlike the household's fixed frequency and voltage. Therefore, from this point, we have room to control the motor speed to meet our requirements. The generated sine wave has a soft start feature which reduces the inrush motor current at the start of the motor. The soft start allows the voltage to increase slowly from 0 V to the maximum in 4 seconds. As shown in Figures 8(a) and 8(b), the voltages are at 40% at 1.6 sec after the start of the motor and 100% at 4 sec, respectively. During the soft start procedure, the frequency was kept at 60 Hz; thus, the motor experienced less voltage, reducing its initial current draw.

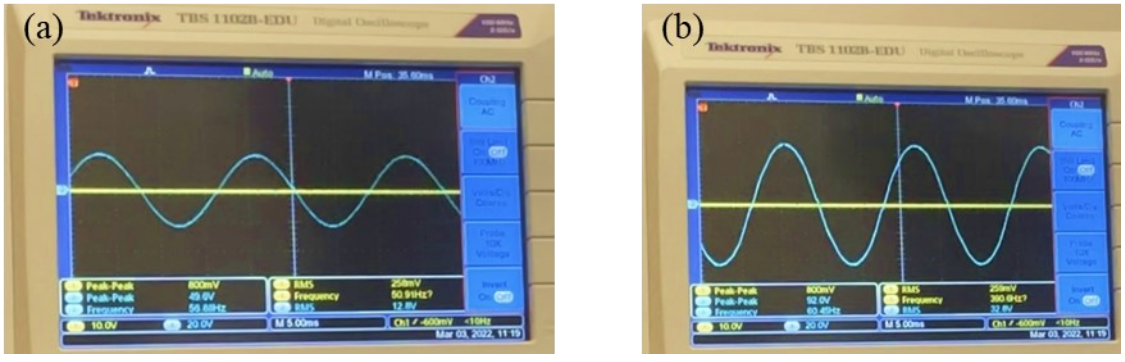

Figure 9. v/f control at 34 Hz (a) and 74 Hz (b)

Similarly, during the frequency control, the voltage was varied respectively following the v/f ratio. In the experiment, we varied the frequency from 25 Hz to 75 Hz. While applying the v/f control to the motor, there are a few concerns to address; there is a minimum voltage at which the motor requires to run, therefore, at that

threshold voltage, the voltage was kept constant, and the frequency was further reduced. While at the rated voltage and frequency (220 VAC, 60 Hz), the voltage was kept constant, and the frequency was increased for higher speeds. In the design, this was considered, and the lower and higher voltage limits were set to 30% and 100% of the rated voltage, respectively. We visualize the frequency control, which follows the v/f balance as shown in Figure 9(a) and (b) by showing the frequency at 34 Hz and 74 Hz, respectively.

3.2 Load Testing

We perform soldering to allow easy connection of all the system components. Here, the solderable breadboard was used. We used a soldering station (HAKKO, FX-888D) set to 400 degrees Celsius and a solder core wire measuring 1 millimeter in diameter. To cut down on the amount of direct soldering of IC components (gate drivers and optocouplers), the IC socket pins (8-pins and 16-pins) were used. This is necessary in the event that the IC malfunctions or burns. It makes it easier to replace and reduces the time the IC is exposed to high temperatures during soldering, which can cause burning. Figure 10 consecutively shows the complete soldered hardware device represented in the rear and front views.

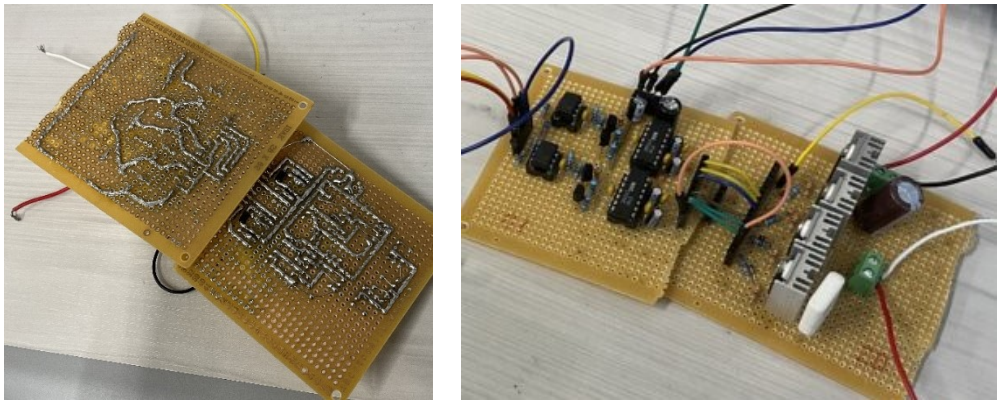

Figure 10. The rear view (left Figure) and the front view (right Figure) of the complete device

