Coordinated Movement of Multiple Robots in Outdoor Environment with Obstacles

Ján Šovčík*, Martin Kajan, František Duchoň, Martin Florek, Peter Hubinský and Khanh Duong Quang

Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovak Republic

Abstract: This article mainly focuses on the proposal of the program that coordinates the movement of multiple robots in outdoor environment. The emphasis of the proposed algorithm is on collision free movement. Simulation with four robots that are controlled in an environment with obstacles is the result of these experiments. The formation of robots consists of a leader and three following robots, which are trying to copy leader's movement. Robots use GPS, laser rangefinder, as well as bumpers sensors for the environment scanning. GPS is used as positioning sensor and other sensors are used for reactive navigation. Besides that, simulation uses virtual reality from Microsoft Robotics Studio 4 (MRS). In addition, programming language used for this simulation is called Visual Programming Language. Advantage of MRS usage is that the final code can be simply transferred into real robots.

Keywords: Multiple Robots, GPS, Microsoft Robotics Studio, differential drive, outdoor environment, obstacle, Introduction.

In this article the main focus is on the control of robots. The group of robots is very similar to the group of people in the terms of their behavior. More robots can make more work, cooperate with each other, send the information to distant places, and when it is necessary, they can replace each other.

The cooperation and replacing of the robots in situations as global catastrophes due to climate changes has led to the development of multi-robot solutions.

At this time, robots are frequently used in lifesaving operations, e. g. in zones hit by radiation, earthquakes etc. Moreover, humans still have far more better communication devices than the robots, and they can predict a lot of situations. However, their perception and motion skills are limited by human body. That is why the robots are often used in such catastrophic scenarios.

The emphasis in proposed algorithms will be mostly on rules, which are important during the seeking or tracking. All the proposed algorithms will respect the hardware constraints of used devices. The scenario takes into account obstacles, with which the robots must to deal with. The mission for the robots is to manage with the obstacles in terms of their avoiding or other operations with them. If the robot chooses to change the position of the obstacle, it must consequently put back this obstacle into the original place. In case that the ground and the formation will allow it, the robots will hold the obstacles during their motion in environment.

The formation includes one leading robot and three supervising robots. Their positions are determined by GPS system [2]. The remaining sensors used on robots are: laser rangefinder and tactile sensors. These sensors ensure the collision-free movement of the robots in environment.

For the navigation of each robot, several algorithms were implemented in Microsoft Robotics Studio and they were written in Visual Programming Language [3].

BASIC FORMATION

Keeping the formation in robotics can be compared to the formations of living organisms. In case that there will be a need for the robots to move forward from one place to another, it is valuable to use the formation similar to marching soldiers (Figure **1**). The algorithms supporting the keeping of such formation are proposed and implemented.

The huge advantage of this formation from the robotic point of view is its simplicity in the manner of following the leader. Each following robot just watches only one robot from the formation and does not need any information about the other robots. However, in this case, there is a possibility of a robot's damage. This can lead to the fact that the robot will stop working and

*Address correspondence to this author at the Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovak Republic; Tel: +421260291809; E-mail: jan.sovcik@stuba.sk

E-ISSN: 2409-9694/14

all the robots from the lower level of hierarchy will stop as well.

Figure 1: Formation of the soldiers.

The advantage of this method is that the leading robot has the information about all following robots and it can manage with any obstacles in the environment. Therefore the following robots will also trace such collision-free movement in the environment. The leading robot must of course consider the dimensions of the whole formation and avoid the obstacles in the adequate distance. In the Figure **2** there is symbolically shown the keeping of the formation. The leading robot (red color) does not follow any other robot. The remaining robots always follow the robot ahead of them or the robot on their right side (outlined by arrows).

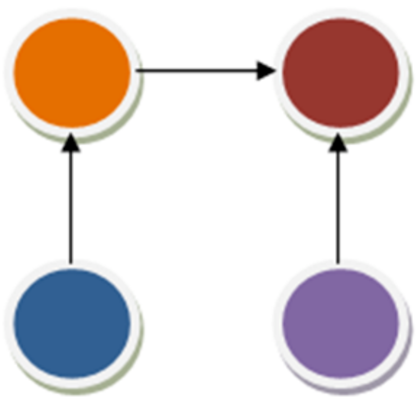

Figure 2: Basic formation.

