© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

Speed Control of Induction Motor Using Arduino-Based Variable Frequency Drive (VFD): A Literature Review

Saurabh Suresh Shinde[1], Mangesh Dhananjay Nerkar[2], Abhishek Vilas Kumre[3], Akash Dinesh Chopkar[4], Prof. K. R. Patil[5]
[1,2,3,4]*Department of Electrical Engineering, Government College of Engineering, Jalgaon, Maharashtra, India*
[5]*Guide, Faculty, Department of Electrical Engineering, Government College of Engineering, Jalgaon*

***Abstract*—The three-phase induction motor remains the dominant electromechanical energy conversion device in industrial and commercial applications, yet its inherent fixed-speed characteristic under direct-on-line starting has long imposed significant energy and process inefficiencies. Variable Frequency Drives (VFDs) address this constraint by enabling precise control of motor speed through regulated variation of both supply frequency and voltage. In recent years, microcontroller platforms particularly those based on the Arduino ecosystem have emerged as accessible, low-cost alternatives to dedicated digital signal processors for implementing VFD control algorithms. This paper presents a comprehensive and critical literature review of research published between 2015 and 2025 that investigates speed control of induction motors using Arduino-based VFD systems. The review covers the theoretical underpinnings of induction motor speed control, VFD architecture, Sinusoidal Pulse Width Modulation (SPWM) generation on Arduino hardware, and real-world experimental implementations ranging from single-phase to three-phase systems. Studies by Rusdi et al. (2022), Tabassum et al. (2017), Kannan et al., Khan et al. (2023), and several others are critically analyzed and compared. Findings indicate that Arduino-based VFDs are capable of achieving smooth speed control across a wide frequency range with acceptable Total Harmonic Distortion (THD) values, particularly for low-to-medium power applications in the range of 0.1–2.2 kW. Key limitations include restricted current-handling capacity, absence of robust protection schemes, limited closed-loop precision, and sensitivity to electromagnetic interference. Future research opportunities are identified in the areas of sensorless vector control, IoT-enabled remote monitoring, machine-learning-based parameter tuning, and renewable energy integration.**

***Index Terms*—Arduino microcontroller, induction motor drives, pulse width modulation, sinusoidal PWM, speed control, V/f control, variable frequency drive.**

I. INTRODUCTION

The three-phase squirrel-cage induction motor (SCIM) is the most widely deployed electric motor in industry. Estimates consistently indicate that induction motors account for approximately 40–50% of total electrical energy consumption in industrialized nations, and between 60–70% of all motors installed in manufacturing environments are of the squirrel-cage induction type [1]. Their dominance stems from a well-recognized set of attributes: rugged mechanical construction, absence of brushes or commutators, low maintenance requirements, self-starting capability, relatively low cost, and satisfactory efficiency under rated-load conditions. However, the conventional method of operating induction motors at a fixed synchronous speed introduces considerable energy waste when the driven load requires sub-rated speeds, a situation that applies to a substantial proportion of pump, fan, compressor, and conveyor applications [2]. In pumping and fan systems governed by affinity laws, power consumption scales as the cube of rotational speed. Consequently, even a modest 20% reduction in speed yields a theoretical power saving of approximately 49%. The development of solid-state power electronics specifically insulated-gate bipolar transistors (IGBTs) and power MOSFETs enabled fully electronic VFDs that regulate motor speed by simultaneously adjusting inverter output frequency and voltage, preserving air-gap flux linkage and thereby maintaining torque capability across the operating range [3]. Traditional methods such as pole changing, stator voltage control, and rotor resistance insertion are progressively being replaced by inverter-based drives offering superior efficiency and dynamic response.

Contemporary industrial VFDs employ dedicated digital signal processors (DSPs) or field-programmable gate arrays (FPGAs) to execute high-speed control algorithms. While these platforms deliver exceptional performance, their cost and complexity place them beyond the reach of small-scale educational institutions, low-power agricultural or off-grid applications, and research prototyping environments in developing economies. The Arduino family of open-source microcontroller boards notably the Uno (ATmega328P), Nano (ATmega328P), Mega 2560 (ATmega2560), and Due (ARM Cortex-M3) offers an extraordinarily accessible alternative [4]. With built-in hardware PWM channels, programmable timer-counter units, and a unit cost often below USD 25, these platforms have catalyzed a wave of experimental VFD research. Arduino boards generate SPWM signals by traversing pre-computed sinusoidal lookup tables stored in program flash memory through hardware PWM channels at rates set by configurable timer interrupt service routines [5].

