

Mathematics with Robotnačka and Imagine Logo

Pavel Petrovič

Department of Applied Informatics

Faculty of Mathematics, Physics, and Informatics, Comenius University, Bratislava

ppetrovic@acm.org

Abstract

Robotnačka, developed by a robotics community in Bratislava, is a mobile robot implementing the Logo turtle metaphor in hardware. It performs the movements of the Imagine Logo turtle while drawing on whiteboard or paper surface with better precision than any other hardware turtles known to us. The robot can also perform circular movements, which are beyond the Logo turtle metaphor. Robot can easily be shared by a group of students in a networking classroom. Remote control can take form of Java applets, applications or Imagine programs, thus creating a virtual tele-operated robotics laboratory open to any Internet user. Visual feedback from a mounted top-view video camera or from a mobile wireless camera mounted on the robot extends the system with percepts from the turtle's environment. Bottom light sensors indicate floor colour or edge of table. A gripper with two degrees of freedom can optionally be attached to the robot for simple object transport and manipulation. We have implemented two open-source software add-ons for Imagine Logo bringing in new commands for image processing and robot control: cv4logo.ocx (Computer vision for Logo) and robot.ocx. They also introduce new networking functionality allowing Imagine users to create simple video conferences. The project is now brought to classrooms with a set of elementary- and secondary- school projects on mathematics, physics, and algorithms, accompanied with easy to follow teachers' and students' exercise books. The platform is ready and available for didactic research and use.

Keywords

Robotnačka, robotics, Imagine, educational tools

1. Introduction

Many years of practice with Logo programming language in classroom education show the great potential of the turtle metaphor. For the first, it is highly motivating, flexible, and universal, for the second, it stimulates student imagination, constructive, and analytical thinking. Applications start from algorithms and informatics, continue through mathematics, and extend to practically all school subjects. Our belief is that the potential of the turtle metaphor does not end in the software implementation, but extends further to hardware implementations, robot turtles, bringing in a new level of educational use of Logo.

A robotic turtle should perform as the turtle on the Logo screen: move and draw. Other requirements include the possibility of control from the Logo environment, simple use and maintenance, robustness, safety, durability, and interactivity. Extendibility of the robot turtle would allow richer applications based on the turtle metaphor, but even perhaps beyond it. People in the robotics community of academic institutions, commercial actors, and volunteers in Bratislava [7] adopted these assumptions and developed several prototypes of robot turtle, Robotnačka, see figure 1.



Figure 1. Robotnačka version 1 (left), and robotnačka version 2 (right).

2. Robotnačka

Robotnačka is a circular mobile robot with two stepper-motor propelled wheels, and a movable pen. It is autonomous and controlled by a firmware C-program stored in a flash-type memory. It runs on a simple processor Atmel AT89S8252, see table 1 for further technical details. The robot is communicating with the PC over Bluetooth serial port using simple bi-directional communication protocol; multiple robots can be controlled simultaneously without interference. The rechargeable battery lasts for several hours of continuous operation and thus it is suitable for use in the schools. Robotnačka features 6 bottom light sensors (two are in front of the wheels, two are behind the wheels, one is in the very front, and one is in the very back of the robot). The sensors detect the edge of the drawing area or black marks placed over the white drawing area, see figure 2 left. An intelligent gripper with two degrees of freedom controlled by a separate processor and communicating with the robot over internal communication link can be attached to the robot for manipulating and transporting objects, see figure 2 right. Robotnačka can draw with standard pen on paper surfaces, or with an erasable pen on ceramic whiteboard surface. In our experiments and projects, we have used an Axis network camera for top-view guided robot navigation and scene analysis. A wireless version of the same camera can be mounted on the robot and powered from its battery. Other cameras (such as standard USB web camera) can be easily used in Logo projects with Robotnačka too. Standard or advanced joysticks can be used to control the robot (and have been positively evaluated by a painter artist, as well as children, see figure 4.

Table 1. Robotnačka technical specifications.

