International Scientific Conference 20 UNITECH 26

INTERNATIONAL SCIENTIFIC CONFERENCE
19-21 November 2026
GABROVO

ПРИЛОЖЕНИЕ НА МИКРОКОНТРОЛЕР В ЕЛЕКТРОПНЕВМАТИЧНА СЛЕДЯЩА СИСТЕМА С ШИМ УПРАВЛЕНИЕ

Георги Славчев Илиев

Катедра: Енергийна техника
Технически Университет Габрово, Габрово, България
Център за компетентност „Интелигентни мехатронни, екологични и енергоспестяващи системи и технологии“,
**кореспондиращ автор: e-mail: g.iliev@tugab.bg*

APPLICATION OF A MICROCONTROLLER IN AN ELECTRO-PNEUMATIC TRACKING SYSTEM WITH PWM CONTROL

Georgi Slavchev Iliev

dept. Power Engineering,
Technical University of Gabrovo, Gabrovo, Bulgaria
Center of Competence "Smart Mechatronic, Eco- and Energy-Saving Systems and Technologies",
**Corresponding author: e-mail: g.iliev@tugab.bg*

Abstract
This paper reports the development and the experimental validation of an inexpensive embedded control platform, built around an Arduino Uno microcontroller, that regulates the position and the speed of an electro-pneumatic drive within a closed loop. The actuator chosen for the installation is a rodless pneumatic cylinder: its carriage can be coupled directly to a linear position transducer, the required stroke is obtained within a short overall length, and the sealing wear and rod buckling that limit conventional rod-type cylinders are avoided. Positioning accuracy has always been the weak point of pneumatic drives, because the compressibility of air, the friction in the seals and the non-linear flow characteristic of the valves make an open-loop command unreliable; microcontrollers have changed the situation, being compact, inexpensive and fast enough to execute digital control algorithms in real time. In the system described here four two-port high-speed valves of the SMC SX10 series are driven by pulse-width modulation, with a carrier frequency adjustable between 10 Hz and 100 Hz and a duty cycle variable from 0 to 90 %, so that the air flow through each valve, and with it the speed of the carriage, is varied continuously instead of being switched in discrete steps. Position feedback is taken from a linear potentiometric sensor whose 0–5 V output is calibrated so that 5 V corresponds to the full 500 mm stroke. The measurements confirm that accurate positioning and independent speed control are attainable with this low-cost hardware and that the modulated supply consumes less electrical energy than conventional on/off control of the coils.

Keywords: Arduino Uno microcontroller, electro-pneumatic systems, pulse-width modulation, high-speed pneumatic valves.

INTRODUCTION

Each new generation of production equipment raises the requirements placed on its drives, and speed, positioning accuracy and energy efficiency are demanded at the same time. Pneumatic

This is an open access article licensed under Creative Commons Attribution 4.0 International
Print ISSN: 1313-230X, Online ISSN: 3033-1404

drives hold a firm position among the available solutions, being robust, cheap to manufacture and to maintain, and intrinsically safe in explosive or chemically aggressive surroundings. Their persistent weakness is positioning: an air cylinder cannot be stopped exactly where it is wanted as easily as an electric servo, because the working medium is compressible and the pressure in the chambers therefore cannot be commanded as a stiff quantity, while seal friction and the non-linear relation between valve opening and mass flow further disturb the motion. A considerable part of the research devoted to pneumatic drives is aimed at closing this gap. Microcontroller technology offers a practical route towards that aim. A modern controller is small, inexpensive and fast enough to sample a feedback signal and to recalculate a control law within a few milliseconds, which is precisely what the compensation of the effects listed above requires. Such a controller has no built-in output stage for a proportional valve, and pulse-width modulation (PWM) is therefore employed instead: by chopping the supply to a fast on/off valve, an effective average flow is produced that can be adjusted continuously, so that a proportional behaviour is obtained at the price of an inexpensive valve [1], [3], [4],[8] The present paper follows this approach. Its purpose is to describe and to evaluate an electro-pneumatic tracking system in which an Arduino Uno microcontroller performs closed-loop control of position and speed. To this end the following tasks were formulated: to examine the pneumatic drive as a controlled plant and to identify the disturbances that dominate its behaviour; to select an adequate system architecture and feedback transducer; to synthesise a control algorithm that fits within the limited resources of the microcontroller; and to assess the positioning accuracy, the smoothness of motion and the consumption of air and electrical energy actually achieved [2],[6],[9],[10].

