

Educational Robot Task Virtual Model Transformation Into Real Environment

Adomas Paulauskas

Panevezys Juozas Balckikonis Gymnasium, Lithuania

Abstract: We propose to extend the learning of programming basics in secondary school, creating an educational environment based on using virtual modelling of educational robot task and model's transformation into real environment. The control principles of a real robot are developed using robot's behavior modelling results received from virtual system. The main contribution of the paper is providing an opportunity for students to explore and manage the complexity and scope of the robot task using various abstraction level models and modelling techniques.

Keywords: educational robotics, meta-program, robot control program, virtual modelling environment,

I. Introduction

The achievements of computer science (CS) rapidly penetrate into different areas of life, and it is necessary to change the education of CS in secondary school. The changes relate to the following challenges [1-4]:

1. Computer-based systems are widely used in transport, healthcare, pharmaceutical and other industries. Students that have received fundamental basics of CS in secondary school, will not to be only educated consumers of technologies, but also the innovators who can use modern technologies to improve the quality of their life.
2. In 21 century, many professions include not only professional, but also CS knowledge to adequately use modern technologies in a specialized domain.
3. Studying CS students develop computational thinking, creativity and problem solving techniques. All listed concepts and skills are applied in many contexts: from the sciences and engineering to business and humanities.

ACM proposes the model of CS education in secondary school, in which CS is defined as the science that studies computers and algorithmic processes, hardware and software design, the applications and impact on society. CS includes programming and programming languages, hardware and software design, computer networks, graphics, databases and information retrieving processes, security problems, logics, programming paradigms, transitions between different abstraction level, artificial intelligence, computer's application limitations and scope, and social aspects of security areas. On the other hand, CS is closely connected with the theoretical basis of robotics, computer vision, intellectual systems, and bioinformatics.

Teaching and learning of programming basics are a significant part in CS education. CS educators and researchers are faced with the pedagogical, cognitive, content and technological issues that are related to the development of the skills and abilities of computational thinking in programming learning. The listed reasons and challenges suggest that programming education in secondary schools should be updated using modern educational environments. In this paper, we propose to extend the learning of programming basics in secondary school, creating an educational environment based on using virtual modelling of educational robot task and model's transformation into real environment. The aim of this paper is to introduce the development of control principles of a real robot using robot's behavior modelling results received from virtual system. The main contribution of the paper is providing an opportunity for students to explore and manage the complexity and scope of the robot task using various abstraction level models and modelling techniques. The remaining part of the paper consists of the related work, the architecture of educational environment, a case study that illustrates the architecture implementation, improvement opportunities and the conclusion.

In real robot-based systems, robot control programs are developed applying online and offline programming methods [5]. Using online programming method the robot's position is recorded at each stage of the task. The sequence of recorded positions goes to end-effector. The main disadvantage of online method is that the quality of the robot control program (RCP) is highly dependent on operator preparedness, and it is difficult to make changes in RCP. This method is popular, because it is intuitive, does not require good programming skills, and it is cheap [5]. The offline programming method is more flexible and it provides more opportunities for the development of robotic systems with sensors. Stilman et al. [6] includes experimental paradigm based on the hybrid real-virtual world for analysis and testing of complex robotic subsystems. The principles of the systems of this type are as follows: 1) modelling of the environment; 2) automatic robot

calibration. The paper [7] presents the modelling of the virtual robot path in an unknown environment using the algorithms of the optimal path, but in this system does not provide the possibility to model environment with obstacles. Some researchers have joined the virtual robot simulation environments with real robotic systems. The result is the robot programming using augmented reality which in real systems is not yet applicable.

II. The architecture of educational environment

The architecture of educational environment is presented in Fig. 1. The environment consists of two interacting components: (1) educational LEGO robot and (2) computer with the software for the modelling, data transformation, meta-program development and robot control program generating, and robot programming environment.

Now, we present the main definitions of our environment.

Definition 1. The virtual modelling environment is an environment in which the digital model of the robot task is created, and the initial data for robot control program development are extracted.

Definition 2. The virtual robot task model is a digital model of the physical object (robot) that enables to extract data for robot control program development.