Base type	Circular with two propelled wheels and one supporting rear wheel
Wheels drive	Stepper motors MICROCON SL17
Pen movements	Solenoid with magnet
Power	6V 3.3Ah maintenance-free lead accumulator
Wheel diameter	50 mm
Wheel distance	180 mm
Base height	68 mm
Base diameter	210 mm
Weight	ca. 2kg
Steps to turn base 360°	2880 steps
Controller	8-bit microcomputer AT89S8252
Wireless communication	INFINEON Bluetooth module communicating at 19200 bps
Indication and control	Sound for operating state, LEDs for recharge, power, communication, and gripper states, on-off switch and reset button

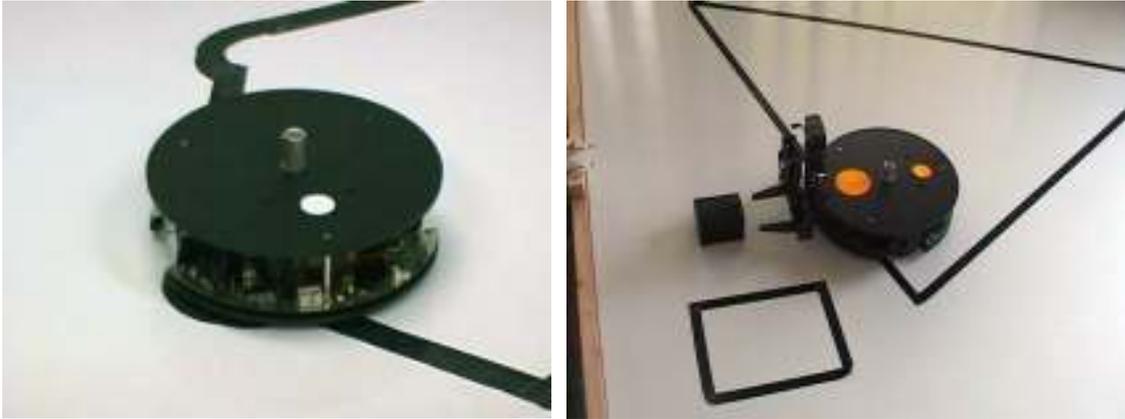


Figure 2.. Robotnačka following line (left), and Robotnačka with a gripper approaching an object (right).

3. Software functionality

The robot can be controlled directly from Imagine Logo [1] environment. Two add-on ActiveX components (robot.ocx and cv4logo.ocx) are loaded automatically, and provide a multitude of commands for robot control and image processing, see table 2, and table 3 for short overview.

Table 2.. Some commands defined by robot.ocx.

Robot position commands move the robot straight or along a circle, or make it rotate	fd, bk, lt, rt, fdx, bkg, ltx, rtx
Robot speed commands modify the current speed of both wheels	fdspeed, bksped, ltspeed, rtspeed
Condition command defines sensory state in which the movements are suppressed	Condition, satisfied
Robot pen commands	pu, pd
Information retrieval commands	getactivity, getsensors, isalive, getvoltage, pos
Networking commands	server, user, password, exclusive, message, logoff
Robot calibration commands	Drawcalib, calibrate
Trajectory recording and replay commands	logfile, replay
Gripper commands	open, close, up, down, upwards, downwards, tighten, loose, fix
Text drawing commands	drawtext, textsize



Figure 3.. Robotnačka (or several of them at once) controlled by a joystick, photos courtesy robotika.sk.

Table 3.. Some commands defined by cv4logo.ocx.

Image window source and control	show, hide, updatarate, getfps, clipview, framesize, mousepos, lastmouseclick, source
Storing frames to files, memory	saveframe, storeframe, forgetframe
Camera calibration commands	calibration, calibrate, calibrationsquaresize
Object localisation commands	newlocalisator, detectrobot, posandheading, relposchange, droplocalisator, steps2pos, pos2steps
Face, polygon, and pattern recognition	tracking, getfaces, trackingparameters, detectpoly, detectpolyparam
Networking commands	frameserver, listen, connect, accept, send, receive, receivevb, isconnected, close, download

Both components compile as stand-alone application or DLL, too. As a consequence, basic control of the robot from Delphi, C++, Visual Basic, or virtually any standard development environment is analogous to control from Imagine Logo. Components can be utilized by Java applications either using JNI or in case of robot component also by separate pure Java API. Detailed documentation of both open-source components is publicly available [7].

