Engineering Software for a Mobile Robot Motion Control System

Igor Ryadchikov¹, Evgeny Nikulchev², Alexander Gusev¹, Semyon Sechenev¹ and Alexey Prutskiy¹

¹ Laboratory of Robotics and Mechatronics, Kuban State University, Krasnodar, Russian Federation
² Department of Control and Modelling Systems, MIREA – Russian Technological University, Moscow, Russian Federation

E-mail: alexandrgsv@gmail.com

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Abstract

The paper centres round the problem of the engineering a motion control system for a mobile robot based on the effective selection of software components with respect to the numerical criterion proposed by the authors. The data for the selection process comes out the reproducible experiments with the sets of alternative components in a Gazebo virtual infrastructure simulating the real robot operating conditions. The genetic algorithm is used to reduce the number of experiments with unpromising sets of software components. The methodology proposed by the authors is applied to the real task of engineering a motion control system for a non-anthropomorphic mobile robot. The virtual infrastructure and genetic algorithm parameters are provided as well as the physical model of the robot for that task. To calculate the integral quality criterion proposed in the paper, 4 partial quality criteria were measured in the experiments with different software components. The motion process of the physical robot with the selected software components is shown.

Keywords: software engineering, quality of software, mobile robot, control system, control moment gyroscope, genetic algorithm

1. Introduction

The effective choice of components for a motion control system of a mobile robot is becoming an important problem of software engineering as the number of components for popular robotics frameworks is increasing [1]. For example, for the Robotic Operating System (ROS) framework there are now thousands of components available. Some of them provide the similar functionality. The interaction of the components in a software stack may lead to an unexpected behavior [2], for example if one of the components blocks the others from accessing a common resource (database, hardware driver, LAN, etc.). Thus, it is important to choose the software components with respect to the quality of their joint functioning in a stack. When developing a real-time motion control system for a mobile robot it is also important to assess the software components in the conditions most similar to the real robot operating conditions. To ensure that the experimental stand for reproducible experiments should be designed. To guarantee the reproducibility of the experiments and to spare the mechanical parts of a real robot it is appropriate to implement the experimental stand in the virtual infrastructure, simulating the operating conditions of the robot using the simulation software like Gazebo [3]. Evolutionary algorithms [4], above all the genetic algorithm [5], can be used to reduce the number of experiments with unpromising software components.
The paper consists of 5 sections: Introduction, Formulation of the problem, Methodology, Solution and Concluding remarks.

2. Formulation of the problem

Let us consider the set of functional features (see table 1) which should be implemented in the motion control system of the robot. For each feature there is a set of alternative software components capable of implementing that feature.

Table 1. List of functional features and components.

| Name of the functional feature | Alternative components implementing that functional feature | Description |
|-------------------------------|------------------------------------------------------------|--------------|
| Servo                         | Dynamixel mx106 driver, Rd50 driver, Rd60 driver, Speedgoat+stm32, Raspberry(Python) | Interaction of the control system with the servomotor of a mobile robot |
| Control                       | Stm32, Stm32+Raspberry, Mpu9250 | Control of the robot movement with a software module implemented for the chosen architecture of the computer |
| Gyro                          | Gyrolab GL109, Gyrolab GLVG 203 | Getting information from a gyroscope |

Thus, there are 36 subsets of software components (stacks) in which for every functional feature there is at least one implementing component. The stacks are to be evaluated in the virtual infrastructure, simulating the operating conditions of the robot (see figure 1). Let us denote the stacks as \( s_j, j = 1,36 \).

![Figure 1. The effective choice of software components.](image)

Table 2 shows the partial quality criteria used to estimate the quality of functioning of the robot motion control system.

Table 2. Partial quality criteria.

| Notation | Criterion | Unit |
|----------|-----------|------|
| \( r_1 \) | Response time of the control system | \( \mu s \) |
| \( r_2 \) | Mass of the robot including the additional equipment to run the selected software components | kg |
| \( r_3 \) | Total mass of additional batteries to ensure autonomous operation of the mobile robot when using the selected software components | kg |
| \( r_4 \) | The sum of the squares of the deviations of the robot trajectory from the calculated trajectory for all experimental samples | m^2 |

Let us define the integral quality criterion of a stack \( s^j \) as:

\[
\Psi(s^j) = \sum_{\xi=1}^{4} w_\xi \bar{r}_\xi^j, \tag{1}
\]

where \( \bar{r}_\xi^j \) are the normalized values of partial quality criteria \( r_\xi^j; \xi = 1,4; \) \( w_\xi \) are the weights of the partial criteria:

\( w_1 = 0.18; w_2 = w_3 = 0.16; w_4 = 0.5. \)

The virtual infrastructure, simulating the operating conditions of the robot is made up with the Gazebo robotics simulator. The configuration of the virtual infrastructure is presented in table 3.