To test the system's performance, instead of using the household electric fan, the 30 VAC-powered motor (RK-384CA-22572A) was utilized as a load. This is due to the shortcoming of the power source required to generate high-voltage DC for the inverter. The rectifier circuit in Figure 2 has some restrictions that must be dealt with. As a result, we could not employ it to serve as a power source for the inverter. Instead, the SMPS was used to power the load through the H-bridge at 20 VDC. The results revealed a change in the speed of the motor. The speed of the motor slowed down because of the frequency (with v/f) being lowered. However, the effectiveness of the speed control was not noteworthy since the construction of the motor is not directly dependent on the frequency. Both DC and AC voltage can be used to power this motor.

Although we did not have access to a high-voltage source, we could still investigate how our system performs when subjected to high voltage by using a step-up transformer (ht-601, SONA, 0-12 VAC: 0-220 VAC) and a filter capacitor (2.5 µF, C4AF3BU4250T12K) to step up the VDC power that the SMPS provided to a smooth sine wave of 220 VAC as shown in Figure 11(a). It is important to note that the load was not applied to this 220VAC because of safety concerns and SMPS's limitation of only offering a limited amount of power. The transformer's output measured was 220 VAC, as shown in Figure 11(b). The frequency control and soft start at the transformer's output were also tested using the Multimeter.

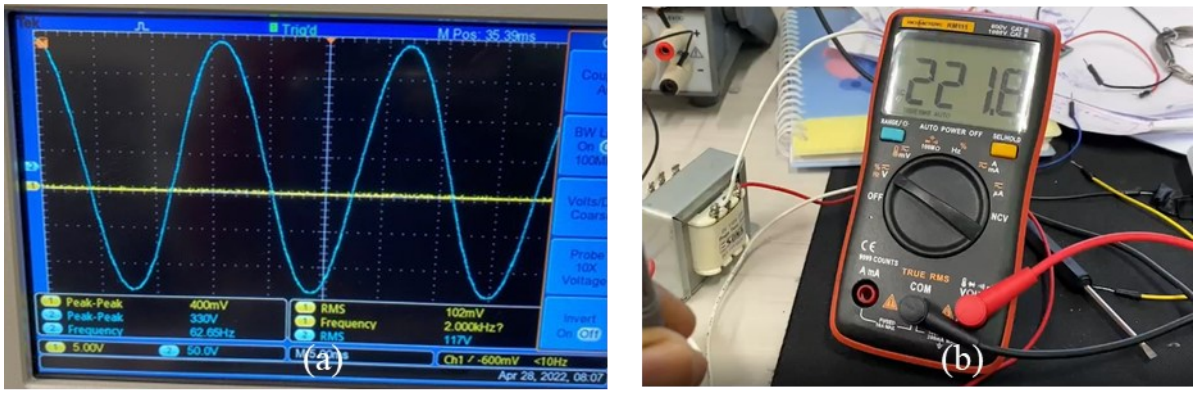

Figure 11. (a) The stepped-up voltage visualization using an oscilloscope (b) The multimeter measurement of the stepped voltage

4. DISCUSSION

The practical design and implementation of the system resulted in an understanding of the factors that guide the study's implication and improvement. This comprehension has prompted a discussion about extensive analysis, safety practices, approach requirements, and comparisons with previously established methods.

In research on how wind generating devices can make virtual reality experiences more immersive, most studies have relied on simulated wind data and video content. On the other hand, when it comes to controlling the airflow of the wind generating device by utilizing an induction motor fan, most studies have provided simulation results that show the generation of the controllable sine wave by using SWPM techniques that are simulated by the built-in simulation software parameters—the simulations give an intuitive realization. The findings bring attention and reveal design challenges and methods for optimizing performance to improve it. Nevertheless, only a few studies have implemented their simulated results and found them consistent with their simulations. In addition, the primary focus of these investigations has been on utilizing step-up transformers to produce high-output sine waves with a constant frequency and voltage from either solar power or DC batteries.

