

Some aspects of research on an Arduino Uno-based CNC plotter machine

Konstantin Nechval¹  ^{A-G}, Vladimirs Reiskarts¹  ^{A-G}, Rafał Chatys²  ^{A,D-G}

¹ Riga Aeronautical Institute, 9 Mezhkalna str., Riga, LV-1058, Latvia

² Kielce University of Technology, al. Tysiąclecia Państwa Polskiego 7, 25-314 Kielce, Poland

Original article

Abstract

Due to their simple kinematics, accessible components, and low operating costs, CNC plotters are among the most affordable and easy-to-use CNC devices. This makes them practically indispensable in universities and laboratories, where they are primarily used for educational purposes, such as learning the basics of CAD/CAM, interpreting control programs, and understanding the fundamental principles of servo drives and feedback systems. This article discusses both the advantages and the limitations of CNC plotters and laser systems. A method for improving pen control throughout the entire operation process is proposed, addressing a limitation commonly observed in CNC machines and 3D printers. Typically, control is performed using end sensors only at the initial stage of processing and calibration. During subsequent operation, such systems function in an open-loop or weakly monitored mode.

Introduction

Numerical control (NC) systems are microprocessor-based controllers that process logical instructions and interface with a computer. Input data may be provided in the form of program code, textual tasks, or graphical descriptions [1]; the controller then converts them into machine commands understandable to actuators. At this stage, interpretation, trajectory planning, and signal generation for the drives are performed. A plotter [2] is a device used for outputting graphical information, in which lines are drawn using a writing instrument (e.g., a pen or marker) on flat, solid materials. Unlike traditional inkjet or laser printers, a plotter creates images using a trajectory-based method:

Keywords

- plotter
- sensing
- measurement
- control loop design
- robustness

Authors contributions

- A – Preparation of the research project
- B – Assembly of data for the research undertaken
- C – Conducting of statistical analysis
- D – Interpretation of results
- E – Manuscript preparation
- F – Literature review
- G – Revising the manuscript

Corresponding author

Konstantin Nechval
e-mail: k.nechval@rai.lv
Riga Aeronautical Institute
9 Mezhkalna str.
Riga, LV-1058
Latvia

Article info

Article history

- Received: 2025-09-17
- Accepted: 2026-03-20
- Published: 2026-03-30

Publisher

University of Applied Sciences in Tarnow
ul. Mickiewicza 8, 33-100 Tarnow, Poland

User license

© by Authors. This work is licensed under a Creative Commons Attribution 4.0 International License CC-BY-SA.

Financing

This research did not receive any grants from public, commercial or non-profit organizations.

Conflict of interest

None declared.

the tool moves along specified coordinates to reproduce vector contours. A CNC plotter is typically a two-axis (2D) device with X-Y motion, designed for the vector output of text and images using a pen on various flat surfaces. A typical technological cycle includes preparing a layout in a CAD system, generating control commands (e.g., in G-code format), interpreting them within the controller firmware, and executing the trajectory via stepper or servo drives. Application areas include rapid prototyping of printed circuit board (PCB) layouts (e.g., preliminary routing and template generation), development of graphical symbols and logos, production of templates, and educational demonstrations of automated positioning principles.

The aim of this study is to design and analyze a low-cost CNC plotter based on the Arduino Uno platform, with a focus on motion control optimization and pen-position accuracy. The initial stage of the design process involves determining the key parameters of the future system, assuming a novel configuration and control approach. These parameters form the basis for further engineering analysis and system development.

In the field of motion control systems, there is significant competition between various approaches [3, 4]—ranging from lead screw mechanisms and rack-and-pinion systems to belt drives and crank-slider mechanisms—each optimized for specific ranges of load, speed, and accuracy. The conversion of rotary motion from a drive motor into linear motion can be achieved using different kinematic schemes. There is no universal solution; the selection depends on application requirements and performance criteria. Economic factors are equally important, including not only the initial cost of components but also service life, reliability, maintenance frequency, and the total cost of ownership over the project lifecycle. Consequently, high-precision and durable solutions are appropriate for some applications, while simpler and more cost-effective solutions may be sufficient for others. In practice, factors such as load profile and dynamics (acceleration and speed), permissible backlash, repeatability, stiffness, efficiency, operating conditions (e.g., dust, temperature, vibration), as well as maintenance and integration requirements, must all be considered.

