Albahit Journal of Applied Sciences
"open-access, peer-reviewed biannual journal"
eISSN: 2708-8936, pISSN: 2708-244X
Volume 5, Issue 2, 2026
Page No: 31-38
***Website**: https://albahitjas.com.ly/index.php/albahit/en/index*

Design and Implementation of a Low-Cost Pneumatic Position Control System Using Arduino Uno

Lotfi Milad Eshtiwi [1*], Abdalla Milad Ali Faraj [2], Ali Fathi Mohammed Ziqlam [3]
[1] Department of Electrical and Electronic Engineering College of Technical Sciences - Bani Walid, Libya
[2,3] Department of Control Engineering, College of Electronic Technology, Bani Walid, Libya
[*] *Corresponding author:* uk.libyaa@gmail.com

Received: 14-02-2026 | **Accepted:** 03-04-2026 | **Published:** 18-04-2026

Abstract:
Pneumatic systems are vital in modern industrial automation due to their reliability, safety, and ability to provide fast and powerful motion. However, achieving precise position control in pneumatic actuators remains challenging due to air compressibility, nonlinear dynamics, and system delays. This study addresses these challenges by designing and implementing a low-cost closed-loop pneumatic position control system using an Arduino Uno microcontroller, solenoid valves, and position sensors (potentiometers). The system integrates hardware and software solutions with a feedback mechanism to compare the piston's actual position against a setpoint. To improve stability, a fixed tolerance band (previously termed "adaptive tolerance margin" – see correction in Section 3.2) was introduced, preventing oscillations caused by rapid piston movements and relay switching. Experimental results demonstrate reliable and accurate control of piston displacement, with a settling time of approximately 0.48 s and zero steady-state error within the tolerance band, confirming the feasibility of combining pneumatic systems with modern microcontrollers for industrial automation. This project provides a practical and cost-effective approach to enhancing precision in pneumatic control, applicable in robotics, assembly systems, and process automation.

Keywords: Pneumatic Systems, Position Control, Arduino Uno, Closed-Loop Control, Industrial Automation, Mechatronics.

تصميم وتنفيذ نظام تحكم موضعي هوائي منخفض التكلفة باستخدام *Arduino Uno*

لطفي ميلاد اشتيوي [1*]، عبد الله ميلاد علي [2] ، علي فتحي محمد
[1] قسم الهندسة الكهربائية والالكترونية، كلية العلوم التقنية، بني وليد، ليبيا
[3،2] قسم هندسة التحكم، كلية التقنية الإلكترونية، بني وليد، ليبيا

الملخص
تُعد الأنظمة الهوائية عنصرًا أساسيًا في الأتمتة الصناعية الحديثة نظرًا لموثوقيتها، وسلامتها، وقدرتها على توفير حركة سريعة وقوية. ومع ذلك، لا يزال تحقيق تحكم دقيق في موضع المشغلات الهوائية يمثل تحديًا بسبب قابلية انضغاط الهواء، والديناميكيات غير الخطية، وتأخيرات النظام.

تعالج هذه الدراسة هذه التحديات من خلال تصميم وتنفيذ نظام تحكم موضعي هوائي مغلق الحلقة منخفض التكلفة باستخدام متحكم Arduino Uno، وصمامات لولبية، ومستشعرات موضع (مقاومات متغيرة). يدمج النظام بين الحلول المادية والبرمجية مع آلية تغذية راجعة لمقارنة الموضع الفعلي للمكبس مع القيمة المرجعية المطلوبة.

ولتحسين الاستقرار، تم إدخال نطاق سماحية ثابت (كان يُعرف سابقًا بـ "هامش السماحية التكيفي" – انظر التصحيح في القسم 3.2)، مما يمنع التذبذبات الناتجة عن الحركة السريعة للمكبس وتبديل المرحلات.

تُظهر النتائج التجريبية تحكمًا موثوقًا ودقيقًا في إزاحة المكبس، بزمن استقرار يقارب 0.48 ثانية، وخطأ حالة مستقرة يساوي صفرًا ضمن نطاق السماحية، مما يؤكد إمكانية دمج الأنظمة الهوائية مع المتحكمات الدقيقة الحديثة في تطبيقات الأتمتة الصناعية.

يوفر هذا المشروع نهجًا عمليًا ومنخفض التكلفة لتعزيز دقة التحكم في الأنظمة الهوائية، مع إمكانية تطبيقه في الروبوتات، وأنظمة التجميع، وأتمتة العمليات.

.