MAIN TEXT

The experimental installation was built around an Arduino Uno board, which performs both the modulation of the valve commands and the acquisition of the position signal. The mechanical part of the test bench (Fig. 1) is a rodless pneumatic cylinder- Camozzi 52M2P32A0450 with a stroke of 500 mm, the chambers of which are fed through four direct-acting two-port high-speed valves of the SMC SX10 series. Each valve is actuated by a 24 V solenoid coil with a rated power of 80 W, and the coils are supplied through pulse-width modulation rather than by a continuous voltage [3]. The rodless construction was chosen deliberately: for a stroke of this length a conventional cylinder would be almost twice as long overall, since the piston rod has to be accommodated outside the barrel, whereas the rodless design keeps the installation compact and at the same time avoids the seal abrasion and the bending loads to which a long unsupported rod is exposed; the movable carriage is also convenient for the direct mounting of the linear position sensor. An original control program was written for the controller. It generates four independent PWM channels whose carrier frequency can be set anywhere between 10 Hz and 100 Hz, the duty cycle being continuously variable from 0 % to 90 %. The flow through any valve — and hence the speed of the carriage — is thus modulated smoothly instead of being switched in a purely discrete manner, which improves the dynamics of the drive and reduces the shock loading when the carriage reaches the end positions.[4],[5],[7].

The displacement of the carriage is read

by a linear potentiometric transducer-Magauto LSOB-500-K-0005 Its analogue output, which spans 0–5 V, is returned to the controller and compared with the reference value, so that the control loop is closed. The transducer is calibrated over the whole stroke, which means that 5 V corresponds to 500 mm and the sensitivity of the measuring channel is 100 mm/V. The valves are actuated in alternating pairs: one pair remains energised until the feedback voltage reaches 3.50 V, after which it is switched off and the opposite pair is energised until the voltage falls back to 0.5 V, whereupon the sequence is repeated. These two levels correspond to carriage positions of 350 mm and 30 mm and were selected in accordance with the operating range required. Besides the static positioning error, the investigation covered the influence of the modulation parameters on the transient response, on the attainable speed of the carriage and on the electrical power drawn by the coils.

A further function of the program is a routine for forced shut-down. Here the polarity of the coil supply is reversed and a duty cycle of 100 % is commanded for an interval defined by the user; the counter-field produced in this way accelerates the decay of the magnetic flux in the coil and the release of the piston, and the overshoot of the moving mass is thereby limited. The program also permits the operating parameters to be monitored and adjusted while the system is running. The results obtained on the test bench show that a low-cost microcontroller is able to provide both accurate positioning and independent speed control of the pneumatic actuator, while the electrical energy consumed by the drive is reduced in comparison with conventional on/off control.[6],[7],[9].

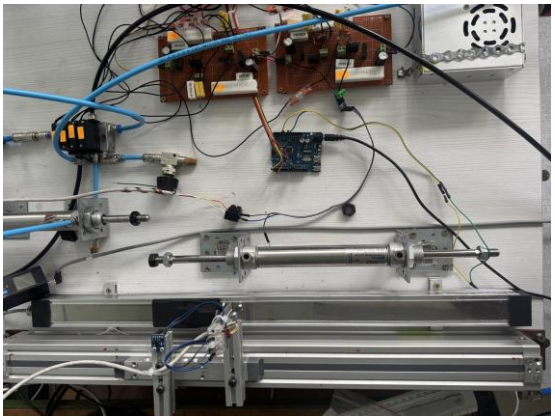


Fig. 1*. Photo of the electro-pneumatic positioning system with an Arduino Uno microcontroller*

Electrical connection of the control electronics

The power circuits and the signal circuits of the installation are kept strictly separate, and only the ground is common to both.

