Algorithms and models of multirobot systems and their implementation in ROS

To cite this article: M V Shikhman and S V Shidlovskiy 2019 IOP Conf. Ser.: Mater. Sci. Eng. 696 012015

View the article online for updates and enhancements.
Algorithms and models of multirobot systems and their implementation in ROS

M V Shikhman and S V Shidlovskiy
National Research Tomsk State University, Tomsk, Russia

E-mail: Shikhmar@gmail.com

Abstract. Conducting research in the Arctic takes place in extreme conditions. In these conditions, it's effective to use multirobot systems that can function for days under any weather conditions. The article discusses the basic principles of multirobot control systems. Mathematical models for the formation of control actions of multirobot are obtained. The basic methods for modeling multirobot systems in ROS are described.

1. Introduction

The development of robotics allows the use of robotic systems in all spheres of human activity. Robotic systems are a combination of sensory, actuating and microcontroller devices. Their collaboration allows us to achieve various goals and to solve a wide range of technological problems. The use of robots is especially effective in places inaccessible to humans, as well as when performing work associated with risks to human life and health. Robotic systems can work around the clock in all weather conditions. These advantages are especially important when carrying out work in the Arctic, and that is why achievements in this area play an important role in the effective development of the Arctic lands.

2. Multirobot control systems

Even greater efficiency can be achieved with groups of robots. Multirobot systems (MRS) allow you to expand the radius of the system, to achieve your goals better because of the distribution of tasks. However, it is obvious that all this entails an increase in complexity in the development and operation of such systems.

Multirobot system is a group of mobile robots that coordinate their actions to achieve a common goal [1]. Thus, the task of multirobot control system is to form control actions for each robot, ensuring optimal achievement of the goal. The system control scheme looks as follows (Figure 1). The system has a specific goal, the multirobot control system generates control signals for each of the robots in accordance with some algorithms. In addition, this system must take into account the state of the external environment for the formation of control actions. The received signals in some way change the state of each of the robots, and as a result, robots also interact with the environment too. Information about changes in the state of the group, as well as changes in the environment should return to the control system. This is necessary in order to understand whether this goal has been achieved and what needs to be done next to achieve it.
3. Coordination strategies for multirobot systems

The operation of the multirobot control system can be based on various control algorithms. Depending on this, the following strategies can be used: centralized MRS and decentralized MRS, which is divided into collective and swarm MRS (Figure 2) [2].

![Diagram of coordination strategies for multirobot systems](image)

**Figure 2.** Coordination strategies for multirobot systems.

The strategy of centralized coordination implies the presence of some central control device, which completely takes all the necessary decisions to achieve the goal, and also forms the control actions for all robots (Figure 3).

![Diagram of strategy of centralized coordination](image)

**Figure 3.** Strategy of centralized coordination.

A type of centralized MRS is hierarchical MRS. The peculiarity of hierarchical coordination is that the central device transfers control actions to a limited group of control devices (central devices of the second level), and it already generates control actions for all the remaining robots. The central devices of the second level can serve as other robots. In addition, there may be several levels of hierarchical access (Figure 4).
The essence of a decentralized coordination strategy is that each robot in a group independently makes decisions about its actions. In the case of swarm MRS, robots make decisions only based on the purpose and interactions with the environment (Figure 5). If the coordination is collective, the robots interact with each other to make more effective decisions (Figure 6). Thus, the difference between swarm and collective coordination is the presence of some kind of information exchange channel between robots.

4. Mathematical model for the formation of control actions for robots
Consider the formation of control actions for moving in multirobot system. With a central coordination strategy, the central control device or central robot can set the final goal in front of the robot, that is, set the coordinates of movement. In this case, the formation of control actions is not difficult and can be considered as the formation of control actions for a single robot. Another situation where the movement of robots in a group depends on the movements of the central robot. In this case, consider the central robot (the leader robot) and the robots following it as material points on the plane.

Let the central robot at some point in time have the coordinates \((x_c, y_c)\). The remaining robots of the group, which consists of \(i\) robots, will have coordinates \((x_i, y_i)\) and an angle of rotation relative to a rectilinear trajectory \(\theta_i\). The distance between the central and \(i\)-th robots will be equal to \(x_{di}\) and \(y_{di}\) along the corresponding axes (Figure 7).
Then the system that combines all these parameters will look like [3]:

\[
\begin{bmatrix}
    x_c \\
    y_c \\
    \theta_c
\end{bmatrix} =
\begin{bmatrix}
    x_c + x_d \cos \theta - y_d \sin \theta \\
    y_c + x_d \sin \theta + y_d \cos \theta \\
    \theta
\end{bmatrix}.
\]

We define a kinematic model of the motion of the central robot. Let us denote the linear speed of the robot \(\upsilon_c\) and its rotation speed \(\omega_c\). At the initial time, the robot has coordinates \(y_c, x_c\). Then, when the robot rotates by a certain angle \(\theta_c\), the following differential expressions can be written:

\[
\begin{bmatrix}
    \dot{x}_c \\
    \dot{y}_c \\
    \dot{\theta}_c
\end{bmatrix} =
\begin{bmatrix}
    \upsilon_c \cos \theta_c \\
    \upsilon_c \sin \theta_c \\
    \omega_c
\end{bmatrix},
\]

where \(\theta_c\) is a set of pairs of closest points; \(\dot{x}_c\) – x-axis instantaneous speed; \(\dot{y}_c\) – y-axis instantaneous speed.