Materials and methods

The design and selection of system components require appropriate guidelines and characteristic features. These are based on the planned performance parameters of the device upon completion. The specific

assumptions and requirements used for the present design are listed below:

- Moving table mass: 0.5 kg.
- Stroke (travel): 420 to 500 mm.
- Maximum linear speed: 0.018 m/s (≈ 18 mm/s).
- Positioning accuracy: ≤ 0.20 mm per 1000 mm of travel.
- Positioning repeatability: 20 μm (as specified).
- Drive type: stepper motor.

To meet the required positioning accuracy of ± 0.20 mm per 1000 mm of travel, a screw should be selected such that its cumulative lead error over 1 m does not exceed ± 0.40 mm. For the actual stroke of 420 mm, this corresponds to an allowable linear error of approximately ± 0.11 mm, assuming proportional scaling. To minimize systematic error in practice, the following should be specified:

- Lead accuracy: cumulative lead error $\leq \pm 0.20$ mm/m (or better).
- Backlash control: use a preloaded (anti-backlash) nut so that lost motion remains below the repeatability target.

Fundamentals

During the design process, particular attention should be paid to ensuring that the mechanical system does not exhibit excessive vibration. Therefore, kinematic sizing (speed vs. lead) is required. Let p be the screw lead (mm/rev) and v_{ax} the required linear speed, thus $n_{re} = (v_{ax}/p) \times 60$. Choose p such that the required motor speed n_{re} lies well within the stepper motor's torque-speed envelope and below the screw's critical speed for the selected length and end supports.

Resolution (also known as step size) is the smallest possible distance the system can theoretically move. For a 200-step/rev motor, the linear increment i per full step is: $\Delta i_{step} = s / 200$ with a microstepping factor m : $\Delta i_{ustep} = s / (200 \times m)$. Select s and m such that the commanded increment and the mechanical backlash are comfortably smaller than the 10 μm repeatability requirement, taking into account compliance and friction.

Load and durability considerations must also be addressed. For a moving mass of 0.5 kg, it should be verified that the selected screw/nut pair provides:

- adequate static and dynamic load capacity;
- acceptable efficiency (and thus sufficient motor torque margin);
- suitable materials and lubrication for the operating environment.

For low speeds (≈ 18 mm/s) and moderate loads, polymer-nut trapezoidal screws can be an economical

solution. Where higher accuracy, stiffness, and longevity are required, a preloaded ball-nut assembly is typically preferred.

In general, the formulas are derived from classical theoretical mechanics and remain standard. Presenting the full set of calculations is unnecessary, as the primary objective is to automate control across all operating modes of the device.

Control circuit, motion-control workflow, and hardware architecture

The block diagram of the plotter is presented in Figure 1. The control circuit defines all interconnections and components required to operate the plotting machine. An Arduino serves as the central controller, coordinating motion and peripheral functions. Actuation is provided by stepper motors (for X-Y motion) and a servo motor (for pen up/down).

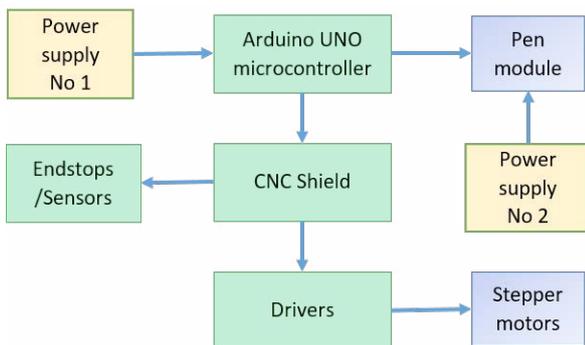


Figure 1. Schematic diagram of the module connection (based on [12])

An example of this approach includes the validation of a control algorithm for an EDS electric vehicle [5] or research into gesture-based (including voice-controlled [6]) systems based on the Arduino platform. The main objective of such systems is to improve speed and torque distribution between the wheels during manoeuvring, as well as to enable smartphone-controlled robotic systems.

The Arduino controller executes the motion firmware and generates command signals [7]. The CNC Shield breaks out the Arduino I/O into standardized headers and buses for motion systems. It routes STEP/DIR/EN signals to the stepper drivers, provides terminals for limit/home switches, and exposes control pins for spindle/coolant triggers (ON/OFF). The shield also

distributes the external motor supply voltage to the stepper drivers.