The controller is supplied from the USB port, or from an external source of 7–12 V connected to the VIN pin. It must never be connected to the 24 V rail, since that would destroy the on-board voltage regulator at once.

The solenoid coils are fed from a separate 24 V unit. One coil draws about 3.33 A, so that four coils energised simultaneously would require 13.3 A; in the operating sequence described above only one pair is active at a time, the current being approximately 6.7 A, and a supply rated at 150–200 W is therefore sufficient.
The two supplies meet at a single common point, the ground bus being arranged in a star configuration. It is essential that the 24 V return current does not flow through the board: a current of several amperes in a narrow copper trace produces a voltage drop which disturbs the position measurement of the ADC. For the same reason, optocoupler isolation between the logic and the power section is recommended.

An interface stage is necessary between the controller and the coils, because the microcontroller can deliver only 5 V at about 20 mA, whereas a solenoid requires 24 V at 3.33 A. As the shut-down routine reverses the polarity of the coil, a single transistor is not sufficient and an H-bridge is needed for every valve; integrated circuits such as the L298N, DRV8871 or VNH3SP30 may be used, or a discrete bridge built from four MOSFETs.

The assignment of the controller pins and the function of each connection are summarised in Table 1.

Table 1. Connection of the Arduino Uno to the driver stage and the actuator

Arduino Uno	Connection	Function
D9, D10, D11, D3	H-bridge PWM input (channels 1–4)	PWM duty cycle 0–90 %, carrier 10–100 Hz
D4–D7	H-bridge DIR input	Selection of the coil polarity
A0	Linear potentiometer output	Feedback voltage 0–5 V (100 mm/V)
5 V / GND	Supply of the transducer and logic	Ratiometric supply, ADC reference

Two control signals are used for each channel. With DIR = LOW the bridge applies the normal polarity and the valve is actuated; with DIR = HIGH and the PWM input held at 100 % the polarity is reversed and the forced shut-down is executed.

A flyback diode of the Schottky or ultrafast type is connected in parallel with every coil. This protection is not optional: without it the inductive voltage spike developed across a 24 V coil reaches several hundred volts and destroys the output stage within minutes.

The position signal is applied to input A0, and the voltage at this input must never exceed 5 V, because the pin has no protection against higher levels.

The transducer is energised from the same 5 V that serves as the reference of the ADC, so that a variation of the supply affects the sensor and the conversion ratio in the same way and does not appear as an error in the measured position.

With the internal reference the resolution of the converter corresponds to 4.88 mV, that is approximately 0.49 mm of carriage travel; a finer resolution would require an external reference (AREF) or a 12-bit converter.

It should also be noted that pins 9 and 10 are the outputs of Timer1. While the modulation is running they are occupied by the hardware PWM channels and cannot be used for any other purpose; assigning them to another function would disable one pair of valves.

