

Robotics Education for Children at Secondary School Level and Above

Anastasia Stone¹ and Ildar Farkhatdinov²

¹ London School of Mathematics and Programming, London, UK

² School of Electronic Engineering and Computer Science,
Queen Mary University of London, London, UK
`i.farkhatdinov@qmul.ac.uk`

Abstract. The present work describes the experience of teaching robotics for children at secondary school level. A set of exercises was designed and evaluated. The exercises were designed for teaching robotic systems, basic programming and control concepts, and the tasks included learning important mathematical and physical science definitions. The proposed robotics exercises were introduced to the curriculum of the London School of Mathematics and Programming and preliminary teaching and learning outcomes showed that the majority of the proposed robotics exercises were attractive to children independently of the tasks' difficulty.

Keywords: robotics education · children · mobile robot · programming

1 Introduction

The London School of Mathematics and Programming's (LSMP) goal is to develop problem solving skills and cultivate interest to science and engineering among children of ages 7-16. In 2016, LSMP has introduced robotics classes to the school's curriculum designed for secondary school level students. Teaching robotics was proven to be successful and complementary to classes in mathematics and computer programming. In the present paper the author's would like to share their experience and observations on introducing robotics to children education through providing examples on how teaching autonomous robotics contributes to learning science disciplines.

Nowadays, robotics classes, clubs and societies are typical in many secondary and high schools. The major reasons for rapid robotics education expansion are (1) decreasing costs of robotics teaching kits, (2) excitement of learning through interactive robotic systems and (3) general growth of interest to robotics and artificial intelligence in the society. Robotics education for children has been widely introduced and various teaching methodologies have been investigated [1–4, 9]. In the majority of the cases wheeled mobile robots were used in teaching [6–8, 10, 11], but less popular applications included teaching humanoid robotics for children [5], robotic manipulators [12] and aerial robotics [13]. In this work TRIK robotics mobile platform [14, 16] was used to teach robotics to children. In contrast with the previous works we specifically address the challenges of formulating the robotics exercises for children. The goal of this work is to systematically

present and evaluate a set of robotics problem for children which could be used to develop general analytical thinking, as well as more specific engineering and mathematical skills.

2 Educational robotic toolkit

The TRIK robotic set was used during robotics classes [14, 16]. The set is composed of one embedded controller, various sensors and actuators, one battery with a charger and a system of *Meccano*-like assembling components. The TRIK set enables assembling different type of actuated mechanisms with controlled behaviour.

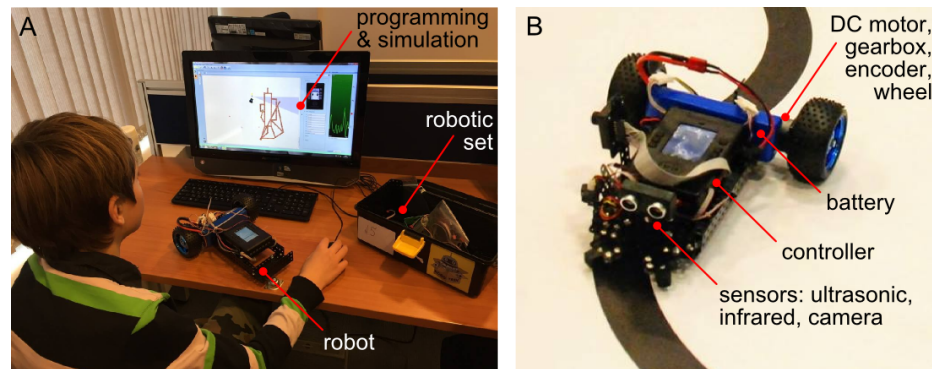


Fig. 1. A: programming the TRIK mobile robotic platform during the class; **B:** TRIK mobile robotic systems performing a line tracking task and its major components.

The behaviour of the robot is programmed with the help of a dedicated development environment [15]. The behaviour of the robot can be implemented with visual flow charts, as well as with JavaScript code. Designed behaviours can be first tested with built-in mobile robot software emulator. The emulated robot environment can be adjusted to have various type of obstacles (rigid immobile bodies) and guidance lines (visual markers on the ground). To test the programme with the physical robot the flow chart is translated into code and then uploaded to the robot controller via wireless network (WiFi).