In addition to the components, we developed a prototype Imagine application for use in the schools. An exploratory environment integrates configuration, demonstrations, and tutorial, with educational projects in mathematics, physics and algorithms, which are described in the section 5.

4. Example: controlling robot by head movements

Assume a single web camera is connected to a USB port, and Robotnačka is controlled over Bluetooth serial port COM4. The following Logo program instructs the robot to move in the same direction as the head of a person looking into the camera. The face recognition in OpenCV library is implemented using the Haarcascade algorithm [6].

```

new "oleobject [comname robot.robotctrl.1 name r]
r'connect "com4 0
new "oleobject [comname cv4logo.cv4logoctrl.1 name cv]
cv'source "cam 0 cv'show
cv'tracking "on
make "leftline first (parse cv'framesize) / 3
make "rightline :leftline * 2
make "bottomline item 2 (parse cv'framesize) / 3
make "topline :bottomline * 2
print [Move your head left, right, up and down to steer the turtle]
forever [
  make "facecenter (apply "sum (first parse cv'getfaces)) / 4
  ifelse first :facecenter < :leftline [make "turning "right]
    [ifelse first :facecenter > :rightline [make "turning "left]
      [make "turning "none]
    ifelse item 2 :facecenter < :bottomline [make "move "back]

```

```

[ifelse item 2 :facecenter > :topline [make "move "forward]
  [make "move "none]
case :turning
  [left [case :move
        [forward [r'fdspeed 30 90]
          back [r'bkspeed 30 90]
          none [r'ltspeed 50 50]]]
  right [case :move
        [forward [r'fdspeed 90 30]
          back [r'bkspeed 30 90]
          none [r'rtspeed 50 50]]]
  none [case :move
        [forward [r'fdspeed 50 50]
          back [r'bkspeed 50 50]
          none [r'fdspeed 0 0]]]]]

```

5. Robotnačka as an educational tool

The technological progress, our robot attempting to be an example of, can be meaningful, only if the technologies improve our lives, and work. Our way of improving the lives of the students is increasing their chances for learning the curriculum material by making it more understandable, motivating, active, entertaining, closer to the real world, challenging, suitable for group project learning, and stimulating constructive and exploratory thinking. For this purpose, we have developed 10 one-lesson projects and curriculum materials that can be used by teachers and students without any technical knowledge.

5.1. Math projects

Circles. Even though the geometric precision of a turtle when drawing straight lines is limited by natural laws (a difference of 0,01 mm in the wheel diameter results in an error in the magnitude of centimeters when the robot draws only a square with the side length 1 m; imprecision can be further compensated by robot calibration, but imperfections have to be admitted), this does not apply to circles. Indeed, the turtle can draw quite precise circles. To many of us, it comes as a surprise that the result of the command “`r'ltspeed 30 70`”, which sets the speed of the left and right wheels to -30 and +70 respectively, results in a clean circle. In the circles projects, the students learn the circle properties, and in a sequence of exercises draw circles of arbitrary diameter with their robot turtle.

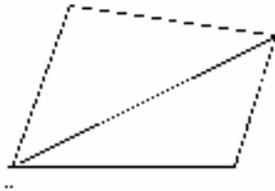
Triangles. Triangles are the simplest closed shapes a turtle can draw, but they provide a plethora of interesting challenges and exercises, where the students test their knowledge of triangles. From computing the angles and sides given various parameters (thus employing various theorems), constructing tangents and heights, lesson continues through drawing and understanding circumscribed and inscribed circles. Triangles can be constructed of coloured insulating tape on the whiteboard surface, if the camera system is installed, or alternately, defined on the student screen only.

Sets of points. Line is a half of a set of points that have a constant distance from another line, a circle is a set of points with a constant distance from a point, a parabola is... In this lesson,

students acquire scene using the vision system and construct various sets of points that share certain property, such as distance to a vertex, distance to multiple shapes, and difference of distances. Students work with high-level concepts, shapes, and points when constructing programs to navigate the drawing robot.