The shut-down routine must reverse the polarity only of the valve pair that was active immediately before braking. If all four coils are inverted simultaneously, both pairs are energised at full power, their forces cancel one another, the braking effect is negligible and the coils are heated without purpose.

```
// ====================================================================
//  RMC-500 - closed-loop position and speed controller
//  RODLESS pneumatic cylinder, 500 mm stroke
//  Arduino Uno (ATmega328P) | 4x SMC SX10 24V/80W valves | 2 opposing pairs
//  Pair A: Timer1 HARDWARE PWM on pins 9/10 (fast, jitter-free)
//  Pair B: SOFTWARE PWM on pins 11/3  (Timer2 cannot reach 10 Hz)
//  Direction/polarity pins 4..7 (H-bridge) | Feedback A0 | CLI 115200
// ====================================================================

#include <Arduino.h>

// ------------------------------ pins ------------------------------
const uint8_t PA1 = 9,  PA2 = 10;        // OC1A / OC1B - owned by Timer1
const uint8_t PB1 = 11, PB2 = 3;         // OC2A / OC2B - kept as plain GPIO
const uint8_t DRV[4] = {4, 5, 6, 7};     // H-bridge direction / polarity
const uint8_t FBPIN = A0;                // linear potentiometric transducer

// ------------------------------ scaling ------------------------------
const float STROKE = 500.0;              // mm
const float MM_PER_V = STROKE / 5.0;     // 100 mm/V -> 1.20 V = 120 mm, 0.30 V = 30 mm
const float HYST = 0.03;                 // V (~3 mm): without it the pairs chatter
                                         // at the threshold and the coils overheat

// ------------------------------ state ------------------------------
uint16_t topA = 2499;                    // ICR1 for the default 100 Hz carrier
float    fA = 100.0, fB = 100.0;         // carrier frequencies, Hz
uint8_t  dA1 = 45, dA2 = 45;             // pair A duties, %  (0..90)
uint8_t  dB1 = 45, dB2 = 45;             // pair B duties, %  (0..90)
float    spa = 1.20, spb = 0.30;         // switching thresholds, V
uint32_t brkMs = 150;                    // forced de-energisation interval, ms
uint32_t pbPeriod = 10000;               // software PWM period, us (100 Hz)
float    vFb = 0.0;                      // filtered feedback voltage
bool     running = false, braking = false;
uint8_t  st = 0;                         // 0 idle | 1 EXT (pair A) | 2 RET (pair B)

// ------------------------------ helpers ------------------------------
static uint8_t clampPct(int v) { return (uint8_t)(v < 0 ? 0 : (v > 90 ? 90 : v)); }

// Pair A: duty lives in OCR1A/OCR1B. digitalWrite() has NO effect on pins
// 9/10 while the COM bits are set - the timer overrides the port latch.
// duty% = OCR1 / (ICR1 + 1) * 100
void setA(int a, int b) {
  dA1 = clampPct(a); dA2 = clampPct(b);
  OCR1A = (uint16_t)((uint32_t)dA1 * (ICR1 + 1UL) / 100UL);
  OCR1B = (uint16_t)((uint32_t)dA2 * (ICR1 + 1UL) / 100UL);
}

// Pair B: only the compare values are stored; the waveform is generated in loop().
void setB(int a, int b) { dB1 = clampPct(a); dB2 = clampPct(b); }

// The normal path is clamped to 0..90 %. Only the braking path is allowed
// to exceed it, so true 100 % is forced here and nowhere else.
void forceFullA() { OCR1A = ICR1; OCR1B = ICR1; }
void forceFullB() { digitalWrite(PB1, HIGH); digitalWrite(PB2, HIGH); }

// Pair B software PWM from micros(). Reason: 8-bit Timer2 has only discrete
// prescalers and bottoms out near 61 Hz, so a 10 Hz carrier is physically
// unreachable in hardware. Here any frequency from 10 to 100 Hz works.
// Timer2 must stay UNCONFIGURED, otherwise it would take over pins 11/3
// and digitalWrite() would stop controlling them.
void applyPairB() {
  if (braking) return;                   // held HIGH by forceFullB()
  uint32_t ph = micros() % pbPeriod;
  digitalWrite(PB1, ph < (uint32_t)dB1 * pbPeriod / 100UL ? HIGH : LOW);
  digitalWrite(PB2, ph < (uint32_t)dB2 * pbPeriod / 100UL ? HIGH : LOW);
}

// 8-sample average kills most of the ADC noise for well under 1 ms.
// 1 LSB = 5/1023 = 4.88 mV = approx. 0.49 mm of resolution.
float readFb() {
  uint32_t s = 0;
  for (uint8_t i = 0; i < 8; i++) s += analogRead(FBPIN);
  return (s / 8.0) * 5.0 / 1023.0;
}

void setFreqA(float f) {
  fA = (f < 10.0) ? 10.0 : (f > 100.0 ? 100.0 : f);
  // ICR1 = 16e6 / (prescaler * f) - 1   ->  100 Hz = 2499, 10 Hz = 24999
  // Atomic: changing TOP mid-count would emit one wrong-length period.
  noInterrupts();
  ICR1 = (uint16_t)(16000000UL / (64UL * (uint32_t)fA) - 1UL);
  interrupts();