With regard to the stepper drivers (A4988), one driver per axis converts STEP/DIR inputs from the Arduino into phase currents for the stepper motors. Microstepping is configured via jumpers on the shield, and driver current is set with the onboard potentiometer. For servo control, a PWM signal from the Arduino drives the pen-lift servo, enabling rapid and repeatable transitions between drawing and travel moves.

Signal interfacing, power distribution, actuation and operating principle

The STEP/DIR/EN lines operate according to the following principle: signals are transmitted from the Arduino → CNC Shield → A4988 inputs for each stepper axis. Endstops/sensors are connected to the shield for homing and travel protection, while auxiliary outputs include shield pins for spindle and coolant control (which may be repurposed as general-purpose triggers, e.g., pen-lift interlocks).

With regard to power distribution, an external DC supply feeds the CNC Shield, which in turn powers the A4988 modules and stepper motors. The Arduino is powered via USB or a regulated input. The software toolchain and data flow can be presented as follows:

Arduino Uno → CNC Shield / stepper driver → motors (X/Y/Z)
 ↓
 Sensors: HC-SR04 (distance) + pen status sensor → feedback (optional)
 ↓
 Subsystem MD (pen effector) → second-order model → position $y(t)$

The following steps apply:

1. Design authoring: The drawing or part model is created in Inkscape (or similar CAD software).
2. CAM/Post-processing: The graphic is converted to G-code (and, where applicable, complementary M-codes for auxiliary functions).
3. Transmission: The G-code file is streamed to the Arduino using Universal G-code Sender (UGS).
4. Interpretation: The Arduino firmware (e.g., a GRBL-class interpreter) parses the incoming G/M-codes, plans motion, and schedules step pulse.

Regarding actuation:

1. For each linear move, the Arduino emits appropriate STEP/DIR pulse trains to the A4988 drivers,

which advance the stepper motors along their respective axes.

2. For pen operations, the controller outputs a PWM command to the servo, lifting or lowering the pen at designated points.

As G-code blocks are executed, the corresponding axis motors are driven to trace the commanded tool-path on the work surface. The servo establishes contact (pen down) during drawing segments and retracts (pen up) during rapid traverses. Limit switches and enable signals safeguard motion, while optional spindle/coolant outputs provide standardized triggers if additional peripherals are integrated.

This architecture yields a modular, low-cost motion platform in which the Arduino/CNC Shield manages I/O and timing, the A4988 drivers handle motor power electronics, and the Inkscape → G-code → UGS tool-chain provides a straightforward path from design to plotted output.

Results and discussion

This work utilizes an open-loop control system modeled as a second-order system, employing small-signal linearization of nonlinear dynamics and step-response simulation. The model is inspired by standard control engineering techniques. Modeling, simulation, and control strategies are generally based on the guidelines presented in [8].

Control loop design considerations have been adopted according to Barton [9]. The subsystem responsible for pen positioning receives signals from the following sensors and devices:

- sensor A: HC-SR04 ultrasonic distance sensor;
- sensor B: pen effector status (in use / not in use);
- control panel signals: user-defined input map.

The stored data include pressure, pressing time, and the effector operating condition. Considering a standard second-order system [10], the graphs presented in Figures 2 and 3 are obtained.

