

A Robotic Simulator Tool for Mobile Robots

¹Mehmet Serdar Güzel,²Hakan Kayakökü,³Vahid Babaei

^{1,2,3}Faculty of Engineering, Department of Computer Engineering Ankara University, Turkey

Abstract:

This paper introduces a new open source simulator software (RST) for mobile robot applications including swarm robotics and multi-agent systems. The software is free and will be published under multiple licenses. The simulator environment is designed and implemented using the MATLAB platform which provides flexibility to both users and developers. This software can be also used in both undergraduate and graduate robotic course for educational reasons. The flexibility of the proposed tool allows developers adding extra modules to software depending on the required problems. The tool will be used in Ankara University robotics laboratory for preliminary robotic education during next semester.

Key words: Robotic software, open source tool, mobile robots, swarm robotics

1. Introduction

Simulation tools are mainly preferable when designing and analyzing of a physical system is expensive and risky [1,2]. Besides, they facilitate to reduce overall effort during teaching and learning phases [3]. Mobile robot navigation is a critical area in the field of robotics. In the last decade several comprehensive open-source software systems platforms are introduced including (Player/Stage [4], ROS [5] and MRS [6]). These platforms allow user to design and test conventional and/or novel algorithms in a reliable and safe manner for different problems without requiring any physical system. However main drawbacks of these software platforms are their complex structures that require high level of software and system programming skills. As well as, even any researcher has those programming skills, he/she may need several months to get used to corresponding platforms. Accordingly, RST has been designed and implemented by a group of researcher from robotics laboratory of computer engineering department of Ankara University since the second half of 2015. Despite, it is still quite primitive in terms of functional capacity and graphical support when compared with other popular platforms, it allows a reliable and flexible experimental environment for swarm robotics.

As well as, it covers several important navigation algorithms and has a fairly simple user interface that can be easily adapted by any student or researcher. In addition, RST has also advanced features such as Genetics Algorithm (GA) and Neural Networks (NNs), inherited and adapted from the libraries of MATLAB.

This paper is organized as follows: The second section, addresses the goals, tasks and learning objectives of the simulation tool, whereas the description of RST and architecture of this simulator based on examples are addressed in the third section. The final section concludes the paper.

2. RST Simulation Tool

The main screen of the proposed tool is illustrated in Figure 1. According to which, obstacle, goal and robot objects can be added into the working space, or removed from it dynamically. The main screen of the RST tool is designed based on the swarm architecture defined in [7] that is illustrated in Figure 2. This swarm architecture introduces a novel system that allows a group of robot to be able to group in a coordinated manner, and to select the leader in a decentralized approach while steering towards a predefined goal safely. Accordingly, the user can run this algorithm by enabling gravitational and leader selection algorithms respectively. The gravitational algorithm basically calculates the resultant force for each robot based on attractive forces applied by nearby robots. On the other hand, without considering those algorithms, a robot can move towards a predefined goal while avoiding obstacles using the Potential field method [8]. The robot moves the direction of the resultant vector inferred from the combination of attractive (the goal) and repulsive (goals and nearby robots) forces. Besides, the user can enable the Wavefront algorithm instead of the Potential field method after preparing a scenario. The Wavefront algorithm mainly estimates a path from start to end points through a discretized workspace [9].