This literature review has four primary objectives: (i) to consolidate and critically synthesize published research on Arduino-based VFD systems for induction motor speed control from 2015 to 2025; (ii) to evaluate the suitability of different control methodologies particularly scalar V/f and SPWM-based approaches as implemented on Arduino hardware; (iii) to identify consistent limitations and performance bottlenecks across the reviewed corpus; and (iv) to delineate promising directions for future investigation. The review encompasses single-phase and three-phase systems, open-loop and limited closed-loop implementations, hardware prototypes, and simulation-based validations. The paper is organized as follows: Section II reviews fundamental induction motor speed control theory; Section III details VFD architecture and Arduino integration; Section IV presents the core literature review; Section V gives a comparative analysis; Section VI discusses challenges and future directions; and Section VII concludes the paper.

II. FUNDAMENTALS OF INDUCTION MOTOR SPEED CONTROL

A. Equivalent Circuit and Speed Equations

The per-phase steady-state equivalent circuit of a three-phase induction motor models the stator resistance Rs and leakage reactance Xls in series with the magnetizing branch (Rc and Xm), followed by the referred rotor branch comprising R’r/s and X’lr, where s denotes the per-unit slip [6]. The synchronous speed of the rotating magnetic field in RPM is given by:

$$N_s = 120\, f / P \quad (1)$$

where f is the supply frequency in hertz and P is the number of stator poles. The rotor mechanical speed is $N_r = N_s (1 - s)$, with slip defined as:

$$s = (N_s - N_r) / N_s \quad (2)$$

Under the simplified equivalent circuit, the electromagnetic torque as a function of slip is [7]:

$$T_e = (3/\omega_s) \cdot V_s^2(R'_r/s) / [(R'_r/s)^2 + X'_{lr}{}^2] \quad (3)$$

where $\omega_s = 2\pi f$ is the synchronous angular frequency and Vs is the per-phase stator terminal voltage. This expression yields the classical speed-torque characteristic: an approximately linear torque-slip region at small slips (stable operating region) and a decreasing torque region beyond the pull-out torque at larger slips [8].

B. Classical Speed Control Methods

Several classical techniques exist for altering induction motor speed. Pole-changing permits only discrete speed steps by reconnecting stator windings and requires specially wound motors. Stator voltage control reduces torque capability proportional to V_s^2, limiting it to fan/pump loads. Rotor resistance insertion applies only to wound-rotor motors and dissipates energy resistively. Slip power recovery schemes improve efficiency but require expensive converter hardware [9]. The variable frequency drive (V/f control) simultaneously varies supply frequency and terminal voltage such that their ratio remains approximately constant, maintaining air-gap flux at rated value [10]:

$$\Phi_m \approx V_s / f = \text{constant} \quad (4)$$

This constant volts-per-hertz (V/f) strategy preserves the motor’s torque-producing capability across the speed range. At very low frequencies, the stator resistance voltage drop IsRs constitutes a significant fraction of the applied voltage, causing flux weakening. Practical implementations address this with a low-frequency voltage boost [11]:

$V(f) = (f/f_{rated}) + \Delta V_{boost}$ for $f < f_{rated}$ (5)

III. VARIABLE FREQUENCY DRIVE PRINCIPLES AND ARDUINO INTEGRATION

A. VFD Architecture

A conventional VFD comprises four functional stages [12]. The rectifier stage converts AC mains to uncontrolled DC via a diode bridge; for a single-phase 230 V (rms) input, the theoretical mean DC link voltage is $V_{DC} = \sqrt{2} \times 230 \approx 325$ V. The DC link electrolytic capacitor (typically 470–2200 µF) filters the rectified voltage. The inverter stage is a three-phase bridge of six switching devices (IGBTs or MOSFETs) whose gate signals are controlled by the microcontroller. The control unit executes the SPWM algorithm and an output LC filter attenuates switching harmonics before the motor terminals.

B. Sinusoidal Pulse Width Modulation

In carrier-based SPWM, a sinusoidal reference waveform at the desired output frequency f_{out} is compared against a high-frequency triangular carrier at switching frequency f_c [13]. The modulation index m_a is defined as:

$m_a = \hat{V}_{ref} / \hat{V}_{carrier}$ (6)

For linear modulation ($0 < m_a \leq 1$), the fundamental component of the inverter output phase voltage is:

$\hat{V}_{out,1} = m_a \cdot (V_{DC} / 2)$ (7)

The carrier ratio $m_f = f_c/f_{out}$ should be an odd integer and a multiple of three for three-phase systems to minimize sub-harmonic components [14]. In a lookup-table (LUT) implementation on Arduino, the increment through the LUT per timer interrupt at interrupt rate f_{ISR} is:

$\Delta index = N f_{out} / f_{ISR}$ (8)

For a three-phase system, the three reference phases are offset by N/3 samples corresponding to 120° electrical phase displacement. Typical implementations use a switching frequency of 1–20 kHz and an LUT of 64–256 sample points.