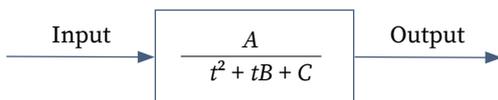


Figure 2. Standard second-order system

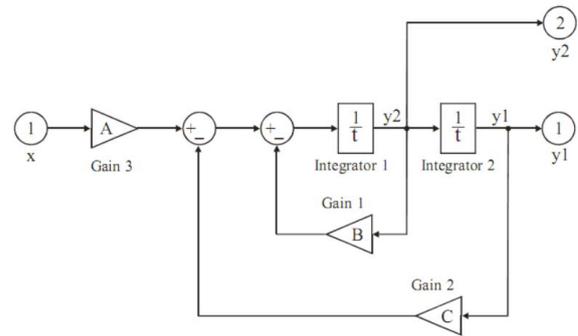


Figure 3. Schematic of the standard second-order system

The dynamic equation of the pen effector (MD), given in (1), can be expressed as a nonlinear function of four variables $f(y, y', y'', n)$, where y denotes the pen tilt/position (mm) y' and y'' represent its first and second derivatives with respect to time (velocity and acceleration), and n is the control signal (e.g., PWM from the Arduino or motor input signal). Due to the nonlinear nature of this relationship, direct analysis and control design are not straightforward. Therefore, the equation is linearized around a selected operating point using the small-deviation (small-signal) method. This approach assumes that system variables vary only slightly around their nominal values. The linearization is performed by expanding the nonlinear function into a Taylor series and retaining only the first-order terms, which yields an approximate linear model suitable for further analysis [11]:

$$f(y, y', y'', n) = f(y_{st}, 0, 0, n_{(st)}) + \left(\frac{\partial f}{\partial y} \right)_{st} + \left(\frac{\partial f}{\partial y'} \right)_{st} + \left(\frac{\partial f}{\partial y''} \right)_{st} + \left(\frac{\partial f}{\partial n} \right)_{st} = 0 \quad (1)$$

Here, $n_{(st)}$ should be considered as a control signal (e.g. PWM from Arduino or engine RPM). In this case $f(y, 0, 0, n_{(st)}) = 0$ (static equilibrium equation). After calculating the partial derivatives, the linearized equation takes the form (2):

$$T_2^2 \Delta y''(t) + T_1 \Delta y'(t) + \Delta y(t) = k \Delta n_r(t) \quad (2)$$

or, equivalently, in operator form (3):

$$T_2^2 p^2 + T_1 p + 1) \Delta y(p) = k \Delta n(p) \quad (3)$$

where:
 T —the time constant;
 k —the gain (coefficient) of the MD;
 Δ —denotes the deviation from the equilibrium point.

The square of the time constant (having considered the inertia of the moving elements of the pen effector), takes the following form (4):

$$T_2^2 = \frac{\left(\frac{\partial f}{\partial y}\right)_{st}}{\left(\frac{\partial f}{\partial y}\right)_{st}} \quad (4)$$

While the gain of the MD (5) can be expressed as:

$$k = \frac{\left(\frac{\partial f}{\partial n_r}\right)_{st}}{\left(\frac{\partial f}{\partial y}\right)_{st}} \quad (5)$$

When damping is taken into account, the effective time constant is given by (6):

$$T_1 = \frac{\left(\frac{\partial f}{\partial y}\right)_{st}}{\left(\frac{\partial f}{\partial y}\right)_{st}} \quad (6)$$

The differential operator is defined as (7):

$$p^m \Delta y = \frac{d^m \Delta y}{dt^m} \quad (7)$$

The reference model is defined using the following second-order differential equation (8), which is the linearized differential equation of the oscillatory link with a single step input action:

$$T_2^2 \ddot{h}(t) + T_1 \dot{h}(t) + h(t) = k \quad (8)$$

where:

$$\bar{h}(t) = \sum_{i=1}^2 \bar{C}_i e^{p_i t} \quad (9)$$

The above mentioned formula is a general solution of the homogeneous differential equation (pi are the roots of the characteristic equation). Thus, the general solution of the homogeneous differential equation is as follows (10):

$$\begin{aligned} \bar{h}(t) &= \bar{C}_1 e^{(-A+iv)t} + \bar{C}_2 e^{(-A-iv)t} = e^{-At} [\bar{C}_1 e^{ivt} + \bar{C}_2 e^{-ivt}] = \\ &= e^{-At} [\bar{C}_1 (\cos vt + i \sin vt) + \bar{C}_2 (\cos vt - i \sin vt)] = \\ &= e^{-At} [(\bar{C}_1 + \bar{C}_2) \cos vt + i(\bar{C}_1 - \bar{C}_2) \sin vt] = \\ &= e^{-At} [C_1 \cos vt + C_2 \sin vt] \end{aligned} \quad (10)$$

where the individual quantities denote:

t —time [s];

$h(t)$ —displacement (deflection) of the system as a function of time;

A —damping coefficient [1/s], determining the rate of decay of vibration amplitude;

ν —damped natural frequency [rad/s];

C_1, C_2 —integration constants determined from the initial conditions of the system (initial displacement and velocity).