The robot control diagrams are composed of interfacing and logical blocks. The primary interfacing blocks are used for applying voltage to the DC motors, stopping the DC motors, enabling the robot's sensors. Major logical blocks are used for control flow management and calculations. They are `if-then-else` statement, loop iterations and `goto` arrows, delay (wait) block and functional blocks to define variables and equations.

During the classes taught at LSMP the students assembled a mobile robot made of TRIK components with differential drive kinematics, as shown in Fig. 1.

The differential kinematics of the robot is shown in Fig. 2. The robots kinematics is described as

$$V = \frac{r}{2}(\omega_L + \omega_R), \quad \Omega = \frac{r}{L}(\omega_R - \omega_L) \quad (1)$$

with the speed of the robot V and the robot's angular velocity Ω ; the wheel's radius r ; the distance between the wheels L ; ω_L and ω_R the angular velocity of the left and right wheels, respectively. By varying the angular velocities of the two wheels, we can vary the trajectories that the robot takes. Therefore, to define the robots trajectory (translation and rotation in plane) angular velocity of the wheels needs to be controlled which was achieved by changing the voltage applied to the corresponding DC-motors:

$$\omega_{L/R} = kU_{L/R} \quad (2)$$

with the voltage applied to left/right DC motors $U_{L/R}$ and constant scalar k assuming linear motor dynamics.

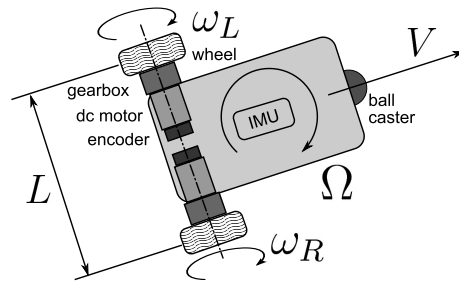


Fig. 2. Differential drive kinematics of the robot.

3 Teaching methodology

3.1 Students

Robotics classes at LSMP were organised in 10×1.5 h weekly sessions with a 2 weeks break in the middle for the school vacation. The age of the students varied from 9 to 14. In 2016, 9 students have attended the robotics classes. Majority of the students were not exposed to robotics, while some of the students had experience in using robotics sets from Lego. Most of the students have attended computer programming course at LSMP (Scratch and basics of Python) prior to attending the robotics classes.

3.2 Robotics exercises

In this subsection we describe typical exercises which were used during the classes.

E1. Motion and timing. The goal of this exercise was to program the robot to move around a given rectangular obstacle and return to initial position as shown in Fig. 3(E1). The rectangular obstacle was defined by each student in the beginning of the class. There are two ways students can approach the problem: 1) activating and deactivating the motors' power for a given time, and 2) activating the motors and stopping them when the wheel's encoders reached the required values. In both cases, the students had to define required motor activation timing or encoder counting using trial and error method. However, the students were explained that they can calculate the required final encoder values if the wheel's geometry and gearbox ratio is known. Students also learn how to steer the robot by changing the control inputs (voltage) applied to the wheel's actuators. At this stage they were taught the differential drive kinematics which is typical for many wheeled robotics systems. Other topics explained during the class include work principles of DC actuators, optical encoders, and kinematic relations between speed, distance and time. Once the exercise is complete, the students were asked to check the defined motor power, timing (or encoder values) for turning parts of their flow charts. In an ideal case, the turning parameters should be same, as the code should contain of four identical corner turns. The students were asked to find the common turn parameters, if this was not the case. As a final stage of the exercise, the students were taught the concepts of functional programming: they were required to create a turning function (subprogram block) and replace all the turning commands with this new block.

E2. Simple collision avoidance. The task was to drive the robot towards an obstacle (a wall), stop at a given distance, which was then followed by a right turn (approx. 90° , using the subprogram block from *E1*) and motion towards the next wall if any (Fig. 3(E2)). In this exercise students learned how to use the ultrasonic sensors and `if-then` statement to develop the collision avoidance behaviour.