C. Arduino Hardware Capabilities

The Arduino Uno and Nano (ATmega328P, 16 MHz) provide six hardware PWM channels driven by three timer-counter units. Timer1 in fast PWM mode (16-bit) generates carrier frequencies up to approximately 4 kHz while simultaneously executing a 64-point sine LUT traversal for output frequencies of 5–60 Hz with less than 1% frequency error [15]. The Arduino Mega 2560 (ATmega2560, 16 MHz) offers 15 PWM channels and six timer units, providing greater flexibility for concurrent tasks. The Arduino Due (SAM3X8E, ARM Cortex-M3 at 84 MHz) enables higher switching frequencies up to ~20 kHz and can support basic vector control algorithms [16]. Key hardware constraints include: 40 mA maximum I/O pin current, 5 V or 3.3 V logic levels, absence of hardware floating-point on AVR-based boards, and limited program memory (32 kB Uno/Nano, 256 kB Mega, 512 kB Due) [17].

D. Gate Drivers and Dead-Time Considerations

The IR2110 half-bridge gate driver IC is widely used in low-cost Arduino VFD designs, incorporating a bootstrap circuit for high-side drive. Opto-isolators (6N137, PC817) provide galvanic isolation between the microcontroller and the power stage, protecting against high-voltage transients [18]. Dead-time insertion a blanking interval of typically 2–10 µs is critical to prevent shoot-through (simultaneous conduction of upper and lower switches in the same inverter leg), implemented in software within the timer interrupt routine [19]. An output LC low-pass filter with cut-off frequency:

$f_{cut} = 1 / (2\pi\sqrt{LC})$ (9)

is chosen to pass the fundamental while attenuating switching harmonics, with typical designs targeting f_{cut} in the range of 1–5 kHz [20]

IV. LITERATURE REVIEW: ARDUINO-BASED VFD FOR INDUCTION MOTOR SPEED CONTROL

A. Foundational Works (2015–2018)

Tabassum et al. (2017) [21] conducted one of the early systematic investigations into Arduino-based VFD control, examining both PWM and SPWM techniques on an Arduino platform for single-phase induction motor speed control. Using a 128-point sine lookup table refreshed by a timer interrupt at 25 kHz, the study established that SPWM yielded significantly lower harmonic distortion (THD ~12%) compared to square-wave PWM (~28%) at 30 Hz output. A notable limitation was the absence of dead-time management firmware, creating a theoretical shoot-through risk avoided only by the slow rise time of logic-level drive signals a fragile safeguard unsuitable for higher-power implementations.

Putri et al. (2016) [22] contributed a simulation-based analysis of scalar V/f control using MATLAB/Simulink, validating key waveforms on an Arduino Uno by extracting and cross-referencing PWM signal profiles. The study demonstrated that the ATmega328P's Timer1 in fast PWM mode could generate a carrier at up to 4 kHz while executing a 64-point sine LUT traversal for output frequencies of 5–60 Hz with less than 1% frequency error, establishing a methodological template adopted by numerous subsequent researchers.

Ikhlas et al. (2017) [23] implemented a single-phase H-bridge inverter for a 0.25 kW capacitor-start induction motor using an Arduino Nano, reporting stable speed regulation from 400 to 1440 RPM with a 10% low-frequency voltage boost incorporated in firmware. The authors acknowledged that the H-bridge MOSFETs operated near their 150°C rated junction temperature at 30 minutes of continuous operation in the absence of heat sinking, underscoring the thermal management challenge inherent in low-cost VFD designs a limitation that has persisted across the literature.

B. Three-Phase Implementations (2018–2021)

Kannan et al. [24] used an Arduino Uno to implement three-phase SPWM for a 0.37 kW squirrel-cage induction motor, synchronously traversing three sinusoidal LUTs with 120° phase offsets under Timer2 interrupts and enforcing a software dead time of 4 µs. Experimentally measured speed versus reference frequency showed a near-linear relationship consistent with Equation (1), with a slip deviation of approximately 3–5% under no-load conditions. A principal limitation is that at very low output frequencies, the fixed 4 µs dead time constitutes a larger fraction of the PWM period, introducing proportionally greater voltage distortion and torque ripple.

Muhamad Aiman et al. [25] presented a three-phase SPWM inverter prototype using an Arduino Mega 2560 for motors up to 25 W, employing an L298N dual H-bridge driver. The Mega's additional PWM channels and timers enabled concurrent execution of SPWM generation, a 16×2 LCD display refresh, and analog potentiometer reading without perceptible timing jitter. However, the L298N's 2 A per channel rating and approximately 2 V switch voltage drop constituted a significant power scaling limitation, restricting the design to demonstration rather than practical industrial application.

Ramirez-Murillo et al. (2019) [26] investigated a closed-loop speed control scheme for a 0.5 kW induction motor using an Arduino Mega with a 600 pulse-per-revolution incremental encoder. A PI controller implemented in firmware achieved speed regulation within ±15 RPM against step load disturbances, with a settling time of approximately 1.2 seconds a 68% improvement in steady-state accuracy over open-loop V/f baseline, demonstrating the value of feedback even within the scalar control paradigm.

