ENGINEER

international scientific journal

ISSUE 2, 2025 Vol. 3

E-ISSN
3030-3893

ISSN
3060-5172

SLIB.UZ
Scientific library of Uzbekistan

A bridge between science and innovation

TOSHKENT DAVLAT TRANSPORT UNIVERSITETI
Tashkent state transport university

ENGINEER

A bridge between science and innovation

E-ISSN: 3030-3893

ISSN: 3060-5172

VOLUME 3, ISSUE 2

JUNE, 2025

engineer.tstu.uz

TASHKENT STATE TRANSPORT UNIVERSITY

ENGINEER

INTERNATIONAL SCIENTIFIC JOURNAL

VOLUME 3, ISSUE 2 JUNE, 2025

Founder of the international scientific journal "Engineer" – Tashkent State Transport University, 100167, Republic of Uzbekistan, Tashkent, Temiryo'lchilar str., 1, office: 465, e-mail: publication@tstu.uz.

The "Engineer" publishes the most significant results of scientific and applied research carried out in universities of transport profile, as well as other higher educational institutions, research institutes, and centers of the Republic of Uzbekistan and foreign countries.

The journal is published 4 times a year and contains publications in the following main areas:

- Engineering;
- General Engineering;
- Aerospace Engineering;
- Automotive Engineering;
- Civil and Structural Engineering;
- Computational Mechanics;
- Control and Systems Engineering;
- Electrical and Electronic Engineering;
- Industrial and Manufacturing Engineering;
- Mechanical Engineering;
- Mechanics of Materials;
- Safety, Risk, Reliability and Quality;
- Media Technology;
- Building and Construction;
- Architecture.

Tashkent State Transport University had the opportunity to publish the international scientific journal "Engineer" based on the **Certificate No. 1183** of the Information and Mass Communications Agency under the Administration of the President of the Republic of Uzbekistan. **E-ISSN: 3030-3893, ISSN: 3060-5172.** Articles in the journal are published in English language.

Developing and validating reactive control for intelligent robot behaviors on the Robotrek platform

S.R. Ochilova[1][a], **M.R. Ochilov**[2][b]
[1]Karshi State Technical University, Karshi, Uzbekistan
[2]University of Information Technologies and Management, Karshi, Uzbekistan

Abstract: The present paper fills a knowledge gap that crucially exists on the imperfectly empirical knowledge of how the three basic intelligent behaviours (environmental adaptation, obstacle avoidance and path planning) scale, in terms of computational demands and degrees and speeds of success as applied to any given robot (using the publicly available Robotrek educational robotic kit (based on Tracduino microcontrollers, Robotrack IDE)). A reactive control software extension was implemented and tested with simulation and in experimentally physical tests. Among the core conclusions, it is possible to notice that the platform showed a high level of efficacy in structured settings: 92% success rate of avoiding obstacles at a threshold of 15 cm with a latency of less than 200 ms, and straight-line tracking within <2 cm average deviation. These findings confirm the potential of Robotrek in carrying out fundamental autonomous behaviours and serves to offer empirical benchmarks to reactive control paradigms in a resource-constrained hardware. Despite this, the study shows inherent shortcomings the research team including: limited ability to perceive the environment; failure to handle dynamic adversaries, or to optimize routes; and lack of robustness in unstructured contexts, which represent a sizeable breach between reactive capability and actual autonomous thought. The implications are future work in computer-vision integration (e.g. Raspberry Pi), and embedded AI (reinforcement learning, path planners), as well as energy-aware operation, to progress towards adaptable, deployable, robots.
Keywords: Robotrek platform, Tracduino microcontroller, STM32F407VGT6, ATmega2560, intelligent robot behavior, reactive control, sensor fusion, ultrasonic sensor (HC-SR04), IR line tracking

1. Introduction

The pace with which robotics is evolving requires platforms through which one can develop and be able to interact systematically with intelligent, autonomous movements such as environmental adaptations, obstacle avoidance, and path planning. Various learning systems, like Robotrek, that paired Tracduino microcontroller (STM32F407VGT6/ATmega2560) and Robotrack IDE software, are vital in prototyping and learning. The control of these abilities has become invaluable not only in the educational domain but also in logistics, service robotics, and explorations cases when the robots required effective navigation and interaction within the dynamic environment. The application of these types of intelligence to devices with resource-limited complexities, though, bring forth considerable issues that span theoretical control paradigms and realistic performance [1].