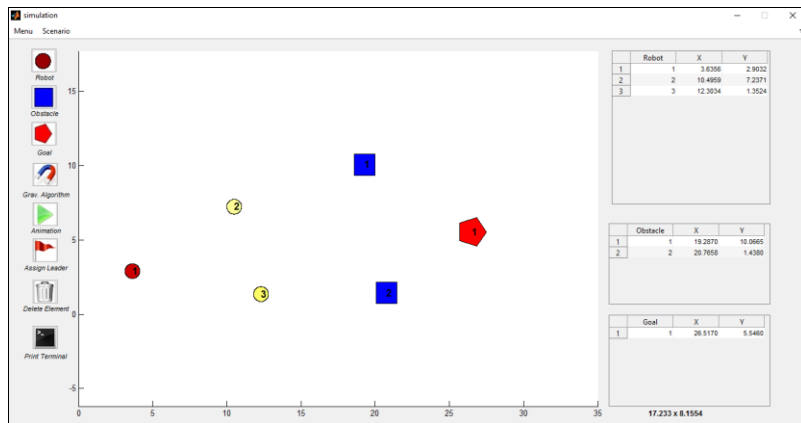


Figure 1: RST main screen with a simple scenario

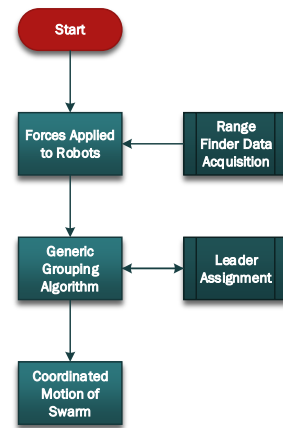
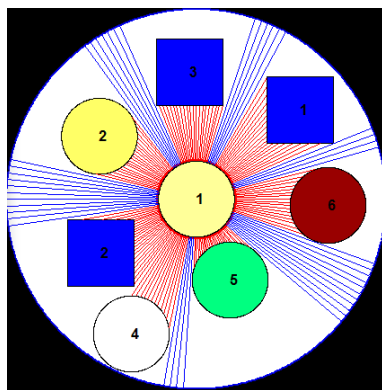
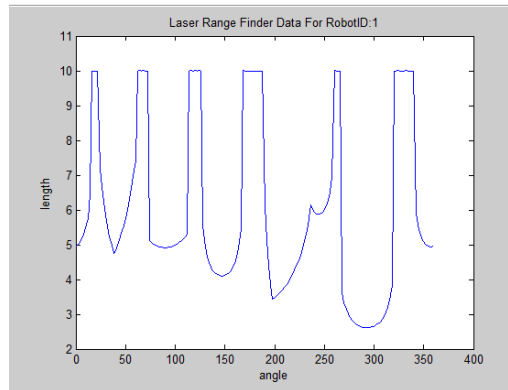


Figure 2: Swarm Architecture [7]



a-) A complex scenario



b-) Output of the range-finder tool

Figure 3: Range-Finder Tool, a-) A complex scenario, b-) Output of the range-finder tool

2.1. Range-Finder Simulation Tool

Another powerful tool provided by the RST is the Range-Finder Simulation Tool. This tool allows each robot to scan the environment in order to avoid obstacles and estimate positions of nearby robots. Each scan returns a point for each unit of angular resolution which denotes the distance between the robot and other objects. This tool simulates a 2-D Range-Finder which can scan 360°. An example scenario and the analysis of the Range-Finder's output are illustrated in Figures 3-a and 3-b.

2.2. AI Tool

RST employs the potential field algorithm as the main navigation strategy. The main drawback of this approach is its highly dependence on parameters. Hence, it is aimed to discover the optimum values for parameters of potential fields. This is an extremely important task, since the behavior of robot is entirely depending on them. Table 1 illustrates the corresponding parameters and their explanations for the potential field technique used in [7]. Defining an effective representation for a chromosome is one of the important GA design decisions, since, it determines the evaluation method for fitness value among with the types of genetic operators. Considering that the chromosomes represents a solution to the problem, each gene (allele) of the chromosome represents a value for the given parameters. In accordance with the desired values of our parameters we choose to encode real values in chromosome representation. A sample chromosome is represented below:

$$\{1.2|-2|-0.5|1.7|3|2\} \quad (1)$$

Upper and lower bounds of each variable is selected by trial and error method. Values of the variables will not cross these boundaries during the run time of genetic operators. The min and max values are as follows:

$$\text{min:}\{[1,3], [-3,-1], [-1,0], [1,3], [2,4], [0,2]\} \quad (2)$$

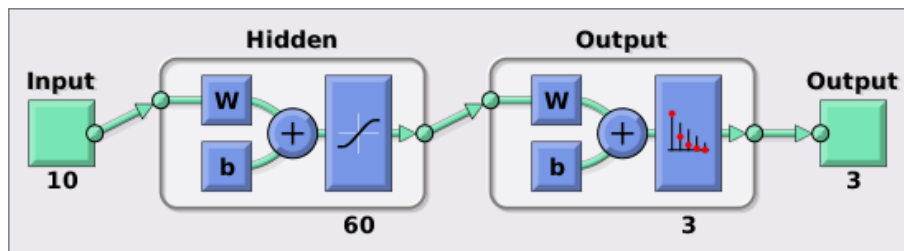


Figure 4: Neural Network design for Range-Finder Tool

Table 1. Parameters for potential field method

Parameters for Experiments

<u>Parameter</u>	<u>Explanation</u>
G_t	Maximum gravitational limit between robots while grouping
R_{mass}	Robots 'initial mass
O_{mass}	Obstacle mass
G_{bg}	Target mass before grouping
G_c	Number of robots belong to each group
A_{mass}	Attraction mass between robots after grouping