  setA(dA1, dA2);                        // duty must be rescaled to the new TOP
}

void setFreqB(float f) {
  fB = (f < 10.0) ? 10.0 : (f > 100.0 ? 100.0 : f);
  pbPeriod = (uint32_t)(1000000.0 / fB);
}

// Forced de-energisation. Reversing the coil polarity needs an H-bridge -
// a single MOSFET cannot invert the current direction. The 100 % duty must be
// time-limited: an 80 W coil left at full power overheats within seconds.
void doBrake() {
  braking = true; running = false; st = 0;
  for (uint8_t i = 0; i < 4; i++) digitalWrite(DRV[i], HIGH);   // reverse polarity
  forceFullA(); forceFullB();
  delay(brkMs);
  setA(0, 0); setB(0, 0);
  digitalWrite(PB1, LOW); digitalWrite(PB2, LOW);
  for (uint8_t i = 0; i < 4; i++) digitalWrite(DRV[i], LOW);
  braking = false;
}
```

```
// Two-position cycle with hysteresis:
//   EXT: pair A energised until the feedback RISES to spa -> hand over to B
//   RET: pair B energised until the feedback FALLS to spb -> hand back to A
void cycle() {
  if (braking) return;
  if (!running) { setA(0, 0); setB(0, 0); return; }
  if (st == 1) {
    setA(dA1, dA2); setB(0, 0);
    if (vFb >= spa - HYST) { st = 2; setA(0, 0); setB(dB1, dB2); }
  } else if (st == 2) {
    setA(0, 0); setB(dB1, dB2);
    if (vFb <= spb + HYST) { st = 1; setA(dA1, dA2); setB(0, 0); }
  }
}

// ------------------------------ CLI ------------------------------
void usage() {
  Serial.println("F1 <Hz> | F2 <Hz> | D1..D4 <%> | SPA <V> | SPB <V> | BRK <ms> | GO | STOP | STATUS");
}

void status() {
  // Same formula as the setup, otherwise the two would disagree:
  // f = 16000000 / (64 * (ICR1 + 1))
  Serial.print("F1="); Serial.print(16000000.0 / (64.0 * (ICR1 + 1)), 1);
  Serial.print(" Hz  F2="); Serial.print(1000000.0 / pbPeriod, 1);
  Serial.print(" Hz   D1-4="); Serial.print(dA1); Serial.print('/');
  Serial.print(dA2); Serial.print('/'); Serial.print(dB1); Serial.print('/');
  Serial.print(dB2); Serial.print(" %");
  Serial.print("   V="); Serial.print(vFb, 3);
  Serial.print("  POS="); Serial.print(vFb * MM_PER_V, 1);
  Serial.print(" mm   SPA="); Serial.print(spa, 2);
  Serial.print(" SPB="); Serial.print(spb, 2);
  Serial.print("   BRK="); Serial.print(brkMs); Serial.print(" ms");
  Serial.print("   PAIR="); Serial.print(st == 1 ? "A" : st == 2 ? "B" : "-");
  Serial.print("  ST="); Serial.println(!running ? "IDLE" : (st == 1 ? "EXT" : "RET"));
}
void execCmd(char *line) {
  char *t = strtok(line, " \t\r\n");
  if (!t) return;
  for (char *p = t; *p; p++) if (*p >= 'a' && *p <= 'z') *p -= 32;  // case-insensitive
  char *a = strtok(NULL, " \t\r\n");

  if      (!strcmp(t, "F1") && a) setFreqA(atof(a));
  else if (!strcmp(t, "F2") && a) setFreqB(atof(a));
  else if (!strcmp(t, "D1") && a) setA(atoi(a), dA2);
  else if (!strcmp(t, "D2") && a) setA(dA1, atoi(a));
  else if (!strcmp(t, "D3") && a) setB(atoi(a), dB2);
  else if (!strcmp(t, "D4") && a) setB(dB1, atoi(a));
  else if (!strcmp(t, "SPA") && a) { float v = atof(a); spa = (v < 0) ? 0 : (v > 5 ? 5 : v); }
  else if (!strcmp(t, "SPB") && a) { float v = atof(a); spb = (v < 0) ? 0 : (v > 5 ? 5 : v); }
  else if (!strcmp(t, "BRK") && a) brkMs = strtoul(a, NULL, 10);
  else if (!strcmp(t, "GO"))  { running = true; st = 1; setA(dA1, dA2); setB(0, 0); }
  else if (!strcmp(t, "STOP")) doBrake();
  else if (!strcmp(t, "STATUS")) status();
  else usage();
}

// ------------------------------ main ------------------------------
void setup() {
  Serial.begin(115200);
  for (uint8_t i = 0; i < 4; i++) pinMode(DRV[i], OUTPUT);
  pinMode(PA1, OUTPUT); pinMode(PA2, OUTPUT);
  pinMode(PB1, OUTPUT); pinMode(PB2, OUTPUT);

  // Timer1: Fast PWM mode 14 (TOP = ICR1), prescaler 64, non-inverting.
  // WGM12 is required for mode 14; CS11|CS10 together give the 64 divider.
  noInterrupts();
  TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM11);
  TCCR1B = _BV(WGM13) | _BV(WGM12) | _BV(CS11) | _BV(CS10);
  ICR1   = topA;
  OCR1A  = 0;  OCR1B = 0;
  interrupts();

  setA(0, 0); setB(0, 0);
  for (uint8_t i = 0; i < 4; i++) digitalWrite(DRV[i], LOW);
  Serial.println("RMC-500 ready - GO to start, STATUS to read");
}

void loop() {
  static char buf[48];
  static uint8_t n = 0;

  vFb = readFb();          // feedback first, so cycle() works on fresh data
  applyPairB();            // pair B waveform
  cycle();                 // two-position state machine

  while (Serial.available()) {
    char c = Serial.read();
    if (c == '\n' || c == '\r') { if (n) { buf[n] = 0; execCmd(buf); n = 0; } }
    else if (n < sizeof(buf) - 1) buf[n++] = c;
  }
}
```