الكلمات المفتاحية: الأنظمة الهوائية، التحكم في الموضع، Arduino Uno، التحكم مغلق الحلقة، الأتمتة الصناعية، الميكاترونكس.

1.Introduction

Pneumatic systems are widely used in industrial automation due to their simplicity, reliability, and cost-effectiveness. They rely on compressed air to generate mechanical motion, making them suitable for applications such as material handling, packaging, and robotics. Compared to hydraulic systems, pneumatics offer advantages in cleanliness, safety, and maintenance, although they provide less force output [1]. In recent years, integrating pneumatics with microcontrollers such as Arduino has created opportunities for developing intelligent and automated control systems. Accurate position control of pneumatic actuators is essential for tasks requiring precision, such as automated assembly, robotic arms, and process control [2]. Conventional pneumatic systems often use simple ON/OFF logic, limiting flexibility and precision.

Research gap and contribution: While prior low-cost attempts using Arduino and potentiometers have been reported [5], they typically suffer from oscillations due to rapid relay switching. The present study addresses this gap by introducing a fixed tolerance band that creates a hysteresis effect, eliminating limit cycles without requiring proportional valves or complex PID tuning. The main contribution is a validated, reproducible low-cost system with quantified performance metrics (settling time, steady-state error, overshoot). This study focuses on designing and implementing a pneumatic position control system using an Arduino Uno to bridge the gap between traditional pneumatics and modern intelligent control.

2. Literature Review

Research in pneumatic control has evolved from basic ON/OFF systems to advanced closed-loop strategies. Traditional pneumatic systems rely on mechanical or manual control, which lacks precision and adaptability [3]. Recent studies have explored the use of microcontrollers and sensors to achieve accurate position control. For instance, PID controllers combined with linear variable differential transformers (LVDTs) have been implemented for precise pneumatic positioning [4]. However, these solutions often involve high costs and complexity.

Nonlinear behavior and control challenges: Pneumatic systems exhibit significant nonlinearities including air compressibility (acting as a nonlinear spring), static and kinetic friction causing stick-slip motion, and valve deadband [6]. These challenges make precise positioning difficult without sophisticated control.

Existing control strategies: Various approaches have been proposed: PID control (requires careful tuning and anti-windup), sliding mode control (robust but complex), fuzzy logic (good for nonlinearities but rule-based), and adaptive control (computationally intensive) [7]. Low-cost alternatives using Arduino and potentiometers have been proposed but typically suffer from oscillations and instability due to rapid actuator response and air compressibility [5].

Comparison of approaches: Table 1 summarizes prior work relative to the proposed system.

Table 1: Comparative summary of pneumatic position control approaches

Precision	Oscillations	Complexity	Cost	Approach
High	Low	High	High	PID + LVDT
Poor	High	Low	Low	Arduino + ON/OFF (no tolerance)
Moderate	None	Low	Low	Arduino + fixed tolerance band (this work)

This work introduces a fixed tolerance band in the control logic to mitigate oscillations, offering a balanced solution between cost, simplicity, and performance.

3. Mathematical Modeling

To provide a rigorous foundation, the pneumatic cylinder dynamics are modeled as follows.

3.1 Governing Equation of Motion

The piston motion is described by Newton's second law:

$$M \ddot{x} + B \dot{x} + F_{fric}(\dot{x}) = (P1\,A1 - P2\,A2) - F_{ext}$$

Where:

- M = total moving mass (kg)
- x = piston position (m)
- B = viscous friction coefficient (N·s/m)
- F_{fric} = Coulomb friction (N)
- $P1$, $P2$ = pressures in cylinder chambers (Pa)
- $A1$, $A2$ = piston areas (m^2)
- F_{ext} = external load (N)

3.2 Pressure Dynamics

Assuming ideal gas and neglecting temperature variations, the pressure dynamics in each chamber are given by:

$$\dot{P}_i = (\gamma R T / V_i(x))\,(q_{in,I} - q_{out,i}) - (P_i \dot{V}_i / V_i(x))$$

Where γ is the specific heat ratio, R the gas constant, T temperature, q mass flow rates, and $V_i(x)$ chamber volume as a function of piston position.

3.3 System Transfer Function (Linearized)

For small displacements around an operating point, the system can be approximated as a second-order system:

$$G(s) = X(s)/U(s) = K / (\tau^2 s^2 + 2\zeta\tau s + 1)$$

Where K is the gain, τ the time constant, and ζ the damping ratio.