E3. Collision avoidance and navigating. This exercise is an extension of *E2*. Students were asked to draw an arbitrary labyrinth in the emulation software. The robot should find the exit without collisions with the wall as shown in example in Fig. 3(E3). Loops were introduced in this exercise to enable moving forward and turning left or right if there was a wall in front of the robot. An extension to this exercise was implementing alternating left/right turn, hence providing more flexible robot behaviour. In this task students learned how to program loops and `if-then-else` statements to implement simple navigation algorithms. They also learned how to use variables, implement loop iteration counters and use them to adjust the robots behaviour (i.e. checking if the loop counter's even or odd to decide which turn to take).

E4. Wall following and line following This exercise is typical for robotics education and competitions. An example is shown in Fig. 3(E4). In comparison to *E3* the behaviour of the robot was defined through a system of control equations,

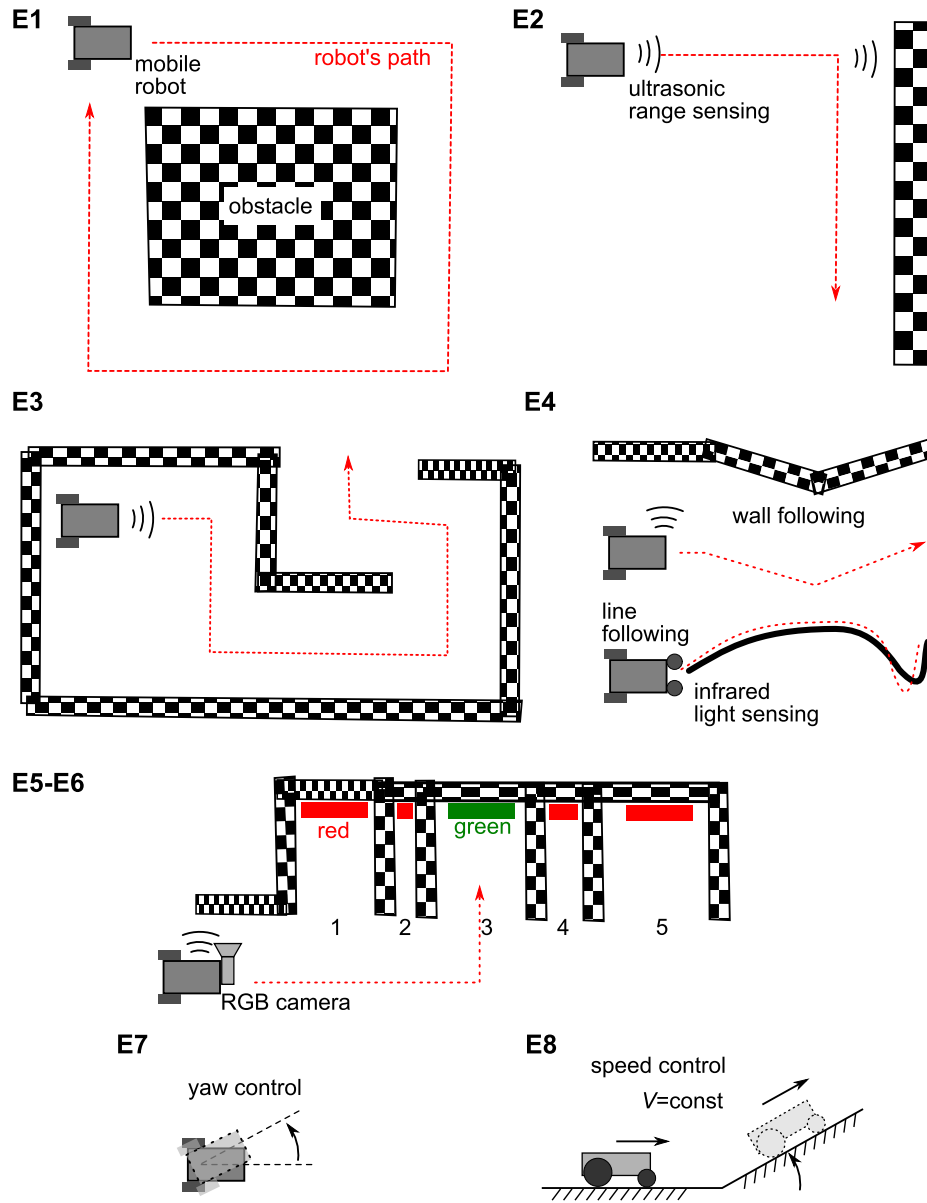


Fig. 3. Schematic diagrams of the exercises.

rather than by using **if-then-else** statements. The following equations were used for the wall following behaviour

$$U_L = k_w(d - \tilde{d}) + U_o, \quad U_R = -k_w(d - \tilde{d}) + U_o \quad (3)$$

with controller gain k_w ; reference distance to the wall d ; measured distance to the wall \tilde{d} and voltage defining the speed of the robot U_o . In the line following algorithm two light sensors (attached to the front corners of the robot's frame) are used to track the line. The following control law was used

$$U_L = k_l(s_L - s_R) + U_o, \quad U_R = -k_l(s_L - s_R) + U_o, \quad (4)$$

with the controller gain k_l , and left and right light intensity sensor measurements s_L and s_R , respectively. In this exercise students learned how to implement control laws for specific tracking tasks and how to use the light sensors. Importantly, the task introduced linear control concepts, as in the both tracking tasks proportional error based feedback control was used.