ARMY FORMATION

Army formation is used when the robots must pass part of the environment containing narrow passages like tunnels, bridges or abysses. This formation is quite often seen in the nature. For example the ants use this formation when they are transferring the collected food. Similar formation can be applied to robots. When it comes to this formation, all the robots follow each other (Figure **3**).

Figure 3: Line formation.

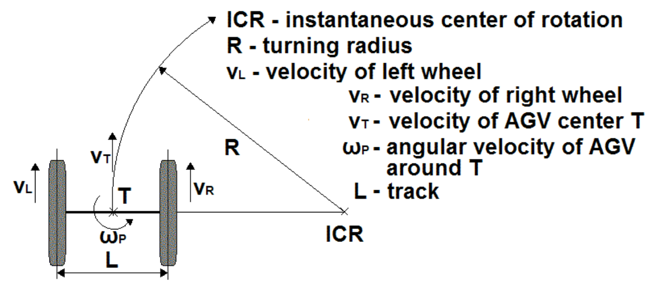

Figure 4: Description of differential driven robot [7].

Figure **4** describes the basic kinematics used for individual robot control. Following equations describe such control:

$$R = \frac{L}{2}\frac{v_R + v_L}{v_R - v_L} \quad (1)$$

$$v_T = \frac{v_L + v_R}{2} \quad (2)$$

$$\omega_P = \frac{v_R - v_L}{L} \quad (3)$$

Constant L in equation (1) represents the distance between the wheels of the robot. In this case a real robot Pioneer 3 – DX was simulated and its distance between wheels is equal to 333,6 mm. Parameter v_t in equation (2) represents the speed of robot's center of gravity and the maximum value of this parameter was limited to 1 ms^{-1}. With modifications of equations (1) and (2) the following equations can be derived [4]:

$$v_R = v_T + \frac{v_T L}{2R}$$
$$v_L = v_T - \frac{v_T L}{2R} \quad (4)$$

Because of the fact that the result of the equations could be various numbers and simulated differential drive in MRS works only with numbers in the range <-1;1>, equation (4) was modified:

$$v_R = \frac{v_R}{abs(v_R) + abs(v_L)}$$
$$v_L = \frac{v_L}{abs(v_R) + abs(v_L)} \quad (5)$$

With this modification, range of the controlled parameters is kept in the range <-1;1> within the maintaining of the basic principles of differential drive control.

IMPLEMENTED ALGORITHM

The robot, whose position is on the right upper side of formation, is the leading robot. This is the only one who does not have any implemented algorithm, which characterizes the behavior of the robot. This leading robot is fully controlled by the human operator. It broadcasts to the other robots its own position from GPS sensor. This position data are used in control algorithms of other robots to determine the positions where the remaining robots should be moved to. [5] The leading robot is always in the formation in the front that is why it is also closer to any obstacle than the other robots. If the leading robot detects the obstacle by means of its laser rangefinder, the formation of the robots is changed from the square to the army formation. This formation was analyzed as the most suitable formation for the obstacle avoidance.

The relation between the leading robot and his followers is described by the two parameters. First represents the distance between the leader and follower and the second describes relative angle between the robots in world coordinate system. These parameters are used as controlled variables in implemented algorithm. Generalized algorithm describing the behavior of followers was implemented into each robot. Individual followers are different from each other only in determination of each parameter. Moreover, the determination of these parameters is dependent on the selected formation. In case, when leading robot has some rotation (α) against world coordinate system, angle α is used for compensation of the rotation of each follower.

