

**ICPE 2017**  
**International Conference on Psychology and Education**

**INTEGRATION OF ROBOTICS SIMULATION PROGRAMS INTO  
EDUCATIONAL PROCESS IN KUBAN STATE UNIVERSITY**

Igor Ryadchikov (a), Semyon Sechenev (a), Alexander Svidlov (a), Sergey Sinita (a), Pavel Volkodav (a), Yury Mamelin (a), Kirill Popko (a) and Evgeny Nikulchev (b)\*

\*Corresponding author

(a) Kuban State University, 350040 Stavropolskay, 149, Krasnodar, Russia  
(b) Moscow Technological Institute, 119334 Leninskiy pr. 38A, Moscow, Russia

*Abstract*

The article describes the practical experience of using the Robocode computer simulator and models in Matlab Simulink Simmechanics in the study of artificial intelligence and robotics systems at the Kuban State University. In this paper, the problem of teaching students of higher educational institutions of robotics with the application of the project-based learning and using the simulation environment is considered. These environments allow, at the initial stage of training, students to replace expensive experiments with real hardware. The actual task when teaching such disciplines is the selection of practical tasks and used software that should help students to learn about non-trivial models and algorithms of artificial intelligence, to awaken interest in discipline and at the same time to be accessible in terms of the complexity of mastering for students. Creating a model of a physical system, taking into account not ideal sensors and actuators, allows you to immerse yourself in the creation of real systems, taking into account the key features of the functioning of robots. The application of different models of behavior on the same robot model makes it possible to identify the strengths and weaknesses of both physical modeling and the AI algorithm. Since 2013 students of two faculties of Kuban State University have been studying programming for RoboCode and Mathlab Simulink.

© 2017 Published by Future Academy [www.FutureAcademy.org.UK](http://www.FutureAcademy.org.UK)

**Keywords:** Robotics, education, project-based learning, simulation environment



## **1. Introduction**

The courses "Artificial Intelligence Systems" and "Intellectual Information Systems" are included in many educational bachelor's programs in the faculty of information technology as basic or advanced courses.

In particular, these courses are taught by the authors of this paper in the fourth course of the Faculty of Computer Technology and Applied Mathematics at the Kuban State University for the specialties "010500 Mathematics and Information Systems Administration" and "230700 Applied Mathematics in Economics". Also, there is the experimental course "Introduction to Robotics and Artificial Intelligence Systems" for students of the Faculty of Physics and Technology and students of the Faculty of Applied Mathematics and Computer Science.

## **2. Problem Statement**

In this paper, the problem of teaching students of higher educational institutions of robotics with the application of the project-based learning and using the simulation environment is considered. These environments allow, at the initial stage of training, students to replace expensive experiments with real hardware.

Also, the use of packages such as MATLAB allows teachers to skip the training of engineering students directly to programming, as they provide a convenient interface for experimenting with physical laws.

In turn, the use of simulation simulations using the example of Robocode and Gazebo allows students of applied mathematics speciality to use ready-made physical engines and debug high-level algorithms.

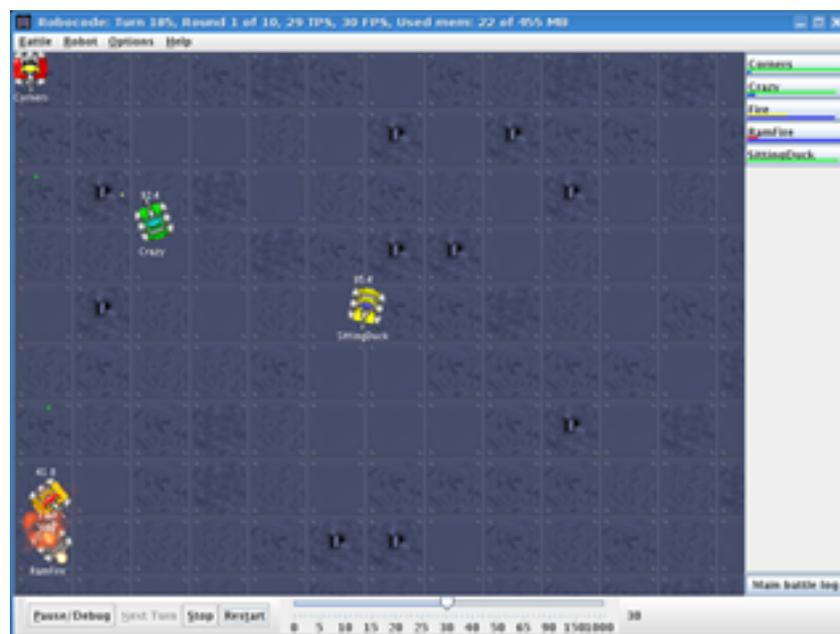
## **3. Research Questions**

The actual task when teaching such disciplines is the selection of practical tasks and used software that should help students to learn about non-trivial models and algorithms of artificial intelligence, to awaken interest in discipline and at the same time to be accessible in terms of the complexity of mastering for students.

