PREDICTION AND COMPARISON FUNCTIONS IN SEMI-AUTOMATIC CONTROL

Veronika Kozhukh
United Institute of Informatics Problems
of the National Academy of Sciences of Belarus
nika.rfe@gmail.com

Mobile robotics is one of the major directions in the field of intelligent robotics systems. It gives an opportunity to presence in areas human cannot or does not want to access. Mobility provides a great range of various applications for robots. Mobile robotics is on the way to be autonomous but still it can't operate in complex environment without outer controller. Nevertheless, there exists a lot situation where controller command can be wrong and leads to robot's failure [1].

The most demanded control for mobile robotic systems is remote control. System does exactly what says controller; controller chooses a command based on the given data from system. Common remote control is shown on the figure 1.

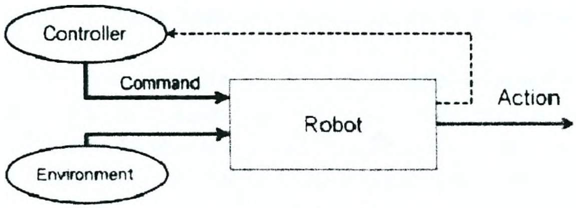

Figure 1 – Common remote control

Let's consider the situation when controller command doesn't reach mobile robot or controller can't receive the information about environment. Such cases can arise by several reasons; e.x. data link is out of reach. So, when robot doesn't have controller command it just take action by command from itself (we consider the one which has all needed computing skills). It is semi-automatic control model and represented by having several control commands: controller command and command generated by robot (figure 2).

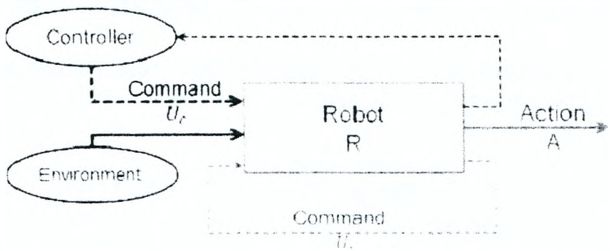

Figure 2 – Semi-automatic remote control

It is obvious when MR takes the action due his own control command only if he doesn't have any connection with the controller. But in other case we have two commands mobile robot to do. So somehow, robot has to choose one. The easiest solution would be when robot command is active only at the time when connection is lost. But loosing connection is not single condition to consider cases of incorrect controller commands. We have to deal with other conditions, such as mistaken interpretation of controller commands or controller wrong commands (the ones that lead robot damage).

On the figure 3 is shown case when controller makes wrong decision (controller command is in red color and robot command is in orange color). In the narrow corridor with obstacle straight and exit on the left. Controller doesn't see the obstacle and his command "straight" in the best way stop robot and in the worse way it can lead robot failure. At that time robot calculates the exit from the corridor. To stay intact it has to ignore controller command and to move left.

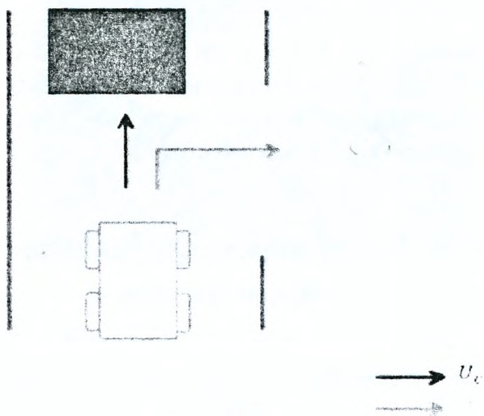

Figure 3 – Two possible commands

So, to find out what command is better robot has to calculate results of each command. For this case can be used prediction function (   ) [2].

Prediction function determines robot state in the next time step.

In that way are counted prediction functions for all types of controllers.

- prediction function of controller
- prediction function of robot

But our goal is some action to take. To figure it out is needed to compare the results of each command.

$$A = C(F_p(R, U_c)F_p(R, U_r), CL)$$

This formula shows what action takes robot. C is comparison function to choose which of predicted results is better. But this function also has to consider who makes less errors, whom robot can trust. For that reason we've included a variable ***CL***, which means a confidence level. It's a dynamic parameter that collects all result of previous evaluations and aggregates wrong decisions for each type of control. More actions are taken - more information about controllers is collected.

Moreover, in prediction function can be calculated not only appropriate state at the next moment of time, but some general goal. For instance, case when robot is to reach some point and complex of controllers commands leads other direction is set as wrong and robot has to search way by itself. Another of the ways to use such approach is controlling a group of robots by one controller. Controller doesn't need to communicate with every agent in group but sends some general commands.

References

[1.] Dudek, G. Computational Principles Of Mobile Robotics. Second Edition / G.Dudek, M. Jenkin - UK: Cambridge University Press,2010. - 11 p.

[2.] Norwig, P. Artificial Intelligence, A Modern Approach. Second Edition / S. Russel, P. Norwig – USA. Prantice Hall, 2003. - 557 p.

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ АВТОНОМНЫМ МОБИЛЬНЫМ РОБОТОМ

Дёмин В.В., Кабыш А.С., Головко В.А.
Брестский государственный технический университет,
vvdemin@bstu.by

Рассматривается система эффективного управления мобильным роботом, в основе которой лежат алгоритмы обучения с подкреплением для сети колесных модулей. В рамках предлагаемого подхода эта сеть рассматривается как многоагентная система, в которой координация поведений агентов осуществляется виртуальным лидером. Предложена модифицированная модель обучения с подкреплением для адаптивной координации индивидуальных стратегий. Модифицированный Q learning алгоритм проводит обучение агентов эффективному управлению каждым колесом, в контексте группы, что позволяет агентам подстраиваться друг под друга.

Мобильные роботы, обучение с подкреплением, многоагентные системы, алгоритмы интеллектуального управления.