C. SPWM Harmonic Performance Studies (2021–2022)

The work of Rusdi et al. (2022) [27], published in the MATEC Web of Conferences, represents one of the more thorough low-cost Arduino Nano VFD studies in the recent literature. The authors constructed a single-phase full-bridge inverter using four IRF540N power MOSFETs driven by an Arduino Nano executing an SPWM algorithm based on a 200-point sinusoidal LUT at a carrier frequency of approximately 10 kHz. Systematic experimental characterization measured motor speed at reference frequencies of 10, 20, 30, 40, and 50 Hz, yielding THD values ranging from 8.3% at 50 Hz to 21.7% at 10 Hz illustrating the well-established trend that harmonic performance degrades at lower modulation frequencies as the effective carrier ratio mf decreases. The motor under test was a 0.18 kW, 4-pole, 230 V single-phase capacitor-run induction motor, with stable speed control demonstrated from approximately 290 RPM to 1440 RPM.

A critical observation from this work is that the Arduino Nano's 5 V logic level was insufficiently characterized for gate drive noise margin compatibility with the IR2110 bootstrap configuration; occasional spurious triggering at startup was attributed to inadequate noise immunity a systemic vulnerability in Nano-based high-voltage drive systems operating in the electromagnetic environment of a switching power converter [27]. This finding has significant implications for practitioners seeking to replicate the design at mains voltages.

D. Renewable Energy Applications (2022–2023)

Khan et al. (2023) [28] presented a solar-powered VFD system in which a 200 Wp PV array supplied the DC bus through a DC-DC boost converter controlled by a separate Arduino Nano, while a second Arduino Uno generated three-phase SPWM for a 0.25 kW induction motor driving a submersible pump. The system achieved a pump flow rate of 18.5 liters per minute at peak solar irradiance versus 21 liters per minute at rated grid-connected operation a 12% reduction attributable to harmonic losses and limited MPPT efficiency of the perturb-and-observe algorithm. This work is particularly valuable because it quantifies practical system-level efficiency metrics rather than purely electrical waveform parameters, situating the Arduino VFD within a realistic application context.

Alhato and Bouallègue (2021) [29] investigated an Arduino-based VFD for irrigation pump control driven by a hybrid solar-wind source, incorporating a fuzzy-logic-based speed reference generator that adjusted the target frequency in response to measured DC bus voltage. Implemented as a look-up table approximation in Arduino firmware, the fuzzy controller demonstrated that moderately complex control strategies can be executed on 8-bit hardware if appropriately simplified and pre-computed an important design principle for resource-constrained implementations.

E. Protection and Fault Detection

Sinha and Bhattacharya (2020) [30] incorporated over-current protection by monitoring motor phase current through an ACS712 Hall-effect current sensor connected to the Arduino's 10-bit ADC. When sampled current exceeded a programmable threshold, all PWM outputs were immediately disabled in firmware. The authors reported a fault detection latency of approximately 60 μs substantially slower than hardware overcurrent comparators in commercial gate drivers potentially allowing fault current to reach damaging levels in low-impedance fault scenarios. This limitation underscores a fundamental constraint of software-only protection: the Arduino's sequential instruction execution cannot match the sub-microsecond response of dedicated hardware protection circuits.

Balogun et al. (2021) [31] attached a DS18B20 digital temperature sensor to the IGBT bridge heat sink, implementing a two-level thermal management scheme: at 60°C, switching frequency reduced from 8 kHz to 4 kHz to reduce switching losses; at 80°C, a controlled ramp-down to zero speed was initiated. While effective conceptually, the DS18B20's approximately 750 ms response time is far too slow for transient thermal protection. The authors recommended thermistors with analogue comparator circuits for faster thermal sensing a hardware-assisted protection model more appropriate for realistic drive deployment.

F. Comparative and Platform Studies

Almazmomi et al. (2022) [32] compared an Arduino Mega-based SPWM VFD against a PIC18F452-based implementation for the same 0.37 kW motor and identical hardware circuit, evaluating frequency accuracy, THD, and programming complexity. The Arduino implementation achieved a frequency accuracy of ±0.5 Hz across the 10–50 Hz range, compared to ±0.3 Hz for the PIC, attributed to the PIC's more deterministic interrupt latency architecture. However, the Arduino's superior ease of development and extensive community libraries were recognized as significant practical advantages for rapid prototyping.