Definition 3. The data transformation program is a set of the rules that enables to transform virtual models and robot calibration data into the form that is relevant to the robot control program.

Definition 4. The meta-program is a generalized program that flexibly generates robot control programs.

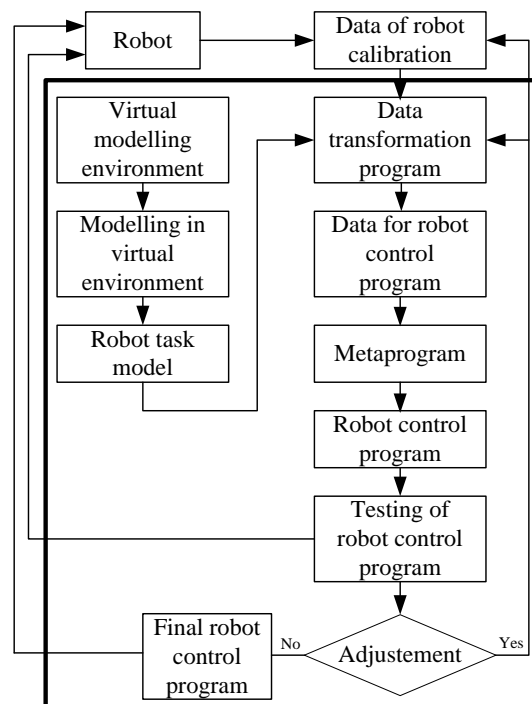


Fig. 1. The architecture of educational environment

III. Architecture implementation

Using the Unity game engine (<http://unity3d.com/>) has developed a virtual modelling environment that simulates the movement of the robot. In this environment the user can create a robot's movement model and data is stored for later processing (see Fig. 2).

The data transformation program has been created as a separate module of the environment and it transforms robot calibration and model data to the robot control program in the proper form. The data transformation program has been developed in C++ programming language using CodeBlocks environment (<http://www.codeblocks.org/>).

Using the heterogeneous meta-programming technology [8] has been developed a meta-program that enables to generate robot control programs flexibly by changing technical parameters of the robot. The generated instances of the robot control programs are transferred to the robot, and are tested. If the accuracy of the robot movement is not satisfied with the user needs, the robot calibration should be repeated and transformation rules should be revised.

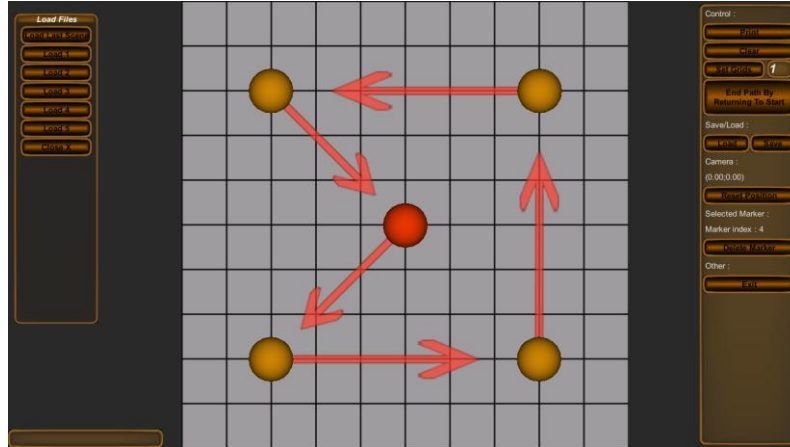


Fig. 2. A robot's movement model

In the development of the real application, we use the DRAWBOT [9] for several reasons: (1) the robot has been calibrated; we have revised and supplemented the calibration data; (2) the robot's movement is displayed clearly, because the robot draws its path.

Four transformation rules to describe the robot movement have been formulated as follows:

1. The time (t_1) of the robot movement straight forward:

$$t_1 = \frac{a_F \cdot k_{ch} \cdot 1000}{v_r \cdot k_1}, \quad (1)$$

there, a_F – the distance travelled by the robot forward, v_r – robot's speed, k_1 – coefficient obtained from the robot calibration data when the robot moves straight forward, k_{ch} – coefficient which modifies the model's space into real space.