Combining the two data systems, we get:

\[
\begin{bmatrix}
    \dot{x}_c \\
    \dot{y}_c \\
    \dot{\theta}_c
\end{bmatrix} =
\begin{bmatrix}
    B_x \begin{bmatrix} \upsilon_c \\ \omega_c \end{bmatrix}
    \\
    B_y \begin{bmatrix} \upsilon_c \\ \omega_c \end{bmatrix}
\end{bmatrix},
\]

where

\[
B_x =
\begin{bmatrix}
    \cos \theta_c & -x_d \sin \theta - y_d \cos \theta \\
    \sin \theta_c & x_d \cos \theta - y_d \sin \theta
\end{bmatrix},
\]

\[
B_y = [0 \ 1].
\]

Knowing the coordinates of each of the robots, you can calculate the length of the vector between them. For the \(i\)-th robot [4]:

\[
L_i = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}.
\]

When we need to move the robot, we ask him the coordinates in which he should be as a result of this movement. By changing its position in space, the central robot will change its coordinates, which will become \((x_{c1}, y_{c2})\). Knowing the corresponding coordinates, we can find the motion vector by the formula:

\[
\overline{R}_c = (x_c - x_{c1}, y_c - y_{c1}),
\]

where \((x_c, y_c)\) – coordinates of the starting point in the time interval; \((x_{c1}, y_{c1})\) – coordinates of the end point in the time interval.

Based on this, we can find the angle of this vector by the formula:

\[
\theta_c = \arctg \left( \frac{y_c - y_{c1}}{x_c - x_{c1}} \right).
\]

The obtained result allows one to form a control action on the robotic engines for the group, that is, to set robots the angles of movement \(\theta_i\).

This approach can also be used in a strategy of collective coordination to avoid collisions between robots and increase the radius of the system by controlling the distance between robots.

5. The basics of implementing multirobot systems in ROS

Modeling is effectively used in the development of robotic systems; this allows us to evaluate the effectiveness of certain algorithms in various conditions. One of the most popular modeling tools is the ROS (Robot Operating System). Let us consider the basic principles of building multirobot systems in ROS.

5.1. Adding a multi-robot
The first thing you need when modeling a system in ROS is to place several robots in a production environment. The environment and robot simulator work through the World file, which contains a description of the scene for the simulation. This is because of the fact that you can add the desired number of robots to the scene. Let us consider this with the example of the ROS Stage Simulator. Stage simulates a population of mobile robots, sensors and objects in a 2D bitmapped environment. Adding the following lines to the world file, we get a simulation of three robots in one environment (Figure 8):

```c
# robots
erratic( pose [ -11.5 23.5 0 0 ] name "robot0" color "yellow")
erratic( pose [ -13.5 23.5 0 0 ] name "robot1" color "red")
erratic( pose [ -12.5 21.5 0 0 ] name "robot2" color "green").
```

![Figure 8. Adding a multi-robot.](image)

5.2. **Robot control**

The next task is to send control actions to robots. Suppose we need to signal a certain robot so that it goes straight. There are two ways to do this.

1. For example, to make robot 2 move forward type: `rostopic pub /robot_2/cmd_vel -r 10 geometry_msgs/Twist '{linear: {x: 0.2}}'`.

2. You can create a ROS node that will make one of the robots move forward. To do this, create a launch file and add the following code there:

```xml
<launch>
  <node name="stage" pkg="stage_ros" type="stageros" args="$(find stage_multi)/world/willow-multi-erratic.world"/>
  <node name="move_robot_2" pkg="stage_multi" type="move_robot" args="2" output="screen"/>
</launch>
```

The result will be the same in both cases (Figure 9).

![Figure 9. Robot control.](image)

5.3. **Coordinate between robots**
When using the strategy of collective coordination, you need to coordinate and synchronize behavior between robots. For example, we create a condition under which robots begin to move only after they receive messages about the readiness of each of the robots. For this, it is necessary to create a shared topic in which robots will publish a ready status message; when such a message arrives from each robot, the monitoring node will publish a broadcast message stating that all robots are ready. Messages are described using msg files; the structure of readiness messages has the following form:

   Header header
   int32 robot_id
   bool is_ready.

Next, we need to create the monitor node that will listen for the ready messages and announce when all robots are ready. As a result, we will be able to receive information about the readiness of robots (Figure 10). When all the robots are ready, they will begin to move (Figure 11).

![Figure 10. Robots are ready.](image)

![Figure 11. Begin to move.](image)

6. Conclusion
Multirobot control systems are effective in extreme conditions of the Arctic. Different group control strategies allow you to use them for various purposes. One of the convenient software environments for implementing group control of robots is ROS, which allows both strategies of centralized and decentralized coordination.

Acknowledgments
The research was carried out in Tomsk State University with the financial support of the Ministry of Science and Higher Education of the Russian Federation, a project «Development of an intelligent control system for a group of robotic machines in the tasks of the agro-industrial complex»
The authors are grateful to Tatiana B. Rumyantseva from Tomsk State University for English language editing.
References

[1] Zhi Yan, Nicolas Jouandeau, Arab Ali Cherif 2013 *International Journal of Advanced Robotic Systems* **10(12)** doi 10.5772/57313

[2] Kalyayev I A 2001 *Modeli i algoritmy kollektivnogo upravleniya v gruppakh robotov* (Moscow: Fizmatlit) p 280

[3] Toru Namerikawa, Chika Yoshioka 2008 *IFAC Proceedings Volumes* pp 5149-5154 doi 10.3182/20080706-5-KR-1001.00865

[4] Ilyichev K V, Mantserov S A 2017 *Vestnik PNIPU* **21** pp 91-108