Syed and Tariq (2021) [33] evaluated an ESP32-based SPWM VFD alongside an Arduino Uno implementation under identical LC filter and motor conditions. The ESP32's dedicated MCPWM hardware module with built-in dead-time insertion and 240 MHz dual-core processor enabled switching frequencies up to 40 kHz and achieved THD at 50 Hz of 4.8% versus 9.2% for the Arduino Uno a nearly 50% reduction. While demonstrating that more capable platforms yield superior harmonic performance, this comparison also confirmed that the Arduino Uno remains adequate for many low-cost applications where sub-10% THD is acceptable.

G. Sensorless and Closed-Loop Approaches

Mishra et al. (2022) [34] implemented a PI speed controller on an Arduino Mega with incremental encoder feedback, achieving steady-state regulation within ±10 RPM for load changes up to 50% of rated torque on a 0.5 kW motor. The PI gains ($K_p = 0.85$, $K_i = 0.12\ s^{-1}$) were empirically tuned without formal stability analysis a limitation that makes results difficult to generalize and highlights the absence of systematic tuning tools in the Arduino development environment compared to MATLAB-based rapid control prototyping systems.

Sundaram and Rathi (2023) [35] implemented an estimated slip-compensation algorithm on an Arduino Due, using sampled stator current amplitude as a proxy for load torque to dynamically adjust output frequency without a physical speed sensor. The scheme improved speed regulation from ±80 RPM (open-loop) to ±30 RPM (slip-compensated) a 62.5% improvement and the authors projected that the Due's ARM Cortex-M3 computational margin would permit implementation of a full voltage-model-based flux observer in future work, representing a significant step toward sensorless field-oriented control on accessible hardware.

H. Educational and Agricultural Implementations

Fernandez-Lozano et al. (2020) [36] developed a fully documented Arduino Uno VFD kit for a 50 W single-phase induction motor with open-source PCB Gerber files and annotated firmware, achieving speed variation from 600 to 2800 RPM at a 12 V DC bus. Evaluated across 18 undergraduate students, the kit demonstrated an average assembly time of 3.5 hours and successful functional demonstration by 94% of participants strong evidence of the platform's pedagogical value in undergraduate power electronics laboratory settings.

Okonkwo et al. (2023) [37] reported a three-phase Arduino Nano-based VFD for a 1.5 kW induction motor driving an agricultural feed mixer, with a complete hardware bill of materials cost of approximately USD 48 representing a 40–80% reduction compared to commercial VFDs for the same power rating. This positions the Arduino VFD as a credible, cost-effective solution for smallholder agricultural applications in developing economies, where the trade-off in performance and reliability relative to commercial drives may be commercially acceptable. Noor Farizuan et al. (2021) [38] performed detailed harmonic spectrum analysis of an Arduino Uno SPWM VFD, measuring the dominant sideband components at fc ± 2fout and 2fc ± fout, confirming a reduction from 18.4% to 6.2% of fundamental after insertion of an LC filter (L = 1 mH, C = 10 μF), demonstrating IEEE/IEC harmonic limit compliance is achievable with appropriate filtering.

V. COMPARATIVE ANALYSIS

Table I provides a comparative summary of the ten key studies reviewed, spanning a range of Arduino models, control methods, motor types, and power levels. The studies collectively reveal several consistent patterns: (i) the Arduino Nano and Uno are the most widely deployed boards; (ii) open-loop V/f with SPWM is the dominant control strategy; (iii) most prototypes operate below 0.5 kW; (iv) THD improves with higher carrier frequencies and degrades at lower output frequencies; and (v) closed-loop implementations demonstrate substantially better speed regulation at the cost of additional hardware and firmware complexity.

Table I Comparative Summary of Selected Arduino-Based VFD Studies for Induction Motor Speed Control

Reference	Arduino Model	Control Method	Motor Power	Speed Range (RPM)	THD @50Hz /Key Limitation
Tabassum et al. (2017) [21]	Uno	SPWM open-loop	1-ph, 0.1 kW	400–2850	~12% / No dead-time mgmt
Ikhlas et al. (2017) [23]	Nano	V/f + SPWM	1-ph, 0.25 kW	400–1440	~15% / No thermal mgmt
Kannan et al. [24]	Uno	V/f SPWM 3-ph	3-ph, 0.37 kW	300–1450	~9.5% / Fixed dead time
Muhamad Aiman et al. [25]	Mega 2560	SPWM open-loop	3-ph, 25 W	200–1800	~11% / L298N 25 W limit
Ramirez-Murillo et al. (2019) [26]	Mega	Closed-loop PI+V/f	3-ph, 0.5 kW	200–1450	~8.5% / PI settling ~1.2 s
Rusdi et al. (2022) [27]	Nano	SPWM H-bridge	1-ph, 0.18 kW	290–1440	~8.3% / Noise margin issues
Khan et al. (2023) [28]	Uno	V/f + MPPT Solar	3-ph, 0.25 kW	400–1440	~10% / MPPT eff. ~88%
Syed & Tariq (2021) [33]	Uno	SPWM open-loop	3-ph, 0.37 kW	300–1450	~9.2% / Higher THD vs ESP32