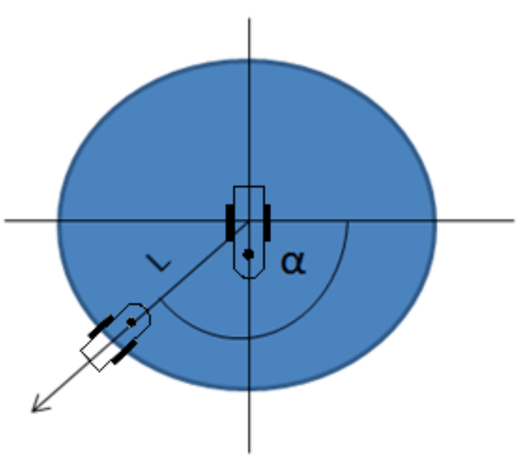

Figure 5: Robots following.

The implemented algorithm for the following of leader robot is described by these equations:

$$X_{slave} = X_{master} + L * \sin\theta$$
$$Z_{slave} = Z_{master} + L * \cos\theta \quad (6)$$

Where L is the distance between the follower and leader, θ is the relative angle between the follower and leader, and $X_{maseter}$ and $Z_{maseter}$ are the coordinates of the leader in world coordinate system. Accordingly the desired positions of the followers in the formation are modified by the minus sign of the second member in equations (6). For example the desired position of the robot, which is placed on the left side of leading robot:

$$X_{slave} = X_{master} - L * \cos\theta$$
$$Z_{slave} = Z_{master} + L * \sin\theta \quad (7)$$

The distance between the robots is described as follows [4]:

$$D = \sqrt{(X_{master} - X_{slave})^2 + (Z_{master} - Z_{slave})^2} \quad (8)$$

This equation must be created as a quite huge enrollment in the code (Figure **6**), because there is absence of exponential function.

The angle between the following robot and local coordinate system of the leader is computed:

$$\propto = \tan^{-1}\left(\frac{Z_{master} - Z_{slave}}{X_{master} - X_{slave}}\right) \quad (9)$$

Representation of equation (9) in the Visual Programming Language is shown in Figure **7**.

In the case of circular movement (Figure **8**), the point of rotation is not placed at the line between the robots. The point of rotation is placed in the center of circumscribed circle, which describes the position of both robots. Radius of this circle is variable and it is calculated:

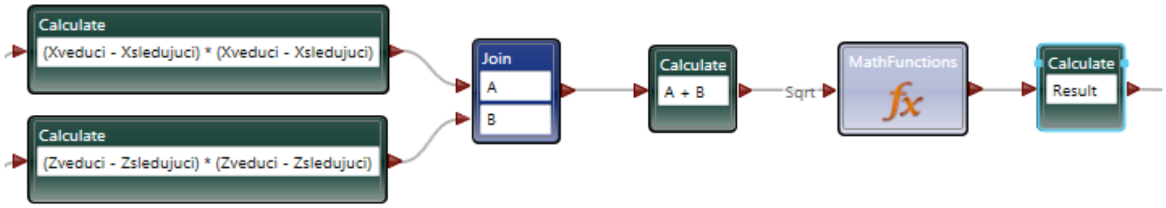

Figure 6: Distance code.

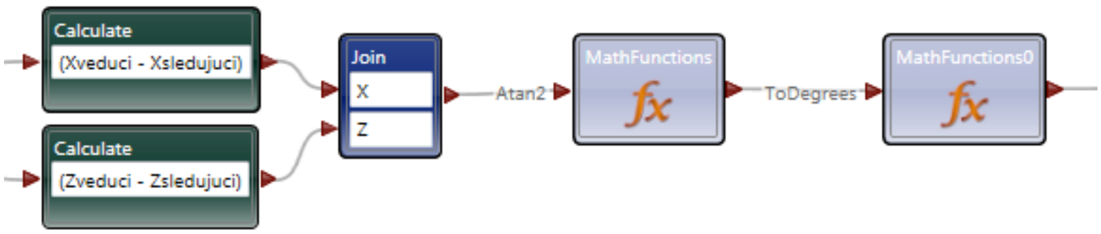

Figure 7: Code example.

$$R = \frac{c}{2\sin\frac{\theta}{2}} \tag{10}$$

Spiral moment doesn't occur, like in case when robot moves on circle during fast calculation of actual position. Sampling time is set to 100ms.