However, our system demonstrated the anticipated sine wave control as hypothesized in the control of the household fan. We intuitively designed the control through Arduino's phase and frequency correct mode, which considers the fan motor’s control as we purposefully target to utilize it to generate variable airflow. Even though the system design and implementation were successful, the current method has yet to prove airflow generation because it was not tested with the household fan. To determine its level of performance, it is anticipated that the SPWM-based speed control will be put through its pace in a test. This will be made possible by intuitively designing the rectifier circuit and providing a power source from the socket wall in a domestic environment. This contrasts with a conventional inverter that uses batteries or solar power to function correctly. During this procedure, concerns about the user's safety will be prioritized. In addition to that, the PCB will be utilized. It will reduce electromagnetic interferences and harmonics caused by high frequencies and improper component soldering with hand soldering.

5. CONCLUSION

It is necessary to address that most inverters operate in fast mode, which allows signals to be generated in a shorter amount of time. However, for motor control, induction motors draw current out of phase. Therefore, it is preferable to have an in-phase switching frequency in the design to avoid output phase complexity. Because

of this, our design is safer for the motor as well. It is less expensive and more compact due to its use of a microcontroller, which, in comparison to other solutions, eliminates the need for a significant number of circuits to generate a sine wave. It is considered to operate on a single-phase power source, which offers advantages for the control of induction motors used in residential applications, such as those that require automatic speed variations or precise speed variation requirements. Because it can be programmed, there is room for flexibility regarding changes and improvements. Furthermore, our controller is isolated from the primary power source, which brings about safety to the controller, which consists of all the instructions, and the operator is safe from high voltages when using the device.

In our previous study [17], we developed a wind generating system using Arduino and relays to control the household fan. We measured the user's emotional states when experiencing virtual reality with the wind generated by the household fans. The result revealed that the positive emotions were significantly increased when there was wind compared to its absence. Along with an improvement in the control of airflow from this study, the entire system will make it possible to experience the natural environment while in a different location. Virtual reality (VR) can make it easier for people of all ages to enjoy themselves remotely, including the disabled and the elderly. In this scenario, they would be able to engage in tourism activities such as visiting the recorded caves experience while remaining in their secure and remote location to experience an authentic view and sense of wind. Some research has captured natural scenery data from the natural environment, such as forests, the sea, and the sky. It showed that the feeling of the natural environment in the virtual reality experience could be used to heal mental conditions in people of all ages [18]. Further, adding environmental conditions like the wind will create a realistic immersion and increase the healing and enjoyment level.

REFERENCES

[1] G. Zwoliński, D. Kamińska, A. Laska-Leśniewicz, and Ł. Adamek, "Vibrating Tilt Platform Enhancing Immersive Experience in VR," *Electronics*, vol. 11, no. 3, pp. 462, doi: https://doi.org/10.3390/electronics 11030462, Feb. 2022.

[2] Moon, T., and Kim, G.J., "Design and evaluation of a wind display for virtual reality*," In Proceedings of the ACM symposium on Virtual reality software and technology (VRST '04),* Association for Computing Machinery, New York, NY, USA, pp. 122–128, doi: https://doi.org/10.1145/1077534.1077558, November 2004.

[3] Lee, H. K., Yoon, N., and Choi, D., "The effect of touch simulation in virtual reality shopping." *Fashion and Textiles*, *9*(1), 34. https://doi.org/10.1186/s40691-022-00312-w, 2022.

[4] S. Jung, R. Li, R. McKee, M. C. Whitton, and R. W. Lindeman, "Floor-vibration VR: Mitigating Cybersickness Using Whole-body Tactile Stimuli in Highly Realistic Vehicle Driving Experiences," *in IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2669-2680, doi: https://doi.org/10.1109/TVCG.2021.3067773, May 2021.

[5] V. Jouke, M. Fabian, V. Joris, J. Anna, K. Darina, N. Zsuzsa, L. Bob, and S. Paul, "Enhancement of Presence in a Virtual Sailing Environment through Localized Wind Simulation", *Procedia Engineering*, no. 60, pp. 435-441, doi: https://doi.org/10.1016/j.proeng.2013.07.050, September 2013.