The variables are linked with the mechanical parameters. The exponential term e^{-At} represents damping, causing a gradual decay of the vibration amplitude over time. The harmonic components $\sin(\nu t)$ and $\cos(\nu t)$ describe the oscillatory nature of the motion. Altogether, the equation represents damped vibrations whose amplitude decreases exponentially with time

A particular solution of a non-homogeneous differential equation is sought as follows. The characteristic polynomial takes the following form (11):

$$h^*(t) = At^2 + Bt + C \quad (11)$$

respectively (12)

$$\dot{h}^*(t) = 2At + B; \quad \ddot{h}^* = 2A \quad (12)$$

The equation is reduced to the following form (13):

$$\begin{aligned} 2T_2^2 A + T_1(2At + B) + At^2 + Bt + C &= k_{uy} \Rightarrow \\ \Rightarrow At^2 + (2At_1 + B)t + 2T_2^2 A + T_1 B + C &= k_{uy} \end{aligned} \quad (13)$$

It follows that

$$\begin{aligned} A = 0; 2AT_1 + B = 0 \Rightarrow B = 0; 2T_2^2 A + T_1 B + C &= k_{uy} \Rightarrow \\ \Rightarrow C = k_{uy} \end{aligned} \quad (14)$$

The equation then takes the form (15):

$$h^*(t) = k_{md} \quad (15)$$

The general solution of a non-homogeneous differential equation equals (16):

$$h(t) = e^{-At} [C_1 \cos \nu t + C_2 \sin \nu t] + k \quad (16)$$

The integration constants C_1 and C_2 are determined by equation (17) on the basis of the initial conditions: when $t = 0, h(t) = 0$

$$\begin{aligned} 0 &= C_1 + k \Rightarrow C_1 = -k \\ \dot{h}(t) &= -A e^{-At} [C_1 \cos \nu t + C_2 \sin \nu t] + e^{-At} [-C_1 \nu \sin \nu t + C_2 \nu \cos \nu t] \Rightarrow \\ \Rightarrow 0 &= -AC_1 + \nu C_2 \Rightarrow C_2 = \frac{-Ak}{\nu} \end{aligned} \quad (17)$$

Finally, the time response of the MD (control element) to a step input is given by (18):

$$h(t) = k \left[1 - e^{-At} \left(\frac{A}{\nu} \sin \nu t + \cos \nu t \right) \right] \quad (18)$$

The response shown in Figure 4 exhibits overshoot and damped oscillations, indicating that the system is stable but underdamped. In practice, this means that when a target position is specified (e.g., axis or

pen position), the system initially overshoots and then undergoes decaying oscillations before reaching equilibrium.

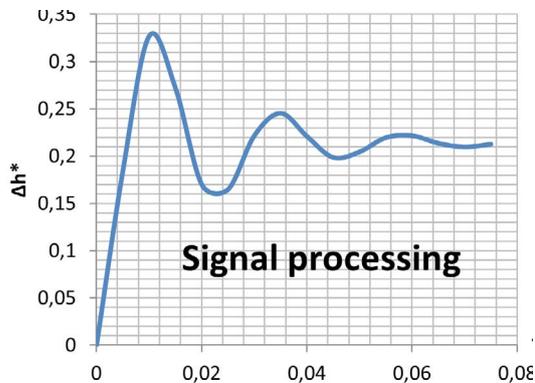


Figure 4. Approximate diagram of the control system in oscillatory process

The desired system response can be achieved using various control synthesis techniques, such as root locus or frequency-domain compensator design. However, these methods are often abstract and may not directly reflect the physical nature of the system.

In this work, the method of inverse dynamics is employed. Its practical significance lies in predicting system behavior over time, particularly the rate of vibration decay and oscillation frequency. This model enables assessment of the impact of vibrations on positioning accuracy and supports the selection of system parameters (e.g., mass, stiffness, damping) to minimize undesired oscillations.

The synthesized control algorithm allows direct incorporation of desired motion parameters into the control law. Simulation results indicate that, for the considered masses and damping values, the system exhibits a slightly underdamped response with small overshoot, which is typical for lightweight plotter mechanisms.

Although the Arduino-based stepper system operates in an open-loop manner and does not explicitly account for these dynamics, the proposed model provides valuable insight into potential vibration effects, especially at higher operating speeds.