2. The time (t_2) of the robot movement straight backward:

$$t_2 = \frac{a_B \cdot k_{ch} \cdot 1000}{v_r \cdot k_2}, \quad (2)$$

there, a_B – the distance travelled by the robot backward, v_r – robot's speed, k_2 – coefficient obtained from the robot calibration data when the robot moves straight backward, k_{ch} – coefficient which modifies the model's space into real space.

3. The time (t_3) of the rotational movement of the robot in clockwise direction:

$$t_3 = \frac{k_+ \cdot 1000}{v_r \cdot k_3}, \quad (3)$$

there, k_+ – the rotation angle, v_r – robot's speed, k_3 – coefficient obtained from the robot calibration data when the robot rotates in clockwise direction.

4. The time (t_4) of the rotational movement of the robot in anticlockwise direction:

$$t_4 = \frac{k_- \cdot 1000}{v_r \cdot k_4}, \quad (4)$$

there, k_- – the rotation angle, v_r – robot's speed, k_4 – coefficient obtained from the robot calibration data when the robot rotates in anticlockwise direction. The coefficients k_1, k_2, k_3, k_4 are obtained from the drawbot's calibration data (see Fig. 3 and Fig. 4) using data transformation program and transferred to meta-program calculated in advance. We use well known partial evaluation principle which is widely used in program optimization and specialization.

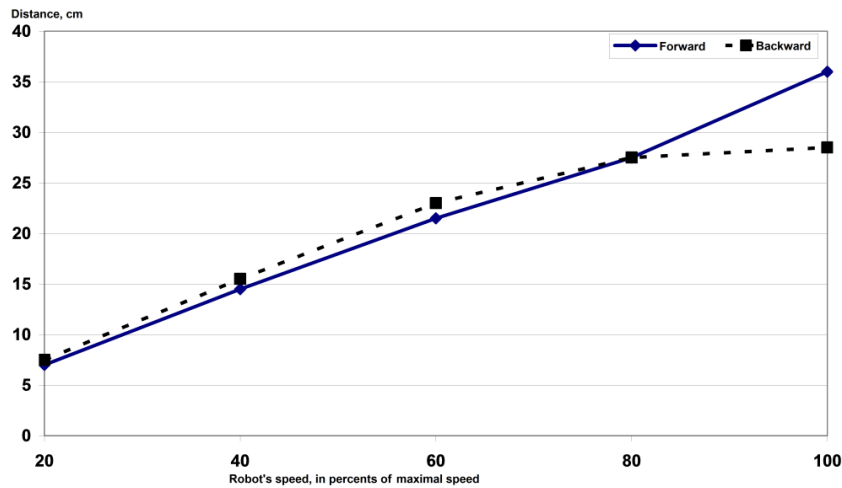


Fig. 3. A distance – speed dependency (movement time $t = 1000$ ms)

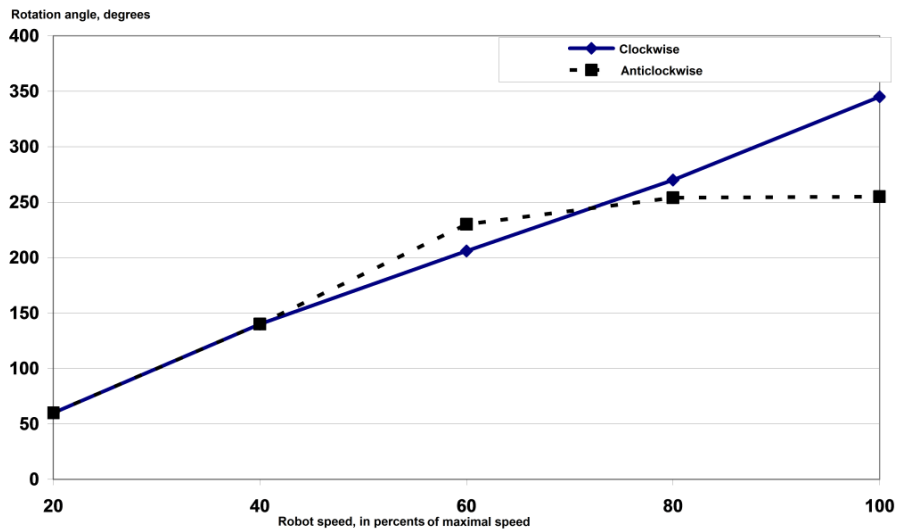


Fig. 4. A robot's rotation angle – speed dependency (movement time $t = 1000$ ms)

IV. Model and real application: experiments and comparison of the results

The path of the drawbot has been modelled (see Fig. 5). It consists of two parts: 1-2 and 2-3. After creation of the model the coordinates of each point are stored in a text file and transferred into data transformation program.