Figure 5: Training Result for Range-Finder Tool

2.3. Software Architecture

The layers of the proposed software architecture is illustrated in Figure 6. According to which, the first layer, is the Application layer that also includes plug-in. The second layer is the layer that includes the customized algorithms and modules, whereas the coordination integrates the customized features with the MATLAB toolbox. The interface layer provides the interaction between users and the proposed simulation software. The lower layer, on the other hand, comprises mathematical and AI tools of MATLAB.

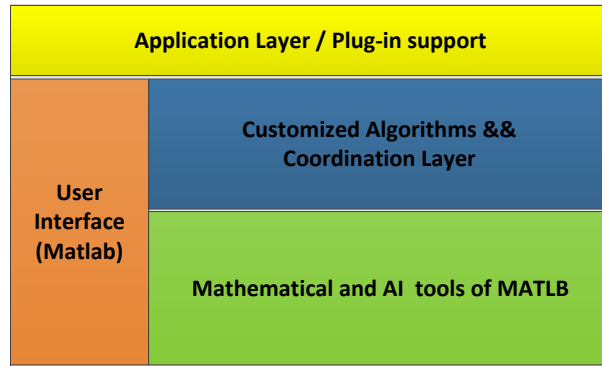


Figure 6: Layers of the proposed software architecture

3. Results

Several experiments are achieved in order to reveal the capacity of the proposed software. An example scenario is illustrated in Figure 7 where there exists 4 robots, 2 obstacles and 1 goal in the corresponding scenario. Once the leader is assigned to each group, COG point is calculated (see Figure 7) based on the parameters, shown in Table 1. As soon as the leader robot starts steering, member robots starts following the COG point. This point is adapted continuously within the movement of the leader as shown in Figure 7.d.

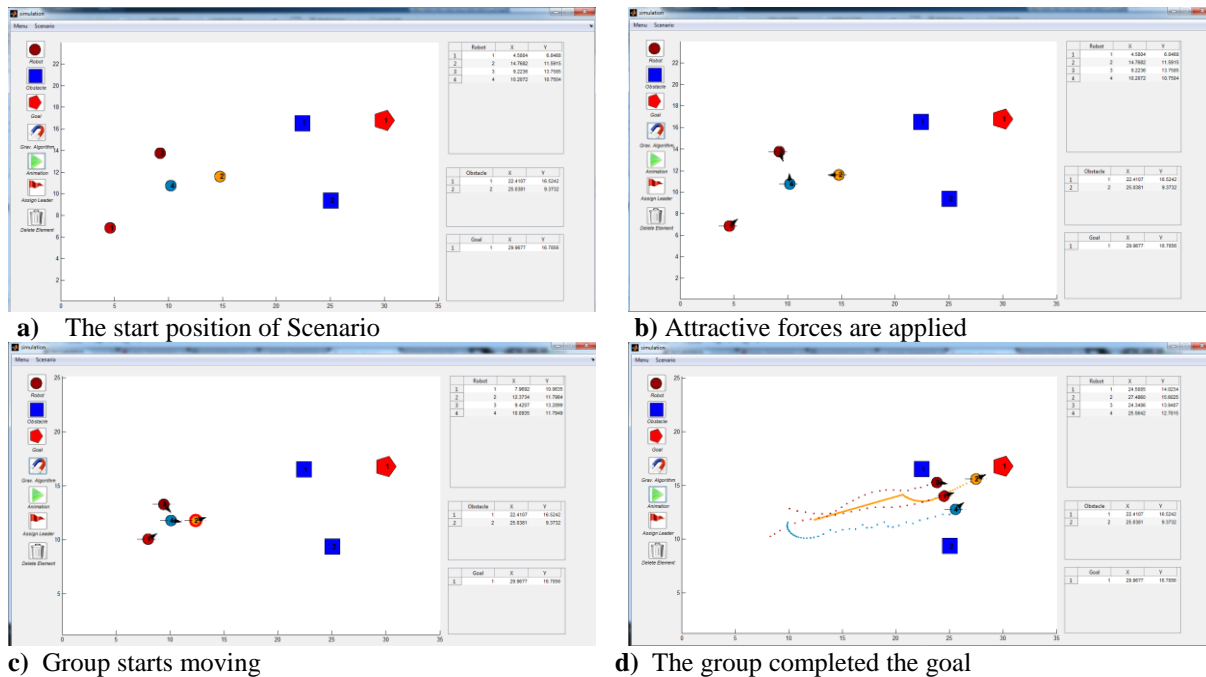


Figure 7: An example scenario, a) Start position, b) Attractive forces, c) Group starts moving, d) The group completed the goal