***Fig. 2**. Listing of the control program of the Arduino Uno controller (RMC-500)*

The heating of the coils also limits continuous operation at a high duty cycle. A supply of 24 V applied to the 7.2 Ω winding gives a continuous current of 3.33 A, which corresponds to the value required for actuation, whereas holding the valve

open requires the current to be limited to the equivalent of 3–6 V — exactly what

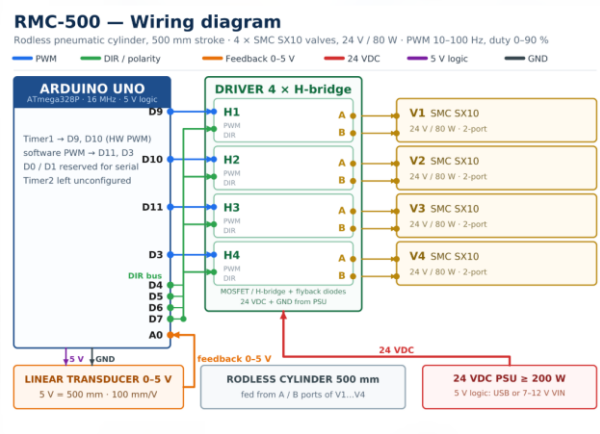


Fig. 3. *Wiring diagram of the electro-pneumatic positioning system with the Arduino Uno controller*

The methodology of the experiment is described in detail in [3].