Conclusions

Based on the simulation results obtained in Excel, an approximate response of the control system was generated. However, it should be noted that the actual system response may differ from the simulated curve due to model simplifications and assumptions.

The results presented in this work are primarily theoretical and generalized. Their practical implementation requires further development, including experimental validation and system testing.

This study is intended to provide a conceptual and analytical foundation rather than a complete implementation guide. Numerous control strategies, including PID controllers and closed-loop feedback systems, are widely used in practice and are well documented in the literature. The approach presented here is oriented toward low-cost and accessible solutions, offering a starting point for further development of DIY and research-oriented projects.

Detailed electronic schematics of the microcontroller system are not included, as the focus of this work is on system modeling and control analysis rather than hardware design. Readers interested in implementation details are encouraged to consult relevant data sheets and technical documentation.

Due to the scope of this paper, not all aspects of the automated system can be discussed in detail. Comprehensive treatments of control algorithms, system architectures, and implementation techniques can be found in specialized literature. Nevertheless, the presented model provides useful insight into system dynamics and may support the design and optimization of similar motion-control systems.

References

- [1] Stefańska A, Dixit S. Digital fabrication of contemporary structures in architectural design optimization. In: Skibniewski MJ, Hajdu M, editors. Proceedings of the Creative Construction e-Conference 2020. Budapest: Budapest University of Technology and Economics; 2020. p. 71–75. <https://doi.org/10.3311/CCC2020-057>.
- [2] Patil P, Lidhade P, Khamkhar S, Mane Y, Dhole A, Patil S. Design and development of CNC plotter machine. International Research Journal of Modernization in Engineering Technology and Science. 2020;2(9):1069–1079. Available at: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_9_september_2020/3812/1628083151.pdf.
- [3] Yusop N, Mocketar NA, Sadikan SFN. (2024). Development of Arduino applications for IoT applications in software engineering education: A systematic literature review. Bulletin of Electrical Engineering and Informatics. 2024;13(3):1824–1831. <https://doi.org/10.11591/eei.v13i3.4506>.
- [4] Heo G. Arduino-compatible modular kit design and implementation for programming education. Advances in Science, Technology and Engineering Systems. 2020;5(5):295–301. <https://dx.doi.org/10.25046/AJ050537>.

- [5] Taha Z, Aydın K. Development and validation of an Arduino-based control algorithm for an electronic differential in quad in-wheel hub motor electric vehicles. *New Energy Exploitation and Application*. 2025;4(2):215-227. <https://doi.org/10.54963/nee.v4i2.1557>.
- [6] Alagulakshmi A. (2023). Smart home automated control system using android application based on Arduino. *International Journal For Multidisciplinary Research*. 2023;5(6):1-7. <https://doi.org/10.36948/ijfmr.2023.v05i06.9306>.
- [7] Banzi M, Shiloh M, Jepson B. *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*. Sebastopol: Maker Media; 2015.
- [8] Haugen FA. *Modeling, Simulation and Control*. 21st edition. TechTeach; 2023. Available at: https://www.tech-teach.no/books/modsimcon/modsimcon_231221.pdf.
- [9] Barton GW, Chan WK, Perkins JD. Interaction between process design and process control: The role of open-loop indicators. *Journal of Process Control*. 1991;1(3):161-170. [https://doi.org/10.1016/0959-1524\(91\)85005-4](https://doi.org/10.1016/0959-1524(91)85005-4).
- [10] Tadashi Ishihara (1995) Discrete-time control systems design via loop transfer recovery. In: Leondes CT, editor. *Digital Control Systems Implementation Techniques*. San Diego-London: Academic Press; 1995. p. 163-198. [https://doi.org/10.1016/S0090-5267\(06\)80029-0](https://doi.org/10.1016/S0090-5267(06)80029-0).
- [11] Koung C-W, Macgregor JF. Identification for robust multivariable control: The design of experiments. *Automatica*. 1994;30(10):1541-1554. [https://doi.org/10.1016/0005-1098\(94\)90094-9](https://doi.org/10.1016/0005-1098(94)90094-9).
- [12] Nechval K, Reiskarts V, Chatys W. Research and development of CNC machine prototype for laser cutting and engraving. *Science, Technology and Innovation*. 2025;22(3):31-35. <https://doi.org/10.55225/sti.659>.