[6] F. A. Purnomo, M. Purnawati, E. H. Pratisto and T. N. Hidayat, "Archery Training Simulation based on Virtual Reality," *2022 1st International Conference on Smart Technology*, *Applied Informatics, and Engineering (APICS),* pp. 195-198, doi: https://doi.org/10.1109/APICS56469.2022.9918716, August 2022.

[7] N. Ranasinghe, P. Jain, S. Karwita, D. Tolley, and E. Y. Do, "Ambiotherm: Enhancing Sense of Presence in Virtual Reality by Simulating Real-World Environmental Conditions", *In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17).* Association for Computing Machinery, New York, NY, USA, pp. 1731–1742, doi: https://doi.org/10.1145/3025453.3025723, May 2017.

[8] N. Ranasinghe, P. Jain, N. T. N. Tram, K. C. R. Koh, D. Tolley, S. Karwita, L. Lien-Ya, Y. Liangkun, K. Shamaiah, C. E. W. Tung, C. C. Yen, and E. Yi-Luen Do, "Season Traveller: Multisensory Narration for Enhancing the Virtual Reality Experience," *In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18).* Association for Computing Machinery, New York, NY, USA, Paper No. 577, pp. 1–13, doi: https://doi.org/10.1145/3173574.3174151, Apri 2018.

[9] FEEL REAL – The world's first multisensory VR mask, *https://www.gadgetany.com/feelreal-vr-mask/*, Accessed: 28th November 2022.

[10] S. Cardin, D. Thalmann, and F. Vexo, "Head Mounted Wind," *proceeding of the 20th annual conference on Computer Animation and Social Agents (CASA2007),* pp. 101-108,2007, URL: http://infoscience.epfl.ch/record/104359, 2007.

[11] Rietzler, M., Plaumann, K., Kränzle, T., Erath, M., Stahl, A., and Rukzio, E., "VaiR: Simulating 3D Airflows in Virtual Reality," *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems,* pp. 5669-5677, doi: https://doi.org/10.1145/3025453.3026009, May 2017.

[12] D. Tolley, T. N. T. Nguyen, A. Tang, N. Ranasinghe, K. Kawauchi, and C. C. Yen, "Windywall: Exploring creative wind simulations," *In Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '19, Tempe, Arizona, USA: Association for Computing Machinery, pp. 635–644, ISBN: 9781450361965, March 2019.

[13] R. Troost, *Exploring the value of adding airflow to the VR-developer's toolkit*, Master's dissertation, School of Natural Sciences, Technology and Environmental Studies, June 2019.

[14] G. Giraldo, M. Servières and G. Moreau, "Perception of Multisensory Wind Representation in Virtual Reality," *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 45-53, doi: https://doi.org/10.1109/ISMAR50242.2020.00024, November 2020.

[15] Heinz, M., Mucha, H. and Röcker, C., "Extending HMD-based Virtual Reality through Wind and Warmth," *In: Dachselt, R. & Weber, G. (Hrsg.), Mensch und Computer 2018 - Workshopband. Bonn: Gesellschaft für Informatik e.V.*, DOI: https://doi.org/10.18420/muc2018-ws07-0465, September 2018.

[16] F. Hülsmann, N. Mattar, J. Fröhlich, and I. Wachsmuth, "Simulating wind and warmth in virtual reality: Conception, realization, and evaluation for a cave environment," *Journal of Virtual Reality and Broadcasting,* vol 11, No. 10, December 2014.

[17] Ye-Sol Cho, Ye-Won Lee, Do-Jeon Lim, Tae-Dong Ryu, John Claud Jonas, Dae-Young Na and Handa-Sung. "Influence of data-based artificial wind on static emotion in yacht VR experience," *Journal of the Korean Computer Graphics Society,* vol. 28, no.3, 67-77, doi: https://doi.org/10.15701/kcgs.2022.28.3.67, July 2022.

[18] M. Kim, "A Study of Development and Production of Relaxing VR Content," *International Journal of Advanced Culture Technology*, vol. 9, no. 4, pp. 194–203, doi: https://doi.org/10.17703/IJACT.2021.9.4.194, Dec. 2021.