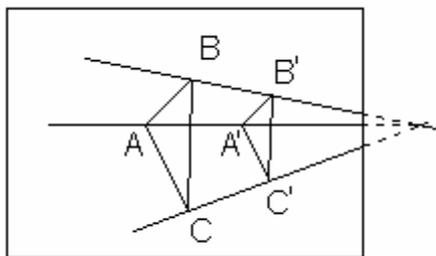
Constructive geometry. This lesson is based on a micro-world environment that defines a set of geometric construction operations. These differ from the standard constructive geometry with ruler and compass. The program can acquire the scene, label the points, divide line segments, determine the distances, and angles. The micro-world environment allows the teacher to define his or her own tasks, and to specify the set of available commands for each task separately. Therefore the students are required to find a creative solution given a very simple command set, instead of directly computing the target location, for instance.

Example 1: Construct a quadrangle, given the side $a=650$, internal angles $\alpha=67^\circ$, $\gamma=90^\circ$, $\delta=105^\circ$, and the diagonal $e=800$. The robot pen is down. Use the commands `lt`, `fd`, `defpoint`, and `fdupid` (`fd` until point is in certain distance):



```
fd e          lt 360 - (alpha + gamma + delta)
lt alpha      fdupid A, e
defpoint A    lt gamma
fd a          fd e
```

Example 2: Given is a point A and two non-parallel lines that intersect outside of the whiteboard. Draw a line that connects A with the intersection of the two lines. The robot pen is up, and it is located somewhere outside the lines, heading towards both of them, intersecting them sufficiently far from the area boundaries. Use the commands `fdul` (forward until line), `defpoint`, `fd`, `heading2pt`, `followline.lt`, `fduhtp` (`fd` until heading to point), `pd`, `moveto`.



```
fdul          defpoint Cx
defpoint B    setheading delta
beta = heading2pt A
fd 100        defpoint Bx
fdul          setheading beta
defpoint C    fduhtp Cx, gamma
gamma = heading2pt A
delta = heading2pt B
followline.lt 700
pd           moveto A
```

The micro-world environment provides more than 30 high-level predefined commands that can be selected for particular tasks. The teacher users are very welcome to define more operators. More examples and the commands reference can be found in the Logo application for Robotnačka.

5.2. Algorithmic projects

Convex hull. The camera system and `cv4logo` detects the list of points of all polygons placed on the whiteboard. Students are instructed to construct programs that draw convex hulls in

cooperation with teacher and using high-level constructs – from specific and simple cases towards a general case that concludes the exercise, see figure 4.

Depth-first search. A maze formed of black insulating tape attached to the whiteboard contains marked locations. Robot uses the sensors to read the marks encoded in binary notation (number and position of black stripes). The task for the students is to make the robot locate the target location (place for the robot parking, or unloading an object) by writing a simple Logo program that will negotiate the maze.

Minimum spanning tree. The vision system detects locations of cities depicted by circles placed over the whiteboard. Students learn about the problem of the minimum spanning tree construction and develop a Logo program for the robot that makes it draw the minimum spanning tree. Students experiment with and evaluate the heuristics for tree-drawing sequencing.

Polygon triangulation. In this advanced project, the students construct a single algorithm for arbitrary polygon triangulation. The task for the robot is to draw the triangulation of a polygon detected by the vision system. The exercise starts with simple convex polygons and extends to general non-convex polygons.

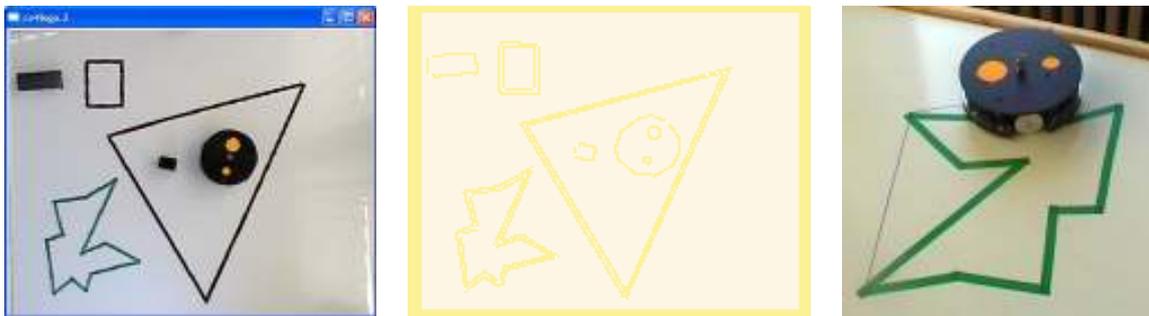


Figure 4. The scene as seen by top-view camera (left), the same scene recognized by cv4logo (middle) and drawing convex hull (right).

