The Decision-Making System for a Multi-Channel Robotic Device Control

Alexander A. Gridnev*, Timofei I. Voznenko†, and Eugene V. Chepin‡

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Moscow, Russia

Abstract
Nowadays, there is a trend of significant robotic devices functionality increase. Accordingly, the robotic devices can perform more operations and have more new ways to control them. Therefore, in some cases there is a situation when the robotic device receives control commands, i.e. instructions for performing different operations from different alternative control channels. It is a problem of a command selection for execution when different control channels have contradictory information. If there are multiple control channels, which have fundamentally different principle of operation, and multiple commands, which might require the same resources for execution, we need a system that will take into account the specifics of channels and commands, store information about incoming commands and decide which command to execute. In this paper we will present decision-making system with an internal control conflict resolution mechanism for robotics and the implementation example.

Keywords: decision-making, control channel, command, robotics

1 Introduction

Modern mobile robotic systems are complex heterogeneous software and hardware systems that should provide a certain level of convenience and reliability of management appropriate to the field of their application. Depending on the situation, certain mechanisms of a human-machine interaction may be preferable. In case of a parallel operation of several data channels (human-machine interaction mechanisms) at the output of which the solutions differ from each other, a technique to make the only correct decision for each control event of a mobile robotic system (MRS) is required. This method should be adaptive to a current situation, the peculiarities of a particular operator ensuring the MRS safety in any situation.

*Corresponding author. E-mail address: aleksgridnev@gmail.com
†Corresponding author. E-mail address: snaipervti@gmail.com
‡Corresponding author. E-mail address: evchepin@gmail.com

2 Related Works

There are some possible probabilistic approaches for the decision-making: Naive Bayesian classifier described by Leung [1], hidden Markov model researched by Rabiner and Juang [2], graph probability model described by Zeng and Liu [3]. However, to specify the internal parameters of these models we have only the statistics of the data channels accuracy. Therefore, significant improvements are required in the field of methodology of application of these models in practice for the solution of the task of the MRS control using several data channels, for example, shown by Shachter and Peot [4].

For the decision making, a machine learning described by Mitchell [5] or neural networks researched by Demuth et al. [6] can be used. However, the data amount that should be given on the input of the machine learning or the neural network is much greater than we can get during the control testing using several data channels.

None of the considered models take into account the fact of the execution of the commands that require the same resources to perform as this particular command. Therefore, we developed our own system described by Dyumin et al. [7] that allows us to take into consideration other commands when deciding on the execution of an incoming command.

3 Theory

3.1 Parameters

Command is an elementary instruction coming from an operator. The commands come from different data channels and may or may not require the same time resources at runtime. The commands that require the same time resources at runtime are called competing. The set of mutually competing commands is called the command-space group and the set of all groups forms the command space.

When deciding on the command performance the operator's emotions are taken into account. At the moment of the command execution, the operator has a certain set of emotions some of which show an interest in the command performance, the other part – a boredom and indifference to the command execution as shown by Chepin et al. [8]. Therefore, for decision-making the parameter minimum proportion of non-negative emotions is introduced. It is set during the training process and is obtained using the brain-computer interface as considered by Voznenko et al. [9].

In order to determine for how long shall we execute the command and how to resolve the conflicts of incoming competing commands from several data channels, the parameters execution time and command priority are inserted. The command priority is set taking into account the specifics of the commands execution and the accuracy of the data channels.

3.2 Variables

There are variables required for the decision-making.

1. Time ($time$) is a time counter that is used to account the execution time of the commands.
2. Background end time ($backgroundEndTime$) is the time of the command end at the channel input that is used to account which commands from which channels came.
3. Foreground end time ($foregroundEndTime$) is the time of the command end at the output from the decision-making system that is used to determine the commands for execution.

3.3 The Algorithm

The algorithm of actions for the commands reception is needed to update the DMS variables and should be called when the command comes at the DMS input (see the 1 algorithm).