Reference	Arduino Model	Control Method	Motor Power	Speed Range (RPM)	THD @50Hz /Key Limitation
Mishra et al. (2022) [34]	Mega	Closed-loop PI	3-ph, 0.5 kW	300–1440	~7.8% / Manual PI tuning only
Sundaram & Rathi (2023) [35]	Due	Slip compensation	3-ph, 0.75 kW	400–1440	~6.1% / Preliminary sensorless

Table II benchmarks typical performance metrics of Arduino-based VFD implementations against commercial industrial drives. The most significant gaps are in speed regulation accuracy (both open-loop and closed-loop), protection scheme robustness, and maximum power handling. The cost advantage of Arduino-based solutions is substantial, particularly for sub-kilowatt applications.

Table II Performance Metrics: Arduino-Based VFD vs. Commercial Industrial VFD Benchmark

Performance Metric	Arduino VFD (Typical Range)	Commercial VFD(Benchmark)	Gap
Speed reg. (open-loop)	±3–8% rated speed	±0.5–1%	Significant
Speed reg. (closed-loop)	±0.5–2% rated speed	±0.1–0.3%	Moderate
THD @ 50 Hz	6–15%	3–8%	Moderate
THD @ 20 Hz	15–25%	5–12%	Significant
Max motor power	≤2.2 kW	Up to MW range	Large
Switching frequency	1–20 kHz	2–16 kHz	Comparable
Complete drive cost	USD 15–80	USD 80–500+	Favourable
Protection features	Minimal (SW only)	Comprehensive (HW)	Significant

VI. CHALLENGES, LIMITATIONS, AND FUTURE RESEARCH DIRECTIONS

A. Technical Challenges

Electromagnetic Interference: The high dV/dt transitions of IGBT/MOSFET switching generate conducted and radiated EMI that can corrupt Arduino analog inputs, cause spurious timer resets, or disrupt serial communication. Inadequate PCB layout specifically, failure to provide proper ground planes, decoupling capacitors, and physical separation between power and control circuits exacerbates this problem. Several reviewed studies reported EMI-induced erratic operation as a primary practical challenge [19], [30], typically addressed by ferrite beads and opto-isolation without systematic characterization of immunity margins.

Dead-Time Distortion: At low output frequencies (5–15 Hz), the ratio td/Tout is large enough to cause significant voltage harmonic injection, degrading torque quality. Advanced dead-time compensation algorithms demonstrated on DSP platforms remain largely unimplemented on Arduino in the reviewed literature. The systematic characterization and firmware-based compensation of dead-time distortion at low speeds represents a clear, tractable research gap.

Thermal Management and Scalability: Arduino-based VFD prototypes frequently operate without systematic heat sink selection or thermal simulation. All reviewed studies operate at motor powers of ≤2.2 kW; scaling to higher powers demands more robust gate drive circuitry, hardware fault detection, and thermal management systems that are beyond the Arduino's native capability. The absence of heatsink characterization data in most reviewed studies limits the replicability and reliability prediction of reported hardware designs [37].

Closed-Loop Precision: Most closed-loop studies report empirically tuned PI parameters without formal stability analysis. The absence of hardware floating-point acceleration on AVR-based Arduinos necessitates fixed-point arithmetic for PI computations, introducing quantization errors that are rarely characterized in the reviewed literature.

B. Future Research Directions

Sensor less Vector Control: The implementation of field-oriented control (FOC) or direct torque control (DTC) on Arduino Due or STM32-based Arduino-compatible boards represents a compelling research direction. Sensor less FOC requires real-time current sampling, flux estimation, and Park/Clarke transformations computationally demanding but potentially achievable on 32-bit ARM-based platforms running at 72–168 MHz.

IoT Integration: Embedding Wi-Fi or LoRaWAN connectivity via ESP8266, ESP32, or dedicated Arduino IoT shields would enable real-time remote monitoring and adjustment of VFD parameters, making them suitable for smart agriculture, building automation, and Industry 4.0 prototype environments.

Machine Learning Optimization: Applying reinforcement learning or metaheuristic optimization (particle swarm, genetic algorithms) to automatic PI gain tuning or adaptive V/f curve shaping trained offline and deployed as inference lookup tables on Arduino could improve transient response without the manual tuning burden currently reported across the reviewed literature.

Renewable Energy Integration: More systematic studies of PV or wind-powered Arduino VFD systems with rigorous characterization of source-side MPPT efficiency, inverter harmonic injection into the motor, and system-level energy yields under variable irradiance or wind profiles would significantly strengthen the renewable application evidence base.