The interactions between the sensor-based perception, re-active decision-making and robust actuation are at the heart of this challenge. This study focuses on integration of the outputs of ultrasonic distance sensors and infrared line trackers to aid in obstacle avoidance and path following. The foundational theory is concentrated in reactive control structures wherein sensor conditions are directly projected to actuator outputs by pre-determined rules (e.g. “IF obstacle < 15 cm THEN turn right”). Technically very straightforward and computationally manageable in microcontrollers, such an approach contrasts with more complex cognitive architectures with planning or learning--abilities that usually lie beyond the reach of an entry-level environment. Simple reactive behaviors have previously been demonstrated on platforms like Arduino or LEGO Mindstorms; however, little has been done to understand how well such methods perform, their constraints, and scalability prospects within the scaffold of Robotrek, and more so in terms of real-time capability and reliability of a behavior against different working conditions[2].

The aim of this research is to determine how well we might manage to apply fundamental navigational behaviors using Reactive control based on the Robotrek platform, and come up with a computationally efficient point of reference to assess intelligent robotics capabilities on an initial level. The study was carried out as a rigorous and systematic study:

(1) theoretical review of reactive controllers, which included adaptation, avoidance, and planning proved;

(2) designing and implementing related algorithms in the Robotrack Integrated Development Environment (IDE) using a combination of a visual level programming of flow control and embedded C++ programming of sensor reading (getDistance) and moving commands (moveForward, turnRight, etc);

(3) the step (2) was verified using simulation in the Robotrek virtual environment;

(4) experimental testing of the step (2) using the hardware platform, The above protocols were formulated with an aim of testing the ability of the Robotrek platform to embrace reactive control on building intelligent behaviors.

The combination of Tracduino and Robotrack IDE called Robotrek can be used in this task according to empirical findings. The deployed module had 92 % success rate of obstacle avoidance, and tracked the planned path with average deviation of less than 2 cm, and responded within sub-200 ms in the real-time test. These results prove the

[a] https://orcid.org/0009-0003-5965-5933
[b] https://orcid.org/0009-0005-1198-9226

platform as viable with regard to reactive behavior, but at the same time show inherent limitations of reactive systems: limited perceptual scope, no planning or learning, unverified robustness in unstructured settings[3].

As a summary, the Robotrek platform provides a promising learning platform on entry level reactive intelligence, however further progress is required corresponding to a higher level of autonomy, especially in the areas of perception, cognition and improving the scalability. The present work provides an empirical benchmark and practical blueprint, thus making a starting point in subsequent work that combines sophisticated sensory areas (e.g., vision via Raspberry Pi), cognition-based algorithms (e.g., lightweight reinforcement learning, path planners like A*), and resilient onboard multi-sensor integration with or within the Robotrek system[4].