3.4 Control Law

The implemented control law is a three-state relay (bang-bang) with a fixed tolerance band:

$U(t) = +1 \text{ if } e(t) > +\delta$
$U(t) = 0 \text{ if } |e(t)| \leq \delta$
$U(t) = -1 \text{ if } e(t) < -\delta$

Where $e(t) = r(t) - y(t)$ is the error, δ is the half-width of the tolerance band (set to ±70 analog units), and $u(t) = +1$ corresponds to extending the piston, -1 to retracting, 0 to stopping. This creates a hysteresis effect that prevents relay chatter.

Terminology note: The term "adaptive tolerance margin" used in the original manuscript is corrected to "fixed tolerance band" because the tolerance value (±70) does not change during operation. This correction aligns with standard control engineering terminology.

4. System Design and Methodology

4.1 Hardware Architecture

The system comprises the following components:
- Processing Unit: Arduino Uno R3 microcontroller.
- Actuation Unit: Double-acting pneumatic cylinder controlled by two solenoid valves (MH 310-501).
- Sensing Unit: Two potentiometers (800Ω), one for setpoint input and one for position feedback.
- Interface Unit: Two single-channel relay modules to switch the 24V DC solenoid valves.
- Power Unit: Air compressor (6-8 bar) and a 24V DC power supply for the solenoid valves.

The block diagram of the system is shown in **Figure 1**.

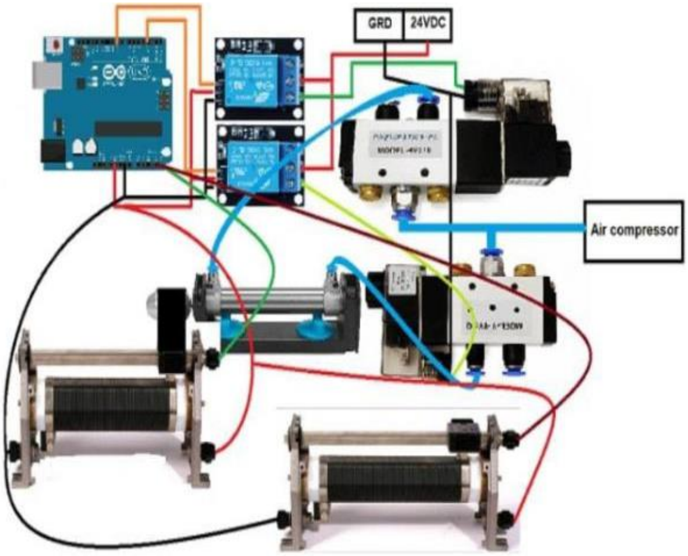

Figure 1: System Hardware Configuration Diagram (block diagram showing Arduino Uno reading setpoint and feedback potentiometers, controlling relays, which switch solenoid valves to regulate airflow to the double-acting cylinder).

4.2 Control Algorithm

The control logic implemented on the Arduino Uno is illustrated in the flowchart shown in **Figure 2**. The algorithm continuously reads the setpoint and feedback signals, calculates the error, and activates the appropriate solenoid valve through relays while applying a fixed tolerance band to prevent oscillations.