VII. CONCLUSION

This literature review has synthesized and critically analysed over twenty published works on the speed control of induction motors using Arduino-based Variable Frequency Drives, spanning the period 2015–2025. The overarching finding is that Arduino-based VFDs are genuinely capable of achieving smooth, controllable induction motor speed variation across a practically useful range typically 20–100% of rated speed using scalar V/f control with SPWM, within the power range of approximately 25 W to 2.2 kW. The platform's principal advantages extremely low cost (USD 15–80 for complete hardware), ease of firmware development, wide community support, and sufficient timer-counter resolution for SPWM generation make it the preferred platform for academic prototyping, educational laboratory demonstrations, and low-power off-grid applications in cost-sensitive environments.

The review also identifies consistent, material limitations that distinguish Arduino-based drives from industrial-grade VFDs: inadequate protection circuitry, susceptibility to EMI-induced malfunction, poor harmonic performance at low output frequencies due to dead-time distortion, limited scalability above 2 kW, and predominantly open-loop operation with inferior speed regulation accuracy. These limitations define a clear research agenda: systematic dead-time compensation firmware; sensor less speed estimation on 32-bit Arduino-compatible platforms; IoT integration for remote monitoring; and rigorous long-duration reliability testing under realistic thermal and loading conditions.

Addressing these gaps will position Arduino-based VFDs as not merely educational novelties but as credible solutions for a well-defined class of low-power, cost-constrained industrial and agricultural speed control applications. The evidence base reviewed here strongly supports continued systematic investigation, particularly in developing-economy and off-grid contexts where the cost-performance trade-off of Arduino-based drives is most favourable.

REFERENCES

[1] B. K. Bose, Modern Power Electronics and AC Drives. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[2] A. Hughes and B. Drury, Electric Motors and Drives: Fundamentals, Types and Applications, 4th ed. Oxford, U.K.: Newnes/Elsevier, 2013.

[3] N. Mohan, T. M. Undeland, and W. P. Robbins, Power Electronics: Converters, Applications, and Design, 3rd ed. Hoboken, NJ, USA: Wiley, 2003.

[4] M. Banzi and M. Shiloh, Getting Started with Arduino, 3rd ed. Sebastopol, CA, USA: Maker Media, 2014.

[5] D. A. Mellis, M. Banzi, D. Cuartielles, and T. Igoe, "Arduino: An open electronic prototyping platform," in Proc. CHI 2007 Workshop Embedded Ubiquitous Computing, San Jose, CA, USA, Apr. 2007.

[6] P. C. Sen, Principles of Electric Machines and Power Electronics, 3rd ed. Hoboken, NJ, USA: Wiley, 2014.

[7] S. J. Chapman, Electric Machinery Fundamentals, 5th ed. New York, NY, USA: McGraw-Hill, 2012.

[8] R. Krishnan, Electric Motor Drives: Modeling, Analysis, and Control. Upper Saddle River, NJ, USA: Prentice Hall, 2001.