2. Research methodology

The current paper has used a multi-stage, systematic research design in the development and verification of software to regulate intelligent robot behaviour, i.e. environmental adaptation, obstacle avoidance and path planning on Robotrek framework. The study was initiated by the thorough theoretical overview of principles of intelligent robotic behavior and detailed evaluation of the Robotrek system hardware components (based on Tracduino microcontroller which combines features of Arduino and STM32) and software framework (Robotrack IDE). This guided the algorithm development stage where fundamental behavioral codes were coded in the Robotrack IDE system[5]. Its visual block based programming interface was used to define reactive rules (i.e. taking turns when an obstacle was detected within 15 cm by the ultrasonic sensor) and path-following methods (based on line sensor input). More complicated tasks, including the proper distance calculation (getDistance()) or sequences of activating motors (moveForward() and turnLeft(), and so forth.), were also written in C++ and integrated. This was followed by intensive simulation of the designed algorithms to be used inside Robotrek virtual world. It was a proving phase to be able to test collision avoidance, route following and decision-making logic on a small scale at repeatable conditions and with limited adverse environmental effects that do not recreate the situation that will exist during a physical deployment. The tested algorithms were subsequently implemented on the physical Robotrek platform, getting the Tracduino controller, ultrasonic and line-tracking sensors and DC motors with drivers in the action. It was experimentally tested in physical arenas, where specific behaviors were to be tested in real-time. The robot was put through its tests repeatedly in a methodical fashion with regard to a set of metrics based on which the robot was to be evaluated on the basis of its critical goals, the distance to which the robot reached avert the obstacles, the precision and the consistency with which the robot was able to line-track, the time it took in completing the path and the smoothness with which the transition between the autonomous choices of the robot was effected. A repeat cycle of improvements occurred: observations and quantitative data on the mechanical testing process updated the algorithms and possible fine tuning of either sensor thresholds or motor control parameters in the Robotrack IDE software[6]. This design, simulation and a physical test and analysis cycle kept on going until the robot exhibited consistently reliable, efficient and contextually apposite intelligent behaviors, which ascertained effectiveness of the built Robotrek-based control software [7].

3. Results and Discussion

The formed piece of software using the Robotrek base and the Tracduino microcontroller (STM32F407VGT6/ATmega2560) managed to show the fundamental smart properties of the device in simulated and real-life conditions and practice environmental adaptability, obstacle avoidance, and path planning. The test results in performance assessment indicated that the obstacle avoidance strategy installed at the 15cm detection threshold using the ultrasonic sensor (HC-SR04) has a success rate of 92 percent in which the evasive movements of the robot (stop, rotate, proceed) are reliable and the delay in the warning signal to motor movement is less than 200ms. Line following (sensing data using sensors, line tracking errors corrected with the proportional control logics (turnLeft(), turnRight())) showed controlled stability on the preset paths with minimum average error (<2cm) in a controlled experiment. The processing element (core behavioral loop) handled the sensor information (distance, line position) effectively to make smooth transitions between forward motion, avoidance of obstacles and changes in path, which characterizes the basic adaptive ability. Robotrack IDE was an excellent interface to perform quick prototyping with colorful block programming and the possibility to use it together with hardware control (button and led, Pixy camera), which helped to adjust parameters more proficiently (sensor thresholds, motor delays)[8].

This approach helps develop skills in algorithmic thinking, problem solving, and experimental practice. The language facilitates the development of prototypes for mobile robots and robotic devices, while its low entry threshold positively impacts the ability to teach robotics using high-level programming languages.

The availability of MicroPython programming for the Trackduino Pro platform allows users to create more complex and efficient programs and robotic systems, compared to visual development environments.

The TRACKDUINO Pro API for MicroPython is a firmware loaded onto the platform that provides a convenient environment for programming using the Robotrek hardware and software components. The extension package includes the following components:

- boot.py — A script that describes the initial operation of the platform immediately after startup. By default, its primary function is to launch the main script.
- main.py — A script that defines the logic of the robot controlled by the platform.
- pybcdc.ini — A configuration file containing platform settings for use on Windows computers.

The Trackduino module includes:

- Execution module — A set of driver modules for Robotrack actuators;
- Sensor module — A set of driver modules for Robotrack sensors;
- common.py file — A collection of general functions and classes used by other modules that simplifies platform programming;
- pins.py file — A file containing definitions for the

available pins on the platform, allowing for simple named access to required pins.