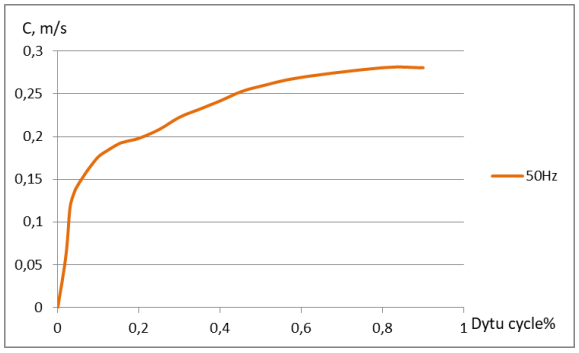


Fig. 4*. Speed response of the pneumatic cylinder at a frequency of 50 Hz with varying duty cycle.*

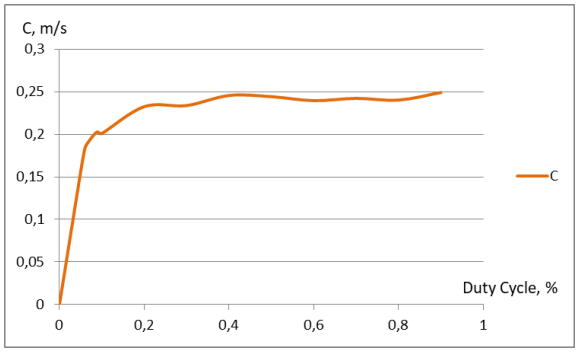


Fig. 5*. Speed response of the pneumatic cylinder at a frequency of 80 Hz with varying duty cycle.*

duty cycles below 90 % provide. Prolonged operation at a high duty cycle should therefore be accompanied by measurement of the coil temperature.

CONCLUSION

The paper presents an electro-pneumatic tracking system for precise position and speed control of a rodless pneumatic cylinder, implemented on the basis of a low-cost Arduino Uno microcontroller. The developed test bench (Fig. 1) comprises a rodless pneumatic cylinder with a stroke of 500 mm, four high-speed two-port SMC SX10 valves and a linear potentiometric transducer used as the feedback sensor. The electrical architecture of the system, described in Table 1 and Fig. 2, separates the 5 V logic supply from the 24 V supply of the coils, employs a common star-point ground and uses an H-bridge driver for every valve, which makes it possible to reverse the polarity of the coil current whenever this is required. The control algorithm (Fig. 3) generates four independent PWM channels with a carrier frequency adjustable from 10 Hz to 100 Hz and a duty cycle from 0 % to 90 %, and it closes the position loop by comparing the transducer voltage with the specified switching levels.

The experimental results confirmed that smooth modulation of the flow rate through the valves, obtained by means of PWM instead of discrete on/off switching, has a favourable effect on the dynamics of the drive and on the magnitude of the shock loads at the end positions. Positioning accuracy of the order of 0.49 mm and independent control of the carriage speed were achieved with a single inexpensive microcontroller, without the use of

proportional pneumatics. Since a duty cycle below 90 % also limits the current in the coils, the electrical power consumed is lower than with conventional on/off control and the heating of the solenoids is reduced. The forced de-energisation procedure, based on polarity reversal at a 100 % duty cycle, accelerates the release of the piston and limits the overshoot of the moving part. The investigations also showed, however, that the achievable accuracy is limited by the 10-bit ADC of the microcontroller, by the compressibility of the air and by the friction in the cylinder. A further improvement of the system could therefore be achieved by using an external reference voltage or a 12-bit converter, by introducing pressure sensors into the cylinder chambers, and by implementing more advanced algorithms, such as PID or adaptive control, on the same microcontroller. In conclusion, the combination of high-speed on/off valves with a microcontroller operating under PWM control offers a practical compromise between cost and achievable accuracy in the control of pneumatic actuators, and it can be applied in automated assembly, handling and packaging equipment.

