IOP Conference Series: Materials Science and Engineering

Genetic Algorithm and Simulation for Selecting PID-Controllers Parameters in the Walking Robot

Genetic Algorithm and Simulation for Selecting PID-Controllers Parameters in the Walking Robot

I V Ryadchikov[1], A A Gusev[1,*], S I Sechenev[1], A S Prutskiy[1] and E Nikulchev[2]

[1]Kuban State University, Krasnodar 350040, Russia
[2]MIREA – Russian Technological University, Moscow 119454, Russia

E-mail: alexandrgsv@gmail.com, nikulchev@mail.ru

Abstract. Walking robot dynamic stabilization is a growing field of research as such robots are being constructed for various applications in human environment. The dynamic stabilization systems for such robots are often highly non-linear with the additional demand for rapid change of model mass parameters to meet the design requirements. The article centres round the problem of mechanical modelling of a walking robot and identifying the robot dynamic stabilization system parameters using the genetic algorithm approach. The stabilization system supposed in the paper is based on inverted pendulum model with paired gyroscopic units with limited kinetic momenta, that shapes out the optimization problem. The proposed control system includes a cascade of two PID regulators, the optimal parameters of which are identified using the genetic algorithm approach. The authors give the short description of the approach and suggest the fitness function suitable for this application. The computational experiment is then conducted, and the optimal parameters set is obtained. The disadvantages of the genetic algorithm solver in Simulink environment are pointed out with the approaches to their partial elimination.

1. Introduction

PID control tuning is a fairly well-studied linear programming problem [1–10]. Among the wide variety of global optimization methods [11–13] in setting up PID regulators, genetic algorithms [14–16] have received comparatively more widespread use to date, which is explained by the comparative simplicity and universality of the problem statement for genetic search , a relatively high degree of protection from “getting stuck” in the local minimum for this method and the presence of an embedded solver of genetic algorithms in lots of mathematical software packages. Nevertheless, it is necessary to note the difficulties in working with genetic algorithms. It is not always easy to translate the target requirements for a non-linear system into a formalism of a fitness function of a genetic algorithm. Additional restrictions on the parameter search area are often necessary in order to avoid obtaining solutions that lead to solver errors when simulating. Genetic search is a resource-intensive process. However, it can be speeded up using parallel computing on multi-core or multi-processor systems. These contradictions create a demand for case studying the applicability of the genetic algorithm approach to specific tasks, constructing suitable fitness functions, and describing and eliminating the difficulties encountered in the application of genetic algorithms.

This paper presents the application of a genetic algorithm to the task of tuning a cascade of two PID controllers for the walking robot body deviation compensation system. To simulate the dynamics of the robot, the model of the inverse pendulum was applied as one of the simple and at the same time adequate models [17]. Despite the above-mentioned common difficulties of using genetic algorithms, the proposed fitness function makes it possible to find an acceptable control after 2 hours of parallel evolution on a 4-core Intel (R) Core (TM) i7-7700 microprocessor with a clock frequency of 3601 MHz and 12 GB of RAM for storing populations of parameters.

2. Formulation of the problem

The mechanical model of the walking robot consists of a beam "a" (see figure 1) of dimensions 0.02 x 0.02×2.5 m, the mass of which is negligible. The cube "body" with the side of 2 m is fixed on the beam. The center of the cube is at the height of 2.5 m (at the end of the beam). The mass of the cube is 3520 kg. Through the center of the cube passes the beam "b" perpendicular to the beam "a". Gyroscopic units GU1, GU2 are attached at the both ends of the beam "b". The maximum kinetic momentum of each of the gyroscopic units does not exceed 9000 N * m * s.