Algorithm 1: Activate command from channel

Data: channels, commands, minEmotion, executionTime, commandPriority, time, endTimeBackground, endTimeForeground
Input: channel, command, emotion
Result: endTimeForeground is recalculated

```
if emotion < minEmotion[channel][command] then
    return
end
endTimeBackground[channel][command] ← time + executionTime[channel][command]
if endTimeForeground[command] > time then
    endTimeForeground[command] ←
      max(endTimeForeground[command], time + executionTime[channel][command])
    return
end
for j ← 0 to commands do
    commandPriority[j] ← 0
    for i ← 0 to channels do
        if endTimeBackground[i][j] > time then
            commandPriority[j] ← commandPriority[j] + priority[i][j]
        end
    end
end
found ← false
foreach group ∈ groups do
    if command ∈ group then
        foreach j ∈ group do
            if endTimeForeground[j] > time then
                if commandPriority[j] > commandPriority[command] then
                    found ← true
                else
                    endTimeForeground[j] ← time
                end
            end
        end
    end
end
if ¬found then
    endTimeForeground[command] ← time + executionTime[channel][command]
end
```

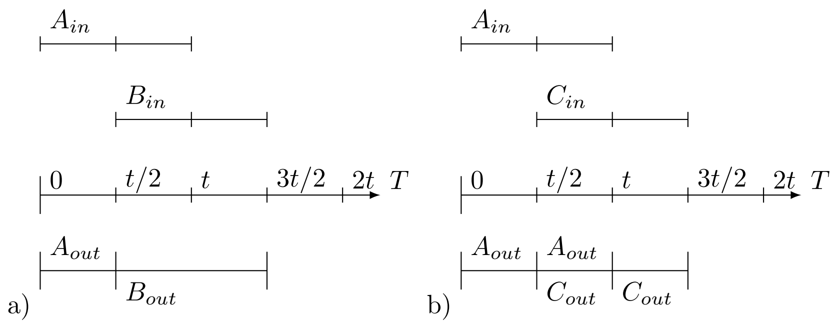

Figure 1: The time diagrams for the examples (a – for 4.1, b – for 4.2)

4 Proof of Concept

In order to proof the DMS operation we need to consider some simple special cases with one and two data channels with various parameters.

4.1 Competing Commands Come from One Data Channel

Let us consider a situation when two competing commands come from the same channel and each command has the same execution time equal to t. Let the priority of these commands be the same. The command A comes first; then after a time $t/2$ the command B comes (Figure 1a). According to the 1 algorithm the command A will be executed during the time $t/2$ and then – the command B with the time t.

Thus, in the DMS the logic is implemented in which the response to the incoming commands occurs in real time (RT). Let us suppose that at the beginning the operator gave the command forward for execution and then realized that they needed to go backward. With this implementation, the command backward will be executed immediately after it comes which is the optimal solution for the task of controlling the MRS in RT.

4.2 Non-Competing Commands Come from One Data Channel

Let us consider an example where the mutually non-competing commands come from the same data channel. The command A comes first; then after a time $t/2$ the command C comes (Figure1b). Let the duration of their execution be the same and equal to t. Since these commands do not require the same resources to be performed, then from the time $t/2$ they will be executed simultaneously.

4.3 Competing Commands Come from Two Data Channels

Let us consider a situation when two competing commands come from two data channels: 1 and 2. Let the priority of the commands from the channel 1 be equal to P_1 and the priority of the commands from the channel 2 – to P_2 ($P_2 < P_1$). Let us consider the example shown earlier in the paragraph 4.1 (the command A comes first; then after a time $t/2$ the command B comes) with two different cases.

In the first case, the command A came from the channel 1 and the command B – from the 2. At time $(0; t/2)$, the command A will be executed. At time $(t/2; t)$, the command A will

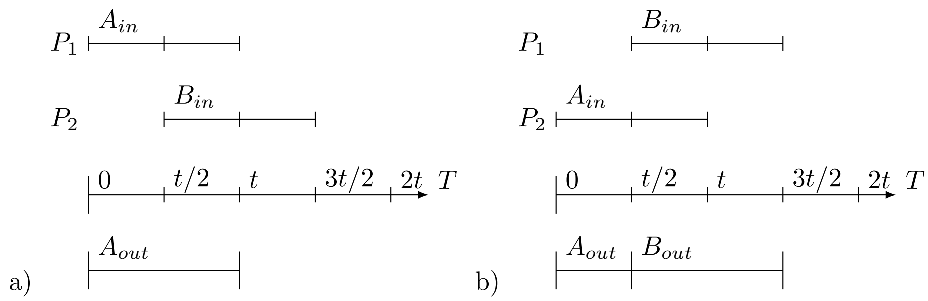

Figure 2: The time diagrams for the examples 4.3 (a – the case 1, b – the case 2)

be executed, since the priority of the incoming command B is lower than ($P_2 < P_1$). At the remaining time nothing will be executed as it runs in the background (Figure 2a).