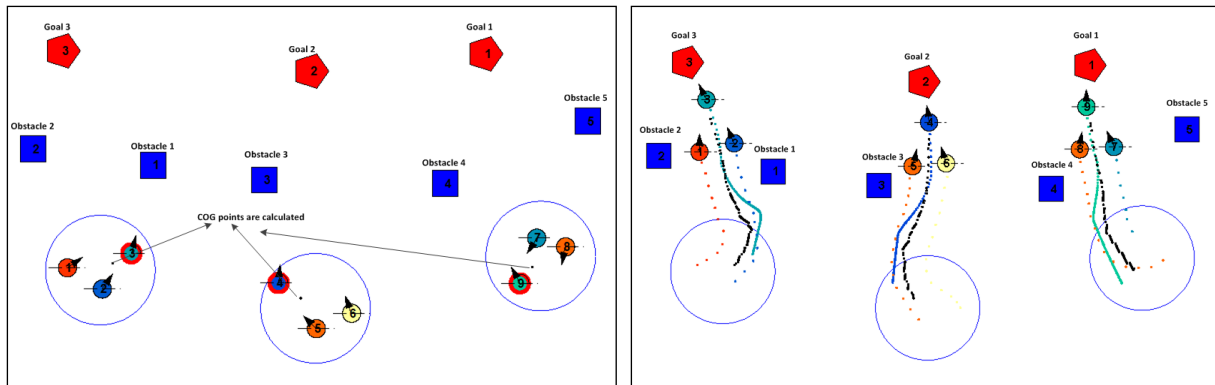


Figure 8: Another example scenario [7], a) 3 groups, 4 obstacles and 3 targets, b) Groups completed the goals.

Figure 8, on the other hand, illustrates another example including three groups that each group consists of 3 robots, 5 obstacles are located in the working environment and there exists target for each group [7].

4. Conclusions

This paper introduces a new simulation tool for mobile research field. The tool is designed based on MATLAB toolbox and has a user friendly interface. Compared with its counterparts, RST has a quite simple architecture and may probably be appreciated by any level of researcher due to its simplicity and flexible design. The tool also supports swarm intelligence and offers several algorithms in both mobile robot navigation and swarm robotics. RST will be an open source software which is available for other researchers who are allowed to access the software code and integrate extra modules easily with respect to their studies.

Acknowledgements

This work is a part of the research projects, supported by Scientific and Technological Research Council of Turkey (Tubitak Project No: 114E648)

References

- [1] Voinescu M, Caramihai S, Dumitrache I. A simulation tool for an agent-based mobile robot application. *Control Conference (ECC), 2009 European*, Budapest, 2009;2367-2371.
- [2] Güzel M.S, Hinislioglu Y. A robotic software for intelligent applications. 2013;18:76-82.
- [3] Wedeward K and Bruder S. Incorporating robotics into secondary education, *Robotics Manufacturing Automation and Control*, 2002;14.
- [4] Biggs G, Rusu R. B, Collett T, Gerkey B and Vaughan R. All the Robots Merely Players: History of Player and Stage Software. in *IEEE Robotics & Automation Magazine*, 2013; 20:3: 82-90.

- [5] Sukvichai K et al. Implementation of visual odometry estimation for underwater robot on ROS by using RaspberryPi 2. International Conference on Electronics, Information, and Communications (ICEIC), Danang, 2016; 1-4.
- [6] Zhou P and Yang L. Simulation Robot Soccer Based on Microsoft Robotics Studio. Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, Hangzhou, 2012; 639-645.
- [7] Guzel M.S. and Kayakökü H. A Collective Behaviour Framework for Multi-Agent Systems. 2nd International Conference on Mechatronics and Robotics Engineeri, 2016:61-71.
- [8] Tang L et al. A novel potential field method for obstacle avoidance and path planning of mobile robot. 3rd. IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010;633–637.
- [9] D. Reyes et al. Mobile Robot Navigation Assisted by GPS. IEEE Latin America Transactions, 2015; 6: 1915-1920.