E5. Parking The task for this exercise was to park the robot as shown in Fig. 3(E5-E6). In the first part of the exercise the robot should park in the n -th slot of the parking space, where $n=1, 2, 3$, etc. In the second part, the robot should park only if the slot was large enough for parking. Ultrasonic sensor installed on the side of the robot was used to count the parking slots and estimate their size. Here students learned how to use ultrasonic sensor measurements, timer and robot's speed to estimate distances (width of parking slots) and how to use loop iteration counters triggered by ultrasonic sensor measurements to count the parking slots.

E6. Parking with vision based control This is an extension of *E5* parking task with additional RGB-camera based control. Parking was initiated only if a green coloured parking sign was presented to the robot as shown in Fig. 3(E5-E6). In this exercise students learned about RGB colour coding and principles of video camera operation for robot vision.

E7. Angular orientation control In this exercise the robot should orient itself to a predefined angle in horizontal plane as shown in Fig. 3(E7). This was implemented with the help of yaw estimation using built-in gyrometer and proportional feedback controller. First, students learned how to use gyrometer to measure angular velocity of the robot, which was followed by implementing one step numerical integration to calculate the angular orientation (yaw) of the robot. A discrete-time linear proportional control law was introduced to turn the robot to desired location:

$$\begin{aligned} \varphi_i &= \tilde{\Omega}T + \varphi_{i-1}, \\ e_i &= \varphi_{ref} - \varphi_i, \\ U_{L,i} &= k_\varphi e_i, \\ U_{R,i} &= -k_\varphi e_i, \end{aligned} \quad (5)$$

with the current loop iteration i ; the sampling time T ; calculated yaw angle of the robot φ ; measured angular velocity of the robot in the horizontal plane $\tilde{\Omega}$ and the controller gain k_φ . The derivative component was not introduced to the controller to keep the exercise relatively simple. It was also tested experimentally, that friction and damping in the robot's mechanism was sufficient to keep the proportional feedback controller stable.