One of the key sections of these training courses are intellectual agents (Russel & Norvig, 2003). Similarly, the division of tasks in one group, which involves the creation of a mathematical model of the kinematics and dynamics of the mechanism by physics students, and the behavioral algorithms by student programmers helps in practice to consolidate the theoretical knowledge gained about intellectual agents, increases student interest in the subject and training, strengthens interfaculty interaction and builds intercollegiate relations. That, in turn, makes it possible to organize competitions among students using computer simulation programs.

#### 4. Purpose of the Study

The Robocode simulator is one of the most mature, popular and convenient programs for conducting intelligent agent tournaments (Robocode official website). He simulates the battle of tanks on a square field without obstacles (Figure 1). In battle it is necessary to destroy enemy tanks, reducing the points of their lives to zero by hits from the gun or ram, and also to dodge enemy shots in order to survive the maximum amount of time in the battle. Matlab Simulink with the Simmechanics package allows to simulate mechanical systems using the ODE Solid State Physics simulation library, takes into account the friction of sliding and rolling of the robot wheels (Riadchykov et al, 2016), the dynamics when changing the trajectory of motion, the features of wheels drive systems and the tower.



**Figure 01.** The main window of Robocode simulator.

Virtual models are controlled by standalone programs, which can be written in various programming languages, including Java, C #, Prolog, Python. In programs there is a basic control loop and event handlers arriving at the beginning of each move. An example of the structure of the Robocode program in Java:

```
package MyPackage;  
  
// Connecting the required packages.  
  
public class MyRobot extends AdvancedRobot {  
  
    // Declaring global variables.  
  
    private void initializeRobot() {  
        // Initialization.  
    }  
}
```

```
private void personalizeRobot() {
}

public void run() {
    while (true) {
        // Main cycle
    }
}

public void onScannedRobot(ScannedRobotEvent e) {
    // Handling the event of enemy detection.
}

public void onHitRobot(HitRobotEvent e) {
    // Battle with opponent.
}
}
```

Sending commands, the robot can control the tracks to accelerate forward and backward, rotate, rotate the tower, rotate the radar on the tower, shoot with varying strength. Simple robots can only execute one command at a time. More complex robots can execute several commands asynchronously, for example, simultaneously move, rotate and shoot. Depending on the strength of the shot, the projectile flies at a different speed, damaging various forces to enemies and consuming a different amount of energy.

Programs can work independently or be combined into a command. The latter possibility can be used to train interaction in multi-agent systems.

Robocode allows teachers to make a tournament among programs with a graphical display of combat simulation with a given speed. The tournament contains a specified number of rounds, which end with the victory of one of the participants or after the expiration of time. Figure 2 shows the result of one of the tournaments.

Rank	Robot Name	Total Score	Survival	Surv Bonus	Bullet Dmg	Bullet Bonus	Ram Dmg * 2	Ram Bonus	1sts	2nds	3rds
1st	ru.ndem1.Destroyer9000 ...	5406 (38%)	2500	500	2070	319	13	5	10	0	0
2nd	tworogue.Velociraptor 1.0	2843 (20%)	1400	0	1303	122	18	0	0	5	1
3rd	PULLO.PulloRobot	1577 (11%)	800	0	748	1	28	0	0	0	2
4th	big.Puziko 1.0	1530 (11%)	750	0	623	17	139	0	0	1	2
5th	lenpack.Lenny 1.0	1506 (11%)	1200	0	288	0	18	0	0	3	3
6th	kampf.Floh 2.0	1479 (10%)	850	0	602	2	25	0	0	1	2

**Figure 02.** The results of simulation

## 5. Research Methods

In this paper the study of using Robocode simulation environment in the educational process with a comprehensive design approach in the training of robotics (fig. 3).

In Europe and USA de facto standards are the free Gazebo open simulation environment, as well as the MathWorks MATLAB & Simulink software package, which offer a more advanced physical simulation (Ryadchikov et al, 2017a).