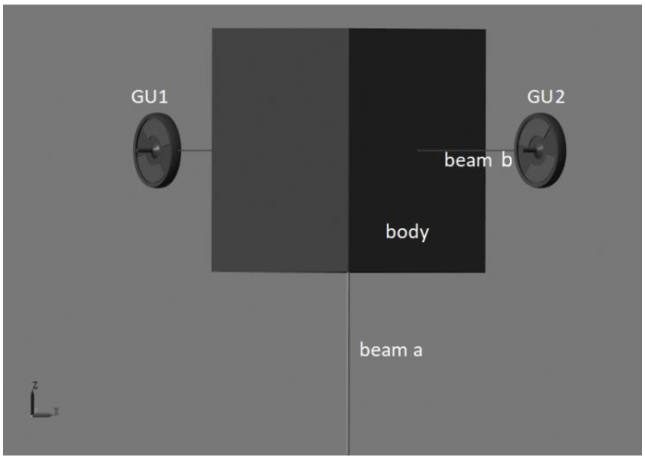

Figure 1. The mechanical model of the walking robot

When the body deviates perpendicularly to the beam "b" by 10°, it is necessary to find the optimal parameters of the PID controller cascade (see figure 2), which control the speed of antisymmetric rotation of gyroscopic units to return the robot body to the equilibrium position. For definiteness, the deviation occurs at the time t = 0. The work of friction forces is not considered.

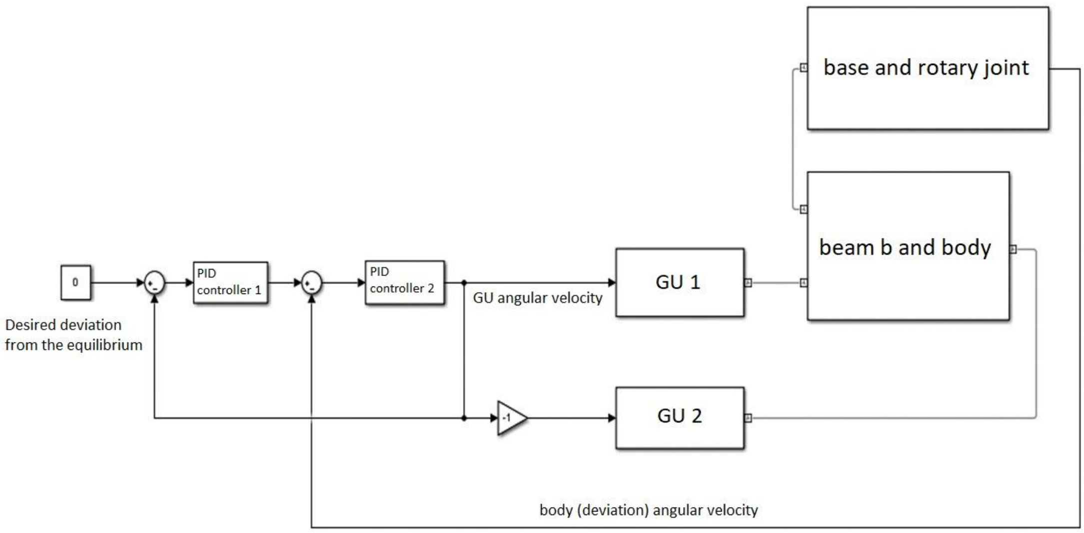

Figure 2. The proposed block diagram of the dynamic stabilization control system.

3. Method

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 [18]. Problem formalization is carried out in the form of a genotype i.e. a vector of "genes", where each "gene" may be a bit, a number or another object. A set of genotypes of the initial population is randomly created. Each genotype receives a specific value of the fitness function. The form of the fitness function is specified by the researcher. The best genotypes in the population are selected based on their fitness values. After that the genetic operators i.e. crossing (crossover) and mutation are applied to the best genotypes, which results in a new generation of solutions for which the above-described procedure is repeated [19]. The process stops after the desired number of generations have evolved, when the allowed computational time is over, or if the best fitness does not change for a specific number of consecutive generations [20].

It should be noted that evolution may go both in the direction of increasing fitness values and in the direction of their minimization. The choice of direction depends on the software settings. The search direction may be easily inversed with the sign of the fitness function. In this paper the genetic search went towards minimizing the fitness function.