E8. Speed control In this exercise students learned how to maintain the speed of the robot, so that the robot could move on inclined surfaces and automatically adjust the voltage applied to the motors. A simple example is shown in Fig. 3(E8). Students learned how to compute the velocity of the robot from the onboard accelerometer measurements using one-step time integration. Based on the calculated linear velocity of the robot the feedback control was implemented as follows:

$$\begin{aligned} V_i &= \tilde{a}T + V_{i-1}, \\ e_i &= V_{ref} - V_i, \\ U_{L,i} &= U_{R,i} = k_V e_i, \end{aligned} \tag{6}$$

with the current loop iteration number i ; the sampling time T ; calculated speed of the robot V ; measured forward acceleration of the robot \tilde{a} and the controller gain k_V . The designed controller was tested directly on the robotic platform because the simulator supports only the movements in the horizontal plane. In the exercise the students compared two control options for the robot driving on an inclined surface: constant voltage applied to the motors (conventional approach) and feedback controlled voltage based on velocity estimation from the accelerometer. Using acceleration in feedback control loop enabled automatic adjustment of the speed, as the calculated speed of the robot decreased when the robot climbed the inclined surfaces. In the exercise *E8*, as well as in the exercise *E7* the students were explained the advantages of inertial measurements with respect to wheel's encoder based control, which was less efficient due to slippage and friction of the mechanisms.

4 Results

The proposed exercises have been taught during one semester at LSMP. For each exercise difficulty, time required for completion and interest of students were evaluated. Difficulty and interest in the task were evaluated subjectively by the class teacher based on the students' response. Three levels were used to define the complexity and the interest: low (L), medium (M), high (H). The results are summarised in Table 1. Additionally, Table 1 summarises the skills and knowledge acquired by the students during their classes at LSMP.

Table 1. Summary of the exercises evaluation

Task	Difficulty	Interest	Completion time, h	Skills and knowledge
E1	L	M	1-2	Hardware: robot's structure, electric motors, encoders. Software: timing, command sequences, subprogrammes. Maths/physics: wheel kinematics, speed-distance-time relations.
E2	L	L	0.5-1	Hardware: ultrasonic sensor. Software: <code>if-then-else</code> statement Maths/physics: principle of ultrasonic sensor.
E3	M/H	H	2	Software: control loops, loop counters, <code>if</code> statements in the loop. Maths/physics: checking even/odd numbers.
E4	M	H	1-2	Hardware: light sensor. Software: variables, control laws and equation blocks. Maths/physics: principles of light detection, equations.
E5	H	H	2-4	Software: counters based on sensor's triggers Maths/physics: kinematics of robot and range measurements to calculate distances.
E6	L	H	1-2	Hardware: RGB-camera. Maths/physics: principles of colour coding.
E7	H	L	2-3	Hardware: gyrometer. Software: control error, feedback control. Maths/physics: definition of angular velocity, one-step integration.
E8	H	M	2-3	Hardware: accelerometer. Software: control error, feedback control. Maths/physics: definition of speed, acceleration, one-step integration, trigonometry to calculate tilting.

Overall feedback from the students was positive and all students were interested in the exercises. The exercises which drew the most attention from the students were *E3*, *E4*, *E5* and *E6*. Navigation exercises *E3* and *E4* caught higher interest because the students were able to implement relatively complex behaviour of the robots and test them in various environments such as in emulated or real maze in comparison to the tasks in *E1* and *E2* when simple robot's activity was observed. Similarly, in the parking exercise *E5*, a real world application was taken as the task model, which made the overall exercise more attractive to the students. Lastly, vision based parking task *E6* was popular as well due to the employment of an RGB-camera and a simple computer vision technique to modify the robot's parking controller. The orientation and speed control exercises were less attractive to the students, as they did not involve complex robot movements. The interest level for the orientation control exercise *E7* was low, while it was slightly higher for the speed control task. First reason for the low interest in these tasks was relatively high complexity, as both of them involved operations with equations and elements of numerical integration. However, in our opinion, one-step numerical integration can be well-explained if simple examples are introduced to the students. One of the major difficulties at this level was understanding how storage of previous and current values of the robot's states (speed, acceleration) can be implemented within the control loop. Nevertheless, all students were able to complete these tasks. Importantly, they were able to understand the application of feedback control as they were asked to compare the robot's performance with open loop and feedback loop control. For instance, for the exercise *E8*, the students compared how the robot could climb on inclined surfaces when a constant voltage is supplied to the motors, and when the speed-based feedback control was implemented. In terms of difficulty, only the first two exercises were assessed as simple. Exercises with the highest interest were characterised by medium and high complexity, except for *E6* when the RGB-camera was used because only minor modifications to the control algorithm from *E5* were required.