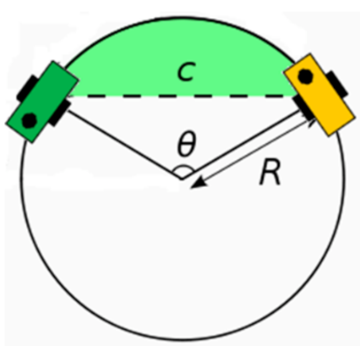

Figure 8: Formation during the rotation.

SAFETY MOVEMENT

For the limiting of possible dangerous situations, contact sensors were used (Figure **9**). In this simulation they are represented by bumpers, which are installed on the front part of the robots. Bumpers transmit only logical values: *true* or *false*. *True* represents pressed bumper and *false* is a status, when bumper goes from pressed to released. In a case when bumper is not pressed, this simulation block representing bumpers transmits signal and values on output that are *null*. If this status occurred in program, robot will stop and wait for unlocking of the path.

OBSTACLE AVOIDANCE – METHOD 1

The simplest way to avoid the obstacles is to compensate the dimensions of the formation in the leader's algorithm. This means that it is needed to hold sufficient distance between leading robot and the obstacle (Figures **10** and **11**). This method is very useful in a real world conditions. Leading robot has to calculate dimensions of the formation and it will use it as its own dimension. This method can be used only when the leading robot has reliable navigation method or the human operator has good coordination abilities.

Another implemented method of obstacle avoidance is characterized by the change from square formation

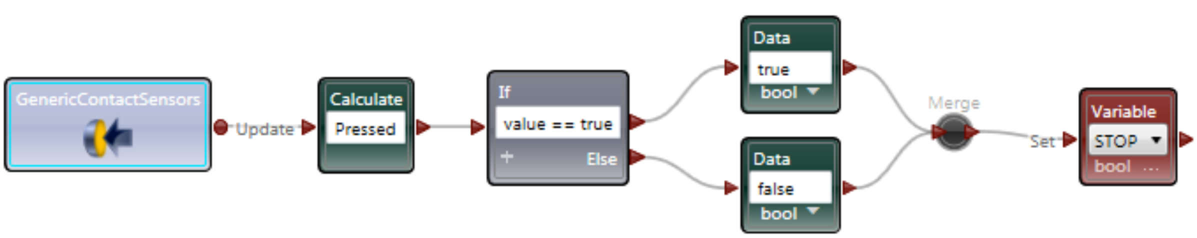

Figure 9: Contact sensor usage.

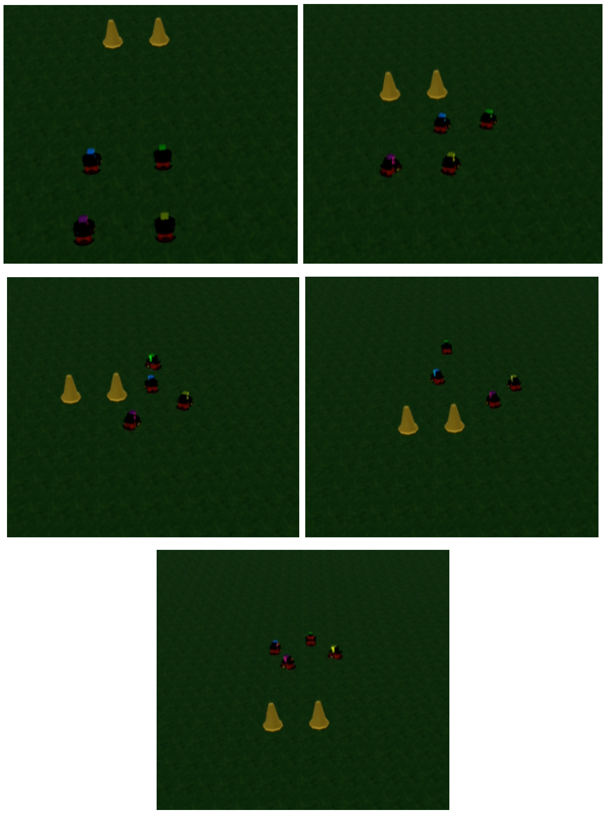

Figure 10: Simulation demonstration.