ACKNOWLEDGEMENTS:

This research was funded by the European Regional Development Fund within the OP "Research, Innovation and Digitalization Programme for Intelligent Transformation 2021-2027", Project No BG16RFPR002-1.014-0005 Center of competence "Smart mechatronics, eco- and energy-saving systems and technologies"

REFERENCES

[1] G. Iliev, Hr. Hristov "Modelling and Simulation of Dynamic Processes in an Electropneumatic Positioning System with High-Speed Valves and PWM Control" 2025, Journal of Physics Conference Series 3127(1),012012; https://iopscience.iop.org/article/10.1088/1742- 6596/3127/1/012012 https://doi.org/10.1088/1742-6596/3127/1/012012

[2] G. Iliev, H. Hristov, "Modelling and Simulation of Dynamic Processes of Pneumatic Lines" Environment. Technology. Resources 14th International Scientific and Practical Conference. June 15-16, 2023, Rezekne Academy of Technologies, Rezekne, Latvia,Page 112-118 Online ISSN:2256-070X https://journals.rta.lv/index.php/ETR/article/view/7190 https://doi.org/10.17770/etr2023vol3.7190

[3] G. Iliev, Hr. Hristov,"Speed Control Of Pneumatic Cylinders Using High Speed 2 Port On/Off Valves With Pulse Width Modulation",Environment.Technologies. Resources. Proceedings of the International Scientific and Practical,Conference.3, (Jun.2024),pp.95–100. https://journals.rta.lv/index.php/ETR/article/view/8117 https://doi.org/10.17770/etr2024vol3.8117

[4] Ivaylo Nikolaev, Ivan Georgiev , Slavi Georgiev and Georgi Iliev "Numerical Investigation of the Actual Volumetric Flow Rate and Volumetric Efficiency and Optimization of the Geometric Parameters of a Three-Rotor Pump with Lantern Meshing—Part I; MDPI Machines 2026, 14, 591. https://doi.org/10.3390/machines14060591

[5] L. Zhonglin, Z. Tianhong, X. Qi, W. Qingyan. "Electro-pneumatic position tracking control system based on an intelligent phase-change PWM strategy" Journal of the Brazilian Society of Mechanical Sciences and Engineering (2018) 40:512 https://doi.org/10.1007/s40430-018-1431-y

[6] N. Bruno, & Y. Zhu, & C. Liu, & Q. Gao, & Y. Li, (2019). Development of a piezoelectric high speed on/off valve and its application to pneumatic closed-

loop position control system. Journal of Mechanical Science and Technology. 33. 10.1007/s12206-019-0521-9. https://doi.org 10.1007/s12206-019-0521-9

[7] M. Smaoui, X. Brun, and D. Thomasset, “Robust position control of an electropneumatic system using second order sliding mode” IEEE International Symposium on Industrial Electronics 2004;PROCEEDINGS OF THE IEEE-ISIE 2004, VOLS 1 AND 2 ,pp.429-434.

[8] P. Dimitrov, M. Alexandrova, "Automatic Control of Fluids in Electropneumatic System" 2025, Annual Journal Of Technical University Of Varna Bulgaria 9(№2):95-103.https://aj.tu-varna.bg/index.php/ajtuv/article/view/365/123 https://doi.org/10.29114/ajtuv.vol9.iss2.365

[9] Dichev D., T. Karadzhov, I. Zhelezarov, H. Hristov, G. Iliev, H. Kovachev “Genetic–Neural Adaptation of the Kalman Filter Process Noise Covariance for Dynamic Tracking in Electro-Pneumatic Systems” MDPI 2026, 16 (17), 8706; https://doi.org/10.3390/app16178706

[10] M. Sukop, R. Jánoš, J. Brna, J. Melko, “Design and Optimization of Universal Inspection Cells with Energy-Efficient Pneumatic Actuation” January 2026, Actuators 15(1):36 https://www.mdpi.com/2076-0825/15/1/36?utm_source=researchgate.net&utm_medium=article https://doi.org/10.3390/act15010036.