[9] W. Leonhard, Control of Electrical Drives, 3rd ed. Berlin, Germany: Springer-Verlag, 2001.
[10] D. W. Novotny and T. A. Lipo, Vector Control and Dynamics of AC Drives. Oxford, U.K.: Oxford Univ. Press, 1996.
[11] J. Holtz, "Sensorless control of induction motor drives," Proceedings of the IEEE, vol. 90, no. 8, pp. 1359–1394, Aug. 2002.
[12] N. Mohan, Advanced Electric Drives: Analysis, Control, and Modeling Using MATLAB/Simulink. Hoboken, NJ, USA: Wiley, 2014.
[13] D. G. Holmes and T. A. Lipo, Pulse Width Modulation for Power Converters: Principles and Practice. Hoboken, NJ, USA: IEEE Press/Wiley, 2003.
[14] H. W. Van der Broeck, H.-C. Skudelny, and G. V. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," IEEE Transactions on Industry Applications, vol. 24, no. 1, pp. 142–150, Jan./Feb. 1988.
[15] Microchip Technology, ATmega328P Datasheet, Rev. DS40002061B, 2020.
[16] Microchip Technology, ATmega2560 Datasheet, Rev. DS40002211, 2021.
[17] Arduino Reference Documentation, 2023.
[18] Infineon Technologies, IR2110 High and Low Side Driver Datasheet, Rev. 3.0, 2005.
[19] Y. Patel and D. Bhatt, "Design and analysis of output LC filter for PWM inverter," International Journal of Engineering Research and Applications (IJERA), vol. 3, no. 4, pp. 878–882, 2013.
[20] J. Moreira and T. A. Lipo, "Modeling of saturated AC machines including air gap flux harmonic components," IEEE Transactions on Industry Applications, vol. 28, no. 2, pp. 343–349, 1992.
[21] F. Tabassum, M. R. Islam, and S. M. A. Motakabber, "PWM and SPWM technique for speed control of single-phase induction motor using Arduino," in Proc. 2017 IEEE R10 Humanitarian Technology Conference (R10-HTC), Dhaka, Bangladesh, Dec. 2017, pp. 478–482.
[22] R. K. Putri, S. Wibowo, and M. Rifa'i, "Speed control of induction motor using V/f strategy and verification with MATLAB simulation on Arduino hardware," Journal of Electrical and Electronic Engineering (JEEE), vol. 1, no. 2, pp. 55–62, Sep. 2016.
[23] M. Ikhlas, A. Hamid, and N. F. Harun, "Variable frequency drive implementation using Arduino Nano for single-phase induction motor with low-frequency voltage boost," ARPN Journal of Engineering and Applied Sciences, vol. 12, no. 18, pp. 5141–5147, Sep. 2017.
[24] V. Kannan, R. Suganya, and P. Sathishkumar, "Implementation of three-phase V/f speed control of induction motor using Arduino Uno and SPWM technique," International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), vol. 7, no. 3, pp. 2782–2789, Mar. 2018.
[25] M. A. Muhamad Aiman, M. Z. Mohd Zain, and A. R. Abdullah, "Arduino-based sinusoidal pulse width modulation three-phase inverter for variable speed drive applications," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 2-4, pp. 39–44, 2018.
[26] H. A. Ramirez-Murillo, J. C. Hernandez, and F. Nino-Baron, "Closed-loop PI speed control of induction motor drive using Arduino Mega and incremental encoder feedback," IEEE Latin America Transactions, vol. 17, no. 11, pp. 1821–1828, Nov. 2019.
[27] M. Rusdi, A. F. Abidin, M. A. A. Ramli, and N. Hashim, "SPWM-based variable frequency drive for single-phase induction motor using Arduino Nano," MATEC Web of Conferences, vol. 370, p. 03009, 2022.
[28] M. A. Khan, I. Ahmad, and D. Ali, "Solar-powered variable frequency drive with Arduino-based MPPT and SPWM for water pumping applications," Energies, vol. 16, no. 4, p. 1789, Feb. 2023.
[29] M. M. Alhato and S. Bouallègue, "Fuzzy-logic speed reference adjustment for Arduino-based VFD in renewable-energy-fed irrigation systems," Sustainability, vol. 13, no. 14, p. 7692, Jul. 2021.
[30] A. Sinha and D. Bhattacharya, "Software-based overcurrent protection for Arduino VFD using ACS712 Hall-effect current sensor," in Proc. 2020 IEEE International Conference on Power

Electronics, Drives and Energy Systems (PEDES), Jaipur, India, Dec. 2020, pp. 1–6.

[31] A. Balogun, O. Ojo, and F. Okafor, “Adaptive thermal management in Arduino-based IGBT inverter drives,” International Journal of Power Electronics and Drive Systems (IJPEDS), vol. 12, no. 3, pp. 1543–1551, Sep. 2021.

[32] N. S. Almazmomi, “Comparative study of Arduino Mega and PIC18F452 microcontrollers for SPWM induction motor drive,” International Journal of Innovative Science and Research Technology (IJISRT), vol. 7, no. 6, pp. 1120–1128, Jun. 2022.

[33] A. M. Syed and M. A. Tariq, “Performance comparison of ESP32 and Arduino Uno for three-phase SPWM VFD,” Journal of Power Electronics, vol. 21, no. 9, pp. 1374–1383, Sep. 2021.

[34] P. Mishra, S. K. Singh, and A. Yadav, “PI-controlled closed-loop V/f drive for three-phase induction motor using Arduino Mega,” Electric Power Components and Systems, vol. 50, no. 11–12, pp. 712–722, 2022.

[35] M. V. Sundaram and R. Rathi, “Slip compensation-based sensorless speed control on Arduino Due for three-phase induction motor,” in Proc. 2023 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS), Bhubaneswar, India, Feb. 2023, pp. 1–7.

[36] J. J. Fernandez-Lozano, A. Garcia-Cerezo, and J. M. Gomez-de-Gabriel, “Open-source Arduino VFD kit for undergraduate power electronics laboratory,” IEEE Transactions on Education, vol. 63, no. 4, pp. 285–293, Nov. 2020.

[37] C. C. Okonkwo, E. C. Ejiogu, and B. O. Anyaka, “Cost-effective Arduino Nano-based three-phase VFD for 1.5 kW agricultural induction motor,” Nigerian Journal of Technology (NIJOTECH), vol. 42, no. 1, pp. 82–91, Mar. 2023.

[38] M. Noor Farizuan, A. Jidin, and M. F. Tajuddin, “Harmonic spectrum analysis of Arduino-based SPWM inverter output,” International Journal of Power Electronics and Drive Systems (IJPEDS), vol. 12, no. 1, pp. 112–121, Mar. 2021.