Table 1

Key Performance Metrics

Behavior	Metric	Result	Conditions
Obstacle Avoidance	Success Rate	92%	Diverse test environments
	Detection Threshold	15 cm (Ultrasonic HC-SR04)	
	Response Latency	< 200 ms	From detection to motor initiation
Path Following	Average Tracking Deviation	< 2 cm	Controlled lighting
	Sensor Thresholds (Analog)	<200 (Right), >800 (Left)	Line tracking sensor
General	Behavioral Transition Smoothness	Effective	Clear path → Obstacle → Path

In theory, this compound verifies the textual deployment of the reactive control paradigms to fundamental Wi-Fi behaviors on available educational environments in terms of viable sensor fusion (ultrasonic + line tracking) and rule-based decision-acting with live autonomy in limited environments. In practice, the Robotrek platform, especially Tracduino, proved to be a good platform, with plenty of computational resources, to run such algorithms, and practical connections (PWM, UART, I2C). Nevertheless, there are large limitations and gaps in information. The nature of this system uses only primitive sensors limiting their knowledge of the environment by missing fancy sensor features (e.g. vision, SLAM) to enable The complex task of recognizing obstacles or localizing themselves accurately. Its architectural simplicity, being purely reactive and following rules, does not support prediction or path planning, or learning, thus it is not possible to optimize its routes, or to efficiently deal with dynamic obstacles in a dynamic environment. More advanced behaviors, such as those more advanced sensor high-bandwidth sensing (vision, lidar) or advanced AI, may experience scalability issues because of the computational limits of the underlying microcontrollers. Moreover, energy efficiency solutions to long-term autonomy are in need of serious closure and robustness has not been demonstrated in high-dynamics, high-noise or unstructured real-life conditions[9-10].

Table 2

Key Knowledge Gaps and Research Needs

Domain	Specific Gap/Limitation	Critical Research Need
Perception & Sensing	Limited sensor fusion; no complex object recognition	Integrate computer vision (Raspberry Pi/TFT LCD) or 2D lidar
	No true localization beyond line following	Implement SLAM techniques
Cognition & Control	Purely reactive rules; no planning/learning	Incorporate path planning (A*, RRT), ML (Reinforcement Learning)
	Poor dynamic obstacle handling	Develop predictive algorithms & dynamic response strategies
System Capabilities	Computational limits for complex AI	Explore hybrid architectures (Tracduino + companion computer)
	Untested energy management	Research & integrate dynamic power management strategies
	Unproven robustness in real-world settings	Conduct rigorous testing in unstructured, dynamic environments
Advanced Functionality	Single-agent focus	Investigate multi-robot coordination (swarm/cooperative tasks)
	Lack of standardization	Explore bridging with ROS (Robot Operating System)

The repair of these gaps requires specific additional research. The most important thing to upgrade is perception; I can either connect computer vision via Raspberry Pi or special modules connected to Tracduino through UART/I2C, or use 2D lidar to achieve strong environmental mapping and object detection. Cognitive functions should go beyond being purely reactive with the inclusion of machine learning (e.g. lightweight Reinforcement Learning on microcontrollers such as TensorFlow Lite, or used in hybrid systems such as using a sidecar computer) to adaptively navigate and symbolic AI to plan. Direct implementation of the pathfinding algorithms (A*, Dijkstra, RRT) onto the Robotrek platform would enable real goal-based navigation with obstacle avoidance[12]. To study swarm characteristics, it is critical to perform research on multi-robotic coordination based on communicative protocols (Bluetooth, Wi-Fi via ESP32). Improved sensor fusion systems, such as methods of fusion IMU (as an example, we can refer to the location sensor discussed), wheel encoders, and possible vision/lidar, are an essential measure in correct state estimate. At the same time, exclusive development of energy-aware autonomy utilizing the low-power modes of STM32F4 is crucial to realistic implementation[12]. And lastly, tight formal verification of safety of control algorithm and wide use of robustness testing in practical complex environments are essential. Crossing the bridge between the Robotrek environment and ROS might as well be the gateway to a whole world of sophisticated tools and standard interfaces to cut down the pace of development achievement towards genuinely reactive, intelligent robots that will be able to work on dynamic, heterogenous environments. The study has served as a reasonable proof-of-concept, although the robust, intelligent autonomy will be difficult to achieve until great progress can be made in the areas of perception, cognitive architectures, and planning, integrated into the embedded systems[11].