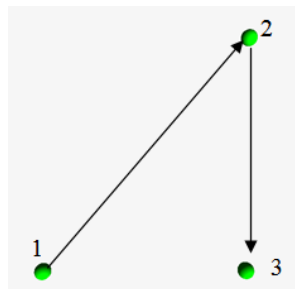


Fig. 5. Drawbot movement's model

We applied data transformation to the drawbot calibration and model data, and the robot control program was generated using transformed data. The obtained result is shown in Fig. 6.

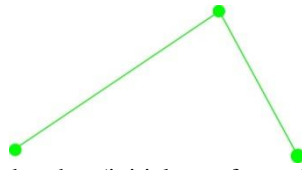


Fig. 6. The real path of the drawbot (initial transformation rules have been applied)

The result after correction of transformation rules is presented in Fig. 7.

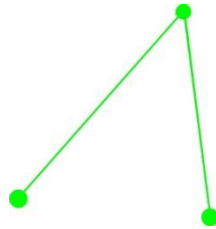


Fig. 7. The real path of the drawbot (corrected transformation rules have been applied)

The measurement results of distances among points 1-2 and 2-3 and values of the angle 2 are presented in Table 1.

Table 1. The comparison of model and real path of the drawbot

Parameter	Model	Real path: initial transformation rules	Real path: corrected transformation rules
Distance 1-2	8.52	64	70
Distance 2-3	6.27	42	58
Angle 2, radians	0.75	1.48	0.87

We performed the experiments modelling the movement of the drawbot's in square, rectangular, equilateral triangle shapes. The speed of the drawbot is 50 % of the maximal speed. The results of the measurements are shown in Fig. 8.

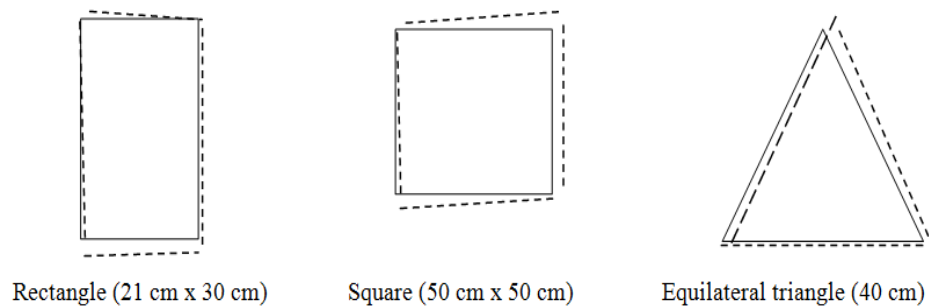
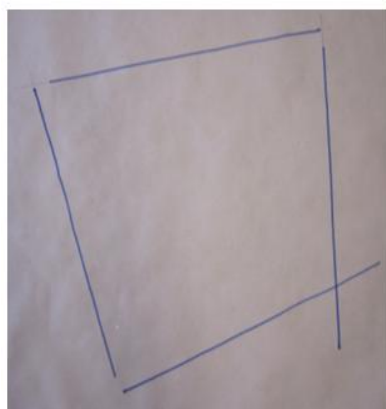
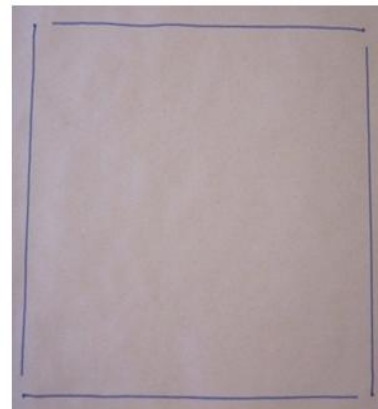


Fig. 8. Model (the solid line) and real path of the drawbot (the dotted line)



Robot's path without corrections



Robot's path with corrections

Fig. 9. Path of the drawbot (square 50 cm x 50 cm)

If the speed of the drawbot is maximal, the movement path is inaccurate (see Fig. 10 c)). After correction of coefficients the path of the robot is more similar to model, but due to the inertia differs from them.