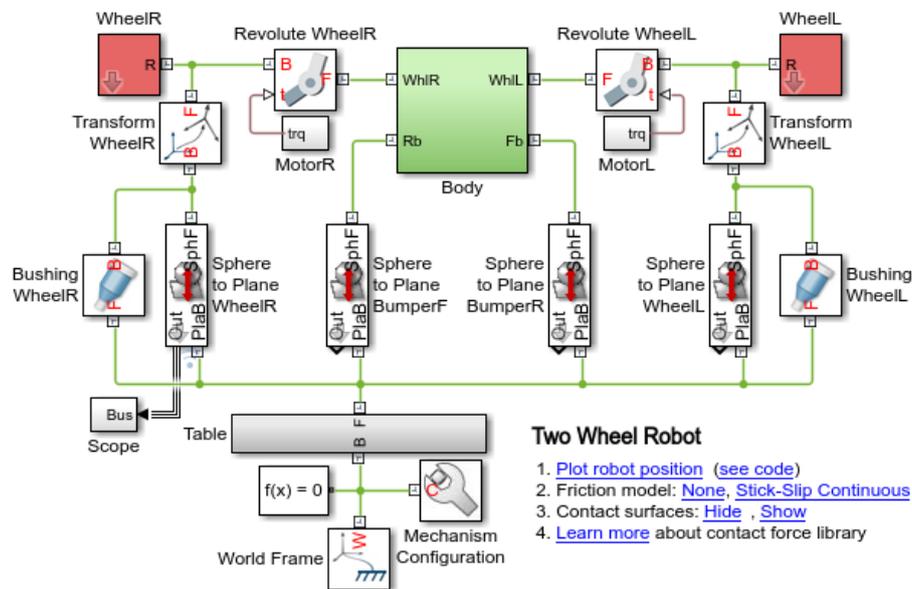


Figure 03. The simulation model of the cart in MATLAB Simulink.

## 6. Findings

Competitions on robotics in virtual environments among students are conducted by authors at the Faculty of Computer Technologies and Applied Mathematics of Kuban State University from 2012, and in the inter-departmental group since 2017.

Writing an intellectual agent program and participating in the competition is part of the author's educational program on artificial intelligence and is necessary for students to receive positive grades in this discipline. This encourages the implementation of robot control algorithms based on the environment model, containing elements of machine learning, forecasting (Ryadchikov et al, 2017b). The most successful and interesting algorithms are reported and discussed by the participants after the tournament. The most interesting classes of algorithms implemented by students:

1. Capture and track the target, switch to the closest and / or dangerous target.
2. Shooting with anticipation on the enemy, taking into account the prediction of its movements based on the results of previous observations.
3. Departure from the line of fire, avoidance of collisions from the analysis of the history of observations.
4. Choice of the strategy of behavior depending on the results of previous rounds.

Creating a model of a physical system, taking into account not ideal sensors and actuators, allows you to immerse yourself in the creation of real systems, taking into account the key features of the

functioning of robots. The application of different models of behavior on the same robot model makes it possible to identify the strengths and weaknesses of both physical modeling and the AI algorithm.

Programs for the first Kuban State University competition for RoboCode were written in the SWI Prolog language using the JPL library for pairing with Java (Alaiba & Rotaru, 2008). SWI Prolog was loaded as a dynamically connected library and all the programs of the tanks participating in the tournament were allocated to the common memory of Prolog. Because of this, the names of predicates and facts Prolog in different programs should not have coincided. In addition, the configuration of the Java bundle, Robocode, JPL, SWI Prolog is laborious enough and required the compilation of SWI Prolog from the source code on separate platforms. Therefore in the future SWI Prolog in tournaments on Robocode we are not used. However, this solution can be used to teach students in programs that, in addition to artificial intelligence, are taught a separate course of recursive-logical programming..

## 7. Conclusion

Since 2013 students of two faculties of Kuban State University have been studying programming for RoboCode in Java and C #.

Since 2014, the faculty tournaments on Robocode with the participation of students of several courses have been holding. In 2017, there will be an inter-faculty competition using Matlab and ODE physics engine for modeling the physical part of the system..

## Acknowledgments

The work was carried out within the framework of the state task of the Ministry of Education and Science, project No. 8.2321.2017 “Development and adaptation of control systems for compensation of dynamic deflecting effects on mobile objects in a state of dynamic equilibrium”

## References

- Alaiba V., Rotaru A. (2008). Agent architecture for building Robocode players with SWI-Prolog. In 2008. International Multiconference on Computer Science and Information Technology, 2008. IMCSIT, pp. 3-7.
- Riadchikov, I., Sechenev S., Sinitsa, S., Nikulchev E. (2016). Constructive solution of the robotic chassis AnyWalker, *ITM Web of Conference*, 6, 01003
- Robocode official website. URL: <http://robocode.sourceforge.net/>
- Russel S., Norvig P. (2003). Artificial Intelligence. A Modern Approach (2nd ed.). Prentice Hall.
- Ryadchikov, I., Sechenev, S., Sinitsa, S., ... & Nikulchev, E. (2017a) Design and Control of Self-Stabilizing Angular Robotics Anywalker. *International Journal of Advanced Computer Science and Applications*, 8(9), 29-34
- Ryadchikov, I., Sechenev S., Svidlov A., Sinitsa, S.s, Nikulchev E. (2017b) Development of a Self-Stabilizing Robotic Chassis for Industry, *MATEC Web of Conferences* 99, 02007