4. Conclusion

The report therefore details the development and full validation of an intelligent robotic behaviour software component to the Robotrek platform that employs the key intelligent robotic behaviours; environmental adaptation, obstacle avoidance and path planning leading to an 92 % success rate when negotiating the avoidance of obstructions and generating stable path tracking with an average variation of less than 2 cm. These outcomes take place due to the running of reactive control algorithms available through the Robotrack Integrated Development Environment (IDE) on the Tracduino microcontroller (STM32F407VGT6/ATmega2560). The above results therefore empirically demonstrate that Robotrek platform is an effective teaching and prototyping tool to teach fundamental autonomous systems and also theoretically confirm the utility of sensor fusion (ultrasonic and line tracking) and rule-based decision to achieve real-time responsiveness in an indoor environment. Still, the study reveals a number of limitations inherent to the purely reactive model such as a limited environmental awareness, absence of ability to engage with changing challenges and formulate optimal route plans as well a lack of proven resilience in more challenging contexts. The combination of these shortcomings points out a significant shortcoming between base-level functioning and actual cognitive independence. In this regard, it is crucial to conduct more research to move closer to resilient and adaptive robots; the key areas of emphasis are to incorporate high-fidelity perception (computer vision using Raspberry Pi/LCD modules or lidar), incorporate AI-driven cognition (reinforcement learning, path planning algorithms like A* or RRT), design energy-approachable operational strategies, research multi-robot coordination, and explicitly validate the results in unmapped, dynamic real-world environments to fill the gap between the current proof-of-concept and deployable intelligent hardware.

References

[1] Зенкевич С.Л. Управление роботами. – М.: Изд-во MGTU, 2000. 456 с.

[2] Aliev A.A. Sun'iy intellekt asosida aqlli boshqaruv algoritmlari. – Toshkent: Fan, 2020.

[3] Робототехнические системы и комплексы: учебное пособие (Х.Н.Назаров.;TGTU, - Т.: 2004, 101 стр.

[4] M. Ben-Ari. Principles of Concurrent and Distributed Programming. – Addison-Wesley, 2017.

[5] Basil Hwang. Microcontroller Applications in Automation. – Springer, 2020.

[6] N.P. Plekhanov. Arduino va robototexnika: darslik. – Moskva: DMK Press, 2018.

[7] Робототехника Ю.Г., Андреанов и др. – М.: Машиностроение, 1984. 348 с.

[8] Шахинпур Н. Курс робототехники.– М.: Мир, 1990.516с.

[9] Banzi M., Shiloh M. "Getting Started with Arduino", 3rd Edition, Maker Media, 2014.

[10] Хасанов П.Ф., Назаров Х.Н. Мобильные робототех нические системы. – Т.:ТашПИ, 1987. 98 с.

[11] Хасанов П.Ф., Киселёв О.Д. Адаптивные роботы и системы технического зрения. Учебное пособие. – Т.: ТашПИ, 1986. 96 с.

[12] Жан М. Рабаи, Ананта Чандракасан, Боривож Николич. Цифровые интегральные схемы. Методология проектирования = Digital Integrated Circuits. — 2-е изд. — М.: Вильямс, 2007. — ISBN 0-13-090996-3.

[13] Мохов А.Д. Разработка математического и программного обеспечения систем управления мобильными роботами произвольной структуры с избыточными связями, 2014.

Information about the author

Ochilova Sojida Rayimberdi kizi	Karshi State Technical University, Senior Lecturer of the Department of Software and Hardware Support of Computer Systems E-mail: sojida.ochilova28@gmail.com Tel.: +998880950999 https://orcid.org/0009-0003-5965-5933
Ochilov Muzaffar Rayimberdi-evich	University of Information Technologies and Management, Senior Lecturer of the Department of Information Technologies. E-mail: muzaffar_3848@mail.ru Tel.: +998906078050 https://orcid.org/0009-0003-5965-5933

CONTEXT / MUNDARIJA