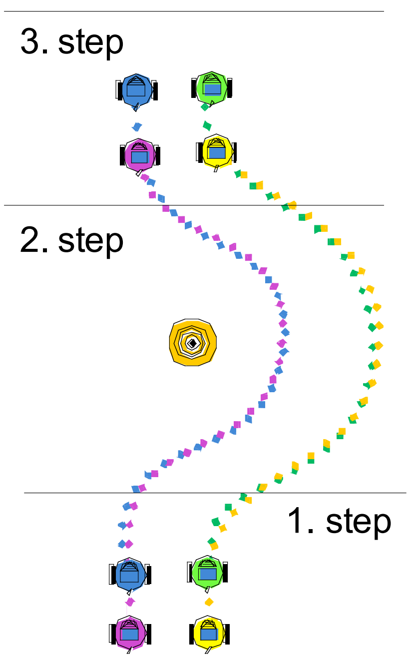

Figure 11: Simulation steps.

to the line formation. If this change takes place, the width of whole formation is reduced to a single robot. Laser rangefinder of the leader was used to detect the obstacles in the environment. The formation change is performed in the following steps:

- Obstacle recognition.
- Extending the distance between leading robot *red* and following robot *purple* (Figure **12**).
- The transition into the crowd (Figure **12**).

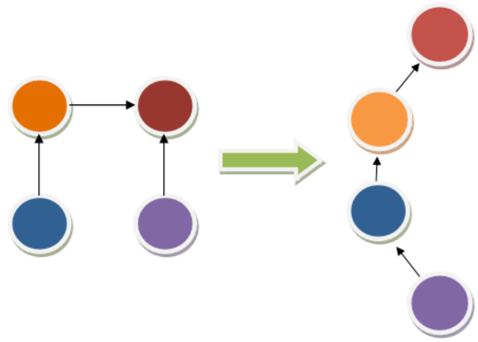

Figure 12: Formation changing.

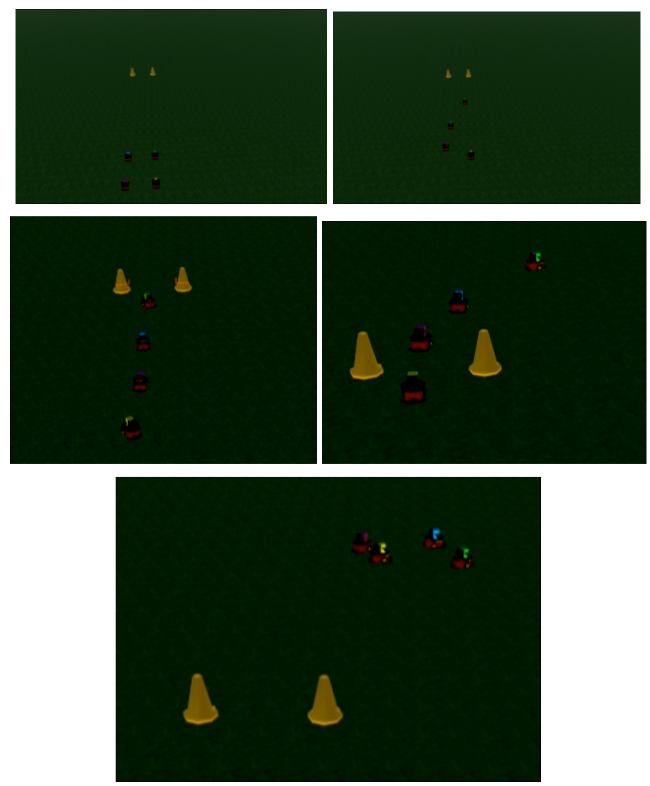

Figure 13: Simulation demonstration.

FUTURE WORK

Future work consists in several points. First, simulation will be extended into the third coordinate, which will allow simulation of flying robots. Second, the algorithms should be verified on the real robots. Third, several other sensors (e. g. ultrasonic rangefinder) should be used to secure the reliability and safety of whole system.