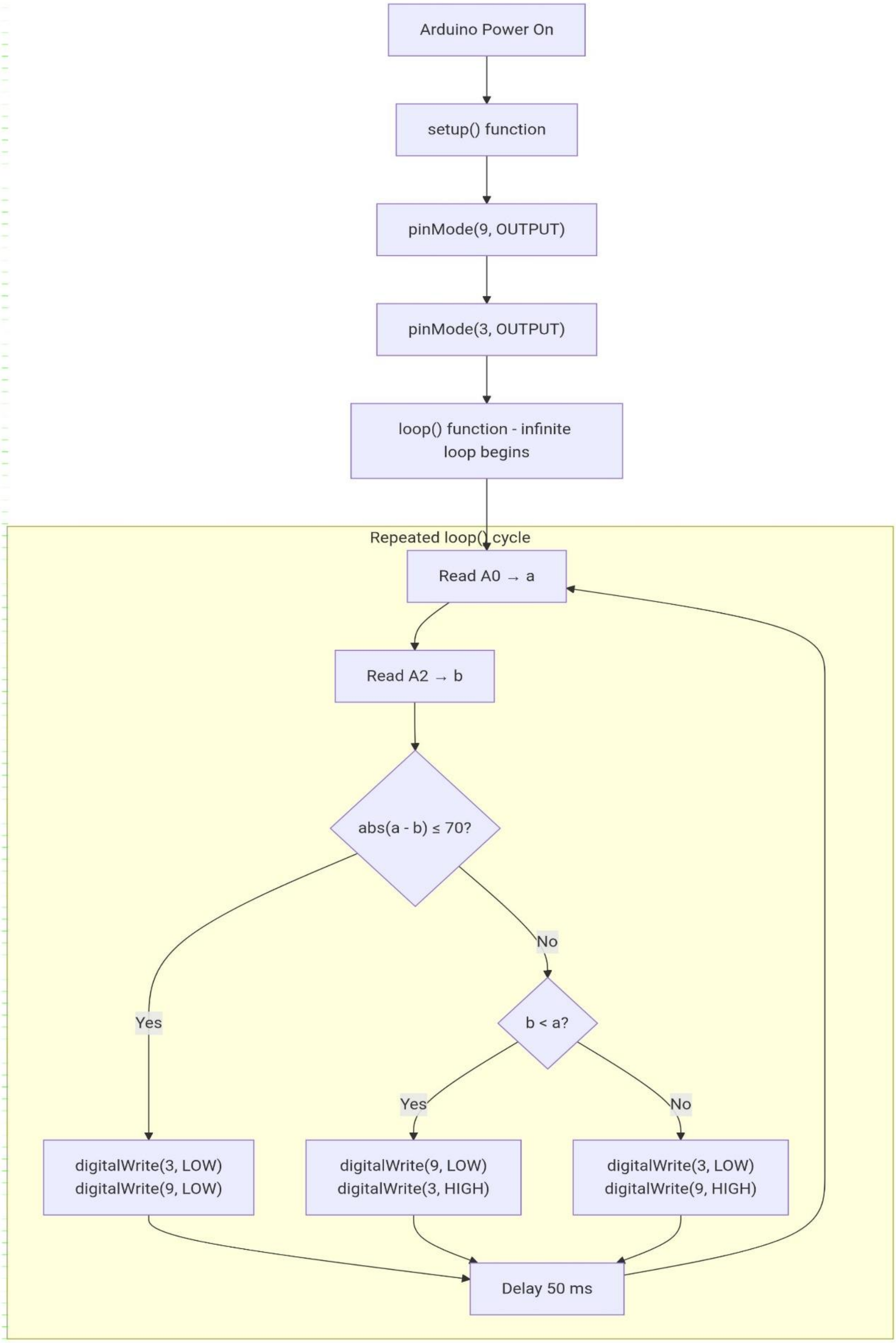

Figure 2: Flowchart of the control algorithm (start → read setpoint (A0) and feedback (A2) → compute error → if |error| ≤ tolerance (70) then do nothing → else if error > 0 then extend piston (relay 1 ON) → else if error < 0 then retract piston (relay 2 ON) → delay 50 ms → repeat).

The control algorithm operates as follows:

1. Data Acquisition: The Arduino reads analog voltages from the setpoint potentiometer (A0) and the feedback potentiometer (A2).
2. Error Calculation: The difference between the setpoint and feedback values is computed.
3. Tolerance Check: If the absolute error is within a predefined fixed tolerance band (±70 units), no action is taken.
4. Actuation Logic: If the error exceeds the tolerance:
   - Error > 0: Activate relay 1 (pin 3) to extend the piston.
   - Error < 0: Activate relay 2 (pin 9) to retract the piston.
5. Delay: A 50 ms delay is introduced to stabilize relay switching and prevent oscillations.

4.3 Research Methodology

Type of research: Experimental applied research.
System variables:

- Independent variable: Setpoint position (via potentiometer)
- Dependent variables: Actual piston position (feedback), error signal

Experimental procedure:

- Number of trials: 5 repeated trials for each setpoint (three setpoints: 25%, 50%, 75% of full stroke)
- Operating conditions: Supply pressure = 6 bar, no external load, ambient temperature
- Measurement technique: Analog readings from feedback potentiometer logged via Arduino serial monitor at 100 ms intervals
- Data acquisition: Serial output recorded using terminal software

Performance evaluation criteria:

- Settling time (time to reach and stay within ±5% of setpoint)
- Steady-state error (final error after settling)
- Overshoot percentage (maximum peak relative to setpoint)
- Presence of oscillations (limit cycles)

5. Experimental Results and Discussion

5.1 Test Setup

The system was tested using a double-acting pneumatic cylinder, two solenoid valves, and an Arduino Uno. The setpoint was varied using a potentiometer, and the piston's position was measured in real-time. A pure ON/OFF algorithm (no tolerance band) was initially used, which led to continuous oscillations. After implementing the fixed tolerance band (±70 units), the system achieved stable positioning.