In the second case, the command A came from the channel 2 and the command B – from the 1. Then at time $(0; t/2)$, the command A will be executed and at time $(t/2; t)$ – the command B, since the priority of the command B is higher than ($P_1 > P_2$) (Figure 2b).

Since the priority parameter takes into account the channel accuracy, then in case of mutually competing commands, the one with the higher priority will be performed first.

5 Experimental Verification

To test the effectiveness of the decision-making system, a control agent model that manages the device using several control channels was developed. The emotions consideration was not performed as their correct generation is a difficult task. For modeling 5 commands were taken that can be given from 4 data channels. The choice of the command, channel, as well as errors generation in the channels operation is carried out using the pseudorandom number generator. The duration of the DMS testing was 10^7 ms.

During the testing the following parameters were measured:

- True Positive – how many times the called command was executed.
- False Positive – how many times the non-called command was executed.
- False Negative – how many times the called command was not executed.

After the DMS testing, the testing with duplicating the input commands to the output and using the developed decision-making system was conducted. The testing results are presented in Table 1.

Result	Duplicating	DMS
True Positive	4482	2776
False Positive	32220	7616
False Negative	107	1350

Table 1: Test result

As a result, it was found out that with a significant reduction of False Positive (by 76%), the number of correct True Positive (by 38%) was not reduced greatly. Thus, it was shown that the DMS works more efficiently than duplicating the input commands to the output for the MRS control using several data channels.

6 Conclusion

Thus, we developed the DMS which decides on the execution of the commands from the different data channels while taking into account the specificity of their work with the help of the adjustable parameters that are obtained from the statistics of the data channels operation (accuracy of work, runtime of the command), and the specifics of the commands (command priority). The DMS supports the groups of commands that allow performing several mutually non-competing commands simultaneously. The verification by the main special cases showed the suitability of this model and the testing on the model confirmed the effectiveness. In the real world, this development is already used in the robotized wheelchair project being its main component.

Acknowledgments

The funding for this research was partially provided by the Russian Foundation for Basic Research (Grant RFBR #14-07-00843) and by the Competitiveness Growth Program of the National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

This work was supported by Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Professional Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

References

[1] K. M. Leung. Naive bayesian classifier. *Polytechnic University Department of Computer Science/-Finance and Risk Engineering*, 2007.

[2] L. Rabiner and B. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[3] J. Zeng and Z. Liu. Probabilistic graphical models. In *Type-2 Fuzzy Graphical Models for Pattern Recognition*, pages 9–16. Springer, 2015.

[4] R. D. Shachter and M. A. Peot. Decision making using probabilistic inference methods. In *Proceedings of the Eighth international conference on Uncertainty in artificial intelligence*, pages 276–283. Morgan Kaufmann Publishers Inc., 1992.

[5] T. M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[6] H. B. Demuth, M. H. Beale, O. De Jesús, and M. T. Hagan. *Neural network design*. Martin Hagan, 2014.

[7] A. A. Dyumin, P. S. Sorokoumov, E. V. Chepin, and G. A. Urvanov. Architecture and prototype of human-machine interface with mobile robotic device. *Vestnik natsional'nogo issledovatel'skogo yadernogo universiteta "MIFI"*, 2(3):376–380, 2013.

[8] E. V. Chepin, A. A. Dyumin, G. A. Urvanov, and T. I. Voznenko. The improved method for robotic devices control with operator's emotions detection. In *NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), 2016 IEEE*, pages 173–176. IEEE, 2016.

[9] T. I. Voznenko, G. A. Urvanov, A. A. Dyumin, S. V. Andrianova, and E. V. Chepin. The research of emotional state influence on quality of a brain-computer interface usage. *Procedia Computer Science*, 88:391–396, 2016.