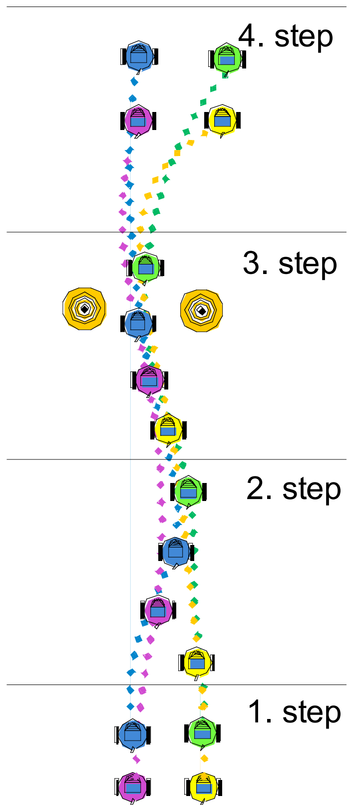

Figure 14: Simulation steps.

CONCLUSION

This article focuses on development of control algorithm for multiple robots in outdoor environment. For the positioning, the GPS sensors were used, because these sensors are easily applicable for outdoor localization. Due to usage of models in virtual reality, data from the sensors do not have to be modified in any way. In the real world conditions, data should be modified by some filters, e. g. Kalman filter.

Microsoft Robotics Studio was chosen as a suitable simulation environment, which adequately replicates the real physical properties of the world. Implemented algorithms were created using Visual Programming Language, which is one of the most user-friendly graphical languages. The programming environment is very easy to use. The advantage of MRS is the parallelism of the individual calculations and simple data acquisition from sensors.

The final result of this work is a simulation of the formation containing four robots. These robots are representing army in some way, which moves in outdoor environment. The formation can change using the control buttons. For the testing of the implemented algorithms, two simple obstacles were placed into the environment. The robots were able to avoid these obstacles if the distance between the leader robot and obstacle is sufficient, or when the robots can change the formation from the square formation to the line formation.

At the beginning, only one or two robots were simulated in MRS. If four robots were deployed, simulation had a very long response time. Therefore it is not recommended to simulate more than 4 robots in MRS.

ACKNOWLEDGEMENT

This work was supported by project VEGA no. 1/0178/13.

REFERENCES

[1] Jurišica L, Vitkoa A, Duchoň F, Kaštan D. Systém GPS - princíp, prednosti a nedostatky [updated 2010; cited 2013 Oct 13]: Available from: http://www.odbornecasopisy.cz/index.php?id_document=42738.

[2] Utilities for Robot Simulation. http://www.conscious-robots.com/en/robotics-studio/robotics-studio-services/crubots.html. (accessed October 13, 2014).

[3] Kyle J, Taylor T. Professional Microsoft® Robotics Developer Studio. Indianapolis: Wiley Publishing, Inc., 2008.

[4] Velagic J, Osmic N, Hodzic F, Siljak H. Outdoor Navigation of a Mobile Robot Using GPS and GPRS Communication System[updated 2011 Sep 16; cited 2013 Oct 13]: Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6044302.

[5] Boeing A, Bräunl T, Reid R, Morgan A, Vinsen. Cooperative Multi-Robot Navigation and Mapping of Unknown Terrain [updated 2010; cited 2013 Oct 13]: Available from: http://rrfx.net/files/RAM2011_Boeing_Braunl_Reid_Morgan_Vinsen.pdf.

[6] Kakinuma K, Hashimoto M, Takahashi K. Outdoor Pedestrian Tracking by Multiple Mobile Robots Based on SLAM and GPS Fusion; [updated 2012 Dec 16; cited 2013 Oct 13]: Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6427329.

[7] Kajan M, Mrafko L, Duchoň F, Hubinský P, Šovčík J. Control of Automated Guided Vehicle with PLC SIMATIC ET200S CPU; [updated 2013 Nov 5; cited 2013 Oct 13]: Available from: http://pubs.sciepub.com/ajme/1/7/38.

Received on 20-10-2014 Accepted on 01-11-2014 Published on 09-01-2015

DOI: http://dx.doi.org/10.15377/2409-9694.2014.01.02.2