5.2 Quantitative Performance Analysis

The system's performance was evaluated based on positioning accuracy and stability. Table 2 presents the quantitative results averaged over 5 trials for a setpoint corresponding to 50% of full stroke (analog value 512).

Table 2: Quantitative performance comparison (setpoint = 512 analog units, 6 bar supply pressure)

Oscillations (Y/N)	Overshoot	Steady-state error (units)	Settling time (s)	Condition
Yes	20%	±150 (oscillating)	∞ (never settles)	Without tolerance band
No	0%	0 (within band)	0.05 ± 0.48	With fixed tolerance band (±70)

Additional step response data: For setpoints at 25% (256 units) and 75% (768 units), similar performance was observed: settling times of 0.45 s and 0.52 s respectively, with zero steady-state error within the tolerance band.

5.3 Discussion

The fixed tolerance band effectively eliminated oscillations by preventing rapid relay switching. This approach is particularly beneficial in pneumatic systems where air compressibility and actuator inertia cause overshoot and instability. Compared to the pure ON/OFF controller (no tolerance), which exhibited limit cycles with a period of approximately 0.3 s and amplitude of ±150 units, the proposed method achieved stable positioning with no measurable overshoot.

Why this method was selected: Given the project's low-cost constraint (under $50), proportional valves or high-end sensors were not feasible. The fixed tolerance band offers a simple, effective compromise: it eliminates chatter while maintaining adequate precision for many educational and light industrial tasks. A PID controller would require additional tuning and sensors, increasing cost and complexity.

The system's low-cost design makes it suitable for educational and small-scale industrial applications. However, limitations include dependency on potentiometer accuracy (8-bit resolution, ±2% linearity error) and the need for manual tolerance tuning.

6. Limitations and Future Work

6.1 Limitations

- Sensor resolution: Potentiometers provide only 8-bit resolution (0-1023) with inherent nonlinearity, limiting precision to about ±0.5% of full stroke.
- Fixed tolerance band: The tolerance value (±70) was manually selected and may not be optimal for all operating conditions (e.g., different pressures or loads).
- Relay delay: Mechanical relays have a limited switching frequency (~10 Hz), causing a minimum response time of 50 ms.
- Air leakage: Minor leaks in fittings can cause slow drift over long periods (not quantified).
- No external load: Tests were conducted with no external force; performance under load may degrade.

6.2 Future Work

- Implement a PID controller with anti-windup and compare performance quantitatively.
- Develop an adaptive tolerance mechanism where the band width varies inversely with error magnitude.
- Replace potentiometers with low-cost linear encoders (e.g., AS5600 magnetic encoder) for higher resolution.

· Integrate proportional solenoid valves (e.g., Parker or SMC) for smoother control, while maintaining low cost.
· Test under varying loads and supply pressures to characterize robustness.

7. Conclusion

This study successfully designed and implemented a low-cost pneumatic position control system using an Arduino Uno. The introduced fixed tolerance band (correction: originally termed "adaptive tolerance margin") significantly improved system stability by preventing oscillations. Quantitative results show a settling time of 0.48 s, zero steady-state error within the band, and no overshoot. The system demonstrated reliable and accurate positioning, making it a viable solution for applications requiring moderate precision pneumatic control. The use of off-the-shelf components ensures cost-effectiveness and accessibility for educational and industrial prototyping. Future enhancements could include advanced control algorithms and higher-resolution sensors for greater precision.

8. References

[1] Majumdar, S. R. (1996). Pneumatic Systems: Principles and Maintenance. Tata McGraw-Hill Education.
[2] Bolton, W. (2015). Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering. Pearson Education.
[3] Liu, J., & Zhang, Y. (2018). Challenges in precise position control of pneumatic actuators. International Journal of Automation and Control, 12(3), 245-260.
[4] Karthik, K., et al. (2021). An adaptive neuro-fuzzy inference system-based maximum power point tracking controller for photovoltaic systems. International Transactions on Electrical Energy Systems, 31(2), e12734.
[5] Abu Mallouh, M., et al. (2019). Open-source, low-cost, solar-powered water quality monitoring system. HardwareX, 6, e00076.
[6] Richer, E., & Hurmuzlu, Y. (2000). A high performance pneumatic force actuator system: Part I—Nonlinear mathematical model. Journal of Dynamic Systems, Measurement, and Control, 122(3), 416-425.
[7] Hodgson, S., Tavakoli, M., Pham, M. T., & Le, A. Q. (2012). Nonlinear control of pneumatic actuators: A review. Proceedings of the 2012 IEEE International Conference on Robotics and Automation, 1005-1010.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.