5.3. Physics projects

Collisions. A turtle robot moving at a controlled constant velocity meets an object in an elastic or inelastic collision. In this laboratory exercise the students learn about the collision physics, measure the velocities using the camera after they calibrate it, and compare their measurements with the computed prediction. At the end, they produce a polished lab report.

Force, power, friction, and work. The more slowly the robot is moving, the stronger force it can generate. For instance, it can push a heavier object, or extend a spring of a Newtonmeter (and the more battery power it consumes as well). This is the property of the stepper-motor drive. Students measure the velocity decline either by marks drawn by the robot, or using the top-mounted camera. In this lesson, they review their previous knowledge of force, power, and energy physics laws and apply it in the set of exercises. Using the computations based on measurements, they compute the friction coefficients for different materials, speeds, and pressures.

6. Conclusions and Future work

Despite the tremendous success of LEGO robotics platform, we believe there is a large gap between the technological advancements in the field of embedded devices, robotics, and

electronics, and the real teaching practice in the schools. The limits today are rather defined by our imagination than the technology itself. However, the teachers often limit themselves only to standardized PC equipment, or traditional old-fashioned tools, especially in mathematics. Technicians, teachers, and thinkers need to work tightly together and see how their implemented designs work in the classroom settings in order to integrate the feedback and shape the tools of the future school.

Robotnačka is a modular, interactive, robust mobile robot that relies on the proven Logo metaphor with large group of users. Its main areas of classroom applications include geometry, an often neglected, or avoided, however very useful, part of mathematics. Applications comprise projects in physics, and algorithms. The platform that includes an optional camera vision system and an intelligent gripper can be fully controlled from Logo environment. As an open-source project, it is ready for improvements, modifications, adjustments, and further projects contributed by the users of this educational tool.

Pilot set of ten projects implemented in Imagine Logo and accompanied by teachers' and students' instruction sheets (4 math, 4 algorithms, and 2 physics) is being evaluated in the schools at the time being. All materials and evaluations will be made available for review, feedback, and discussion at [7].

7. Acknowledgments

Robotnačka version 1 was developed by Albert Diosi, Oto Kužma, and Dušan Ďurina, Robotnačka version 2 was developed by Dušan Ďurina, Oto Kužma, Peter Lacko, and Pavel Petrovič. The intelligent gripper was developed by Ondrej Chlebo, Pavol Krasňanský, David Gustafík, Filip Tóth, and Dušan Ďurina. The company Microstep-MIS has supported the development process generously. We acknowledge the authors of Imagine Logo, especially Dr. Peter Tomcsányi, for the help with interfacing our software with Imagine Logo. Cv4Logo component is developed using open-source library OpenCV from Intel.

References

- [1.] Kalaš, I., Hrušecká, A.: The Great Big Imagine Logo Project book. Logotron, 2004.
- [2.] KORTENKAMP, D., BONASSO, R.P., Murphy R.: Artificial Intelligence and Mobile Robots. MIT Press, 1998, ISBN 0-262-61137-6.
- [3.] LANDRÉ, J.: Programming with Intel IPP and Intel OpenCV under GNU Linux, A Beginner's Tutorial, Université de Bourgogne, 2003.
- [4.] Murphy, R. Introduction to AI Robotics, MIT Press, 2000, ISBN 0-262-13383-0.
- [5.] SONKA, M., HLAVAC V., BOYLE R.: Image Processing, Analysis, and Machine Vision. Brooks-Cole Publishing Company, 1999. ISBN 0-534-95393
- [6.] Open Source Computer Vision Library, Reference Manual, INTEL 2001, order number A77028-004.
- [7.] Robotics community in Bratislava, <http://www.robotika.sk/>