Informal communication with the students and their parents indicated the significant increase of children interest in robotics in particular and in science and technology in general. This result was achieved by careful planning of the robotics exercises and systematic inclusion of additional teaching material enabling the students to broaden and deepen their knowledge in mathematics and physical sciences, as well. The results reported in present work are promising, however they are initial and based on the observations from the first semester of robotics education at LSMP. In future, more detailed analysis of the learning outcomes over a longer period of time will be required.

Acknowledgments. The authors would like to thank E. Galuzo, I. Kirilenko, I. Shirokolobov, R. Luchin and D. Deliya for fruitful discussions and technical assistance.

References

1. Johnson, J., 2003. Children, robotics, and education. *Artificial Life and Robotics*, 7(1-2), pp.16-21.
2. Petre, M. and Price, B., 2004. Using robotics to motivate back door learning. *Education and information technologies*, 9(2), pp.147-158.
3. Ruiz-del-Solar, J. and Aviles, R., 2004. Robotics courses for children as a motivation tool: the Chilean experience. *IEEE T. on Education*, 47(4), pp.474-480.
4. Benitti, F.B.V., 2012. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), pp.978-988.
5. Robins, B., Dautenhahn, K., Te Boekhorst, R. and Billard, A., 2005. Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills?. *Universal Access in the Information Society*, 4(2), pp.105-120.
6. Jojoa, E.M.J., Bravo, E.C. and Cortes, E.B.B., 2010. Tool for experimenting with concepts of mobile robotics as applied to children's education. *IEEE T. on Education*, 53(1), pp.88-95.
7. Nourbakhsh, I.R., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., Richards, S. and Wilkinson, K., 2005. The robotic autonomy mobile robotics course: Robot design, curriculum design and educational assessment. *Autonomous Robots*, 18(1), pp.103-127.
8. Mataric, M.J., Koenig, N.P. and Feil-Seifer, D., 2007. Materials for Enabling Hands-On Robotics and STEM Education. In *AAAI spring symposium: Semantic scientific knowledge integration* (pp. 99-102).
9. Mubin, O., Stevens, C.J., Shahid, S., Al Mahmud, A. and Dong, J.J., 2013. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1, pp.209-0015.
10. Kulich, M., Chudoba, J., Kosnar, K., Krajnik, T., Faigl, J. and Preucil, L., 2013. SyRoTek Distance teaching of mobile robotics. *IEEE T. on Education*, 56(1), pp.18-23.
11. Magnenat, S., Shin, J., Riedo, F., Siegwart, R. and Ben-Ari, M., 2014. Teaching a core CS concept through robotics. In *Proc. of the 2014 conference on Innovation & technology in computer science education* (pp. 315-320). ACM.
12. Cook, A.M., Bentz, B., Harbottle, N., Lynch, C. and Miller, B., 2005. School-based use of a robotic arm system by children with disabilities. *IEEE T. on Neural Systems and Rehabilitation Engineering*, 13(4), pp.452-460.
13. Gaponov, I. and Razinkova, A., 2012. Quadcopter design and implementation as a multidisciplinary engineering course. In *Proc. of IEEE International Conference on Teaching, assessment and learning for engineering*, (pp. H2B-16).
14. Terekhov, A., Litvinov, Y. and Bryksin, T., 2013. QReal: Robots an environment for teaching computer science and robotics in schools. In *Proc. of the 9th Central & Eastern European Software Engineering Conference in Russia* (p. 10). ACM.
15. Sedov, B., Pakharev, S., Syschikov, A. and Ivanovva, V., 2015. Domain-specific approach to software development for microcontrollers. In *17th Conference of Open Innovations Association (FRUCT)*, (pp. 179-185), IEEE.
16. Terekhov, A.N., Luchin, R.M. and Filippov, S.A., 2012. Educational cybernetical construction set for schools and universities. *IFAC Proceedings Volumes*, 45(11), pp.430-435.