Table 3. Virtual infrastructure configuration.

| No | Parameter                              | Value               |
|----|----------------------------------------|---------------------|
| 1  | Physics type                           | ode                 |
| 2  | Solver type                            | World               |
| 3  | Max step size                          | 0.001               |
| 4  | Gravity                               | [0.0 0.0 – 9.81]    |
| 5  | Surface contact ode max_vel            | 10                  |
| 6  | Surface contact ode min_depth          | 0.001               |
| 7  | Collision geometry sphere radius      | 0.5                 |
| 8  | Surface contact ode kp                 | 1e15                |
| 9  | Surface contact ode kd                 | 1e13                |

The task is to choose the stack \( s^* \) satisfying the following condition:

\[ s^* = \arg\min_{s^j L^{1,36}} \Psi(s^j). \]

3. Methodology

To organize a reproducible experiment for evaluation the quality of functioning, a physical model of the robot [6, 7] was assembled (see figure 2) and an experimental stand in the virtual infrastructure was created.
Figure 2. Physical model of the robot.

The automated methodology for selecting an effective set of software components involves using a genetic algorithm to generate and experimentally evaluate stacks of components. Genetic algorithm is a global optimization algorithm based on evolutionary computations. It uses the mechanisms of random selection, combination, and variation of the desired parameters similar to the natural selection.

The motion control process for the experimental stand in the virtual infrastructure and for the physical model of the robot is shown in the figure 3. The genetic search configuration is shown in table 4.

![Motion planning](image)

**Figure 3.** Robot motion control process.

**Table 4.** Genetic search configuration.

| No | Parameter                  | Value        |
|----|----------------------------|--------------|
| 1  | Selection operator         | Tournament selection |
| 2  | Mutation operator          | Extended power mutation |
| 3  | Crossover operator         | Laplace crossover |
| 4  | Probability of mutation    | 0.01         |
| 5  | Probability of crossover   | 0.8          |
| 6  | Elite count                | 3            |
| 7  | Population size            | 4            |
| 8  | Maximum generations        | 100          |
| 9  | Maximum stall generations  | 5            |
| 10 | Function tolerance         | 0.01         |

4. Solution

Using ga toolkit from the Global Optimization Toolbox of the Matlab R2018a the choice of software components minimizing the integral criterion (1) was made. The graph of evolutionary selection is shown in the figure 4. The motion of the robot is depicted in the figure 5.

![Graph of evolutionary selection](image)

**Figure 4.** Graph of evolutionary selection of a stack of components. The penalty value refers to the value of integral criterion $\Psi$ for a given stack.

It should be noted that a small “oscillation” in the evolutionary selection graph is explained by the unavoidable measurement noise. However, the genotype of the best solution from generation to generation begins to prevail more and more in the population (this is seen by the decrease in the average value of $\Psi$) and the identification of the best solution occurs because the number of experiments falling on the best genotype is increasing, which eliminates random factors in assessing this genotype.

![The motion of the robot](image)

**Figure 5.** The motion of the robot.

For the considered example, the selected effective solution $s^*$ (genotype $= [2 \ 1 \ 2]$) is as follows:

- Servo feature is implemented with the “Rd50 Driver” component;
- Control feature is implemented with the “Speedgoat + stm32” component;
- Gyro feature is implemented with the “Gyrolab GL109” component.
5. Concluding remarks

The results of the practical implementation of the methodology for the experimental selection of software components demonstrate the applicability and the effectiveness of the proposed approach in engineering software for motion control systems of mobile robots. Further research will be aimed at improving the methodology in terms of increasing the convergence rate of the evolutionary algorithm and considering larger sets of partial quality criteria.

Acknowledgements

The work was carried out within the framework of the state task of the Ministry of Science and Higher Education, project No. 8.2321.2017/4.6 “Development and adaptation of control systems for compensation of dynamic deflecting effects on mobile objects in a state of dynamic equilibrium”.

References

[1] Romanov A M 2019 A review on control systems hardware and software for robots of various scale and purpose Russian Technological Journal 7(5) 30–46 (doi: 10.32362/2500-316X-2019-7-5-30-46)
[2] Wienke J, Wigand D, Koster N and Wrede S 2018 Proc Second IEEE International Conference on Robotics Computing (Laguna Hills, CA, USA) (Hoboken, New Jersey: I.E.E.E. Press) pp 25–32
[3] Koenig N P and Howard A 2004 Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Sendai, Japan) (Hoboken, New Jersey: I.E.E.E. Press) 3 pp 2149–54
[4] Xiaohu Y, Fazhi He, Neng H and Haojun A 2018 International Journal of Cooperative Information Systems 27(1) 1741001
[5] Pate B K, Parhi D R K, Jagadeesh A and Kashyap S K 2018 Computers & Electrical Engineering 67 708
[6] Ryadchikov I, Sechenev S, Sinitsa S,... and Nikulchev E 2017 International Journal of Advanced Computer Science and Applications 8(9) 29–34
[7] Ryadchikov I., Sechenev S., Nikulchev E ... and Vishnykov R 2018 International Review of Automatic Control, 11(4) 160–5