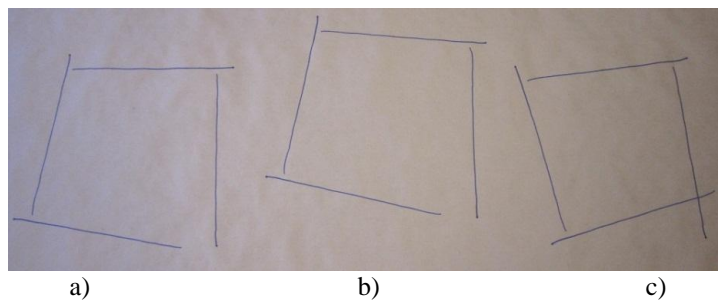


Fig. 10. Path of the drawbot (square 50 cm x 50 cm, maximal possible speed): a) and b) – with corrections of the coefficients, c) – without corrections of the coefficients

V. Robot control program’s development

We have developed the drawbot control programs using heterogeneous meta-programming. Meta-programming is the high level paradigm, when the effective development of programs is ensured by creating generalized programs that are called meta-programs (see *Definition 4*). Heterogeneous meta-programming use at least two programming languages. A meta-language is used to describe a generalization of an algorithm; a target language describes a functionality of the domain [8]. We have chosen C++ language as the meta-language and RobotC as target language. RobotC is used for development of the drawbot control programs. The model of the meta-program is shown in Fig. 11.

The transformed data those are suitable for robot control programs development
Meta-interface consists of metaparameters and their values
Meta-body consists of the set of the functions of metalanguage and the algorithm of the drawbot’s movement expressed in the target language

Fig. 11. A model of a meta-program

VI. Conclusion

The created system can be successfully used to simulate the path of the robot according to the user needs, when the path of the robot is without obstacles. Developed program of the adjustment of robot’s calibration coefficients allows the user flexibly revise the real path of the robot. The system can be used as an experimental environment for solving optimal path search tasks. The main improvements of the developed system are as follows: (1) the functionality of the system will be increased by adding different sensors; (2) the possibility for the user to adjust calibration coefficients remotely.

References

- [1]. Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., Trees, F.: Ensuring exemplary teaching in an essential discipline: Addressing the crisis in computer science teacher certification. *Final Report of the CSTA Teacher Certification Task Force. ACM*, (2008)
- [2]. Fletcher, G.H., Lu, J.J.: Education Human computing skills: rethinking the K-12 experience. *Communications of the ACM* 52 2, 23-25 (2009)
- [3]. Hazzan, O., Lapidot, T., Ragonis, N.: Guide to teaching computer science: an activity-based approach. Springer, (2015)
- [4]. Barr, V., Stephenson, C.: Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads* 2 1, 48-54 (2011)
- [5]. Pan, Z., Polden, J., Larkin, N., Van Duin, S., Norrish, J.: Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing* 28 2, 87-94 (2012)
- [6]. Stilman, M., Michel, P., Chestnutt, J., Nishiwaki, K., Kagami, S., Kuffner, J.: Augmented reality for robot development and experimentation. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-55 2 3*, (2005)
- [7]. Ong, S.-K., Chong, J., Nee, A.Y.: Methodologies for immersive robot programming in an augmented reality environment. In: Proceedings of the 4th international conference on computer graphics and interactive techniques in Australasia and Southeast Asia, ACM, pp. 237-244 (2006)
- [8]. Štuikys, V., Damaševičius, R.: Meta-programming and model-driven meta-program development: principles, processes and techniques. Springer Science & Business Media, (2012)
- [9]. Burbaitė, R., Štuikys, V., Marcinkevičius, R.: The LEGO NXT robot-based e-learning environment to teach computer science topics. *Elektronika ir Elektrotechnika* 18 9, 113-116 (2012)