4. Solution

Using *ga* toolkit from the Global Optimization Toolbox of the Matlab R2018a the search for solution of the problem of minimizing the fitness function f was carried out. The fitness function f reflected both that robot body deviation angle is non-zero, N and that it is decreasing non-monotonously, M :

$$f = N + M \text{ ,}$$

where

$$N = \sum w_1 \cdot |\varphi_i|$$

$$M = R \cdot w_2 \text{,}$$

where R is the number of cases when $\varphi_i > \varphi_j$ provided $i > j$, φ_i is the body deviation angle for the time moment t_i, w_1, w_2 are the weight factors both equal to 100 in this study.

Optimal parameters (table 1) were found after 50 generations of parameters selection. Corresponding minimum fitness value is 2571.77. The best value of the fitness function for a randomly generated initial population of solutions was 3200.

Table 1. PID parameters found.

Controller	P	I	D	N (filter coefficient)
PID controller 1	-0.1628	-0.0439	0.0650	2.3974
PID controller 2	99.8108	143.6955	0.9534	40.5466

The dynamics of the robot body with the found parameters of the PID controller cascade is shown in figure 3.

To increase the speed of the genetic algorithm, parallel computations were used on 4 executors using the MATLAB Parallel Computing Toolbox package.

The disadvantage of the implementation of genetic algorithms in the Matlab / Simulink environment is that the solver may stuck on the parameter set corresponding to an unlimited increase in auto-oscillations in the system. The exact calculation of such oscillations requires a significant reduction in the solver time step. In some cases, the accuracy of calculations specified in the solver settings cannot be achieved with the minimum specified step length leading to the crash of the entire genetic algorithm.

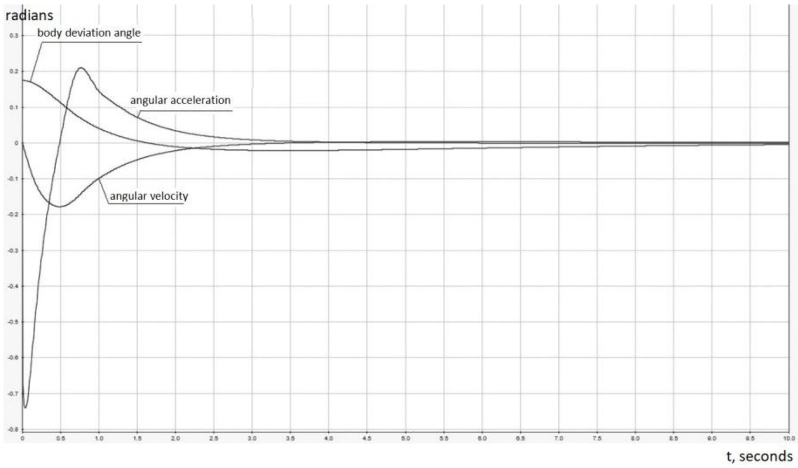

Figure 3. Robot body deviation compensation using the genetically found PID parameters.

We found two ways to deal with this drawback. The first solution is restricting the search for optimal parameters using the *InitialPopulationRange* property of the *ga* solver. However, the application of this approach can lead to finding a local minimum due to the established restrictions. In addition, it is not always possible to a priori estimate the boundaries of parameter sets that could potentially lead to difficulties for the solver.

The second way is to handle the *DiagnosticError* exception by calling the *ExecutionInfo.StopEvent* property of the metadata class of the current model simulation. In the case of such an exception, the value of the fitness function can be artificially set sufficiently large so that the genetic algorithm considers such a set of parameters undesirable and leaves the area where occur the difficulties.

5. Conclusion

The paper demonstrates the application of genetic algorithm approach for selecting the PID-controller coefficients for the dynamic stabilization system of a walking robot. A gyroscopic-based model of the robot dynamic stabilization system is proposed. The type of the fitness function to optimize the PID coefficients using a genetic algorithm is suggested. The results of the simulation of the stabilization process of the robot body using the coefficients found are presented. The results are optimal for the specified parameters of the mechanical model of the robot [21–23]. When changing the model configuration, the genetic search must be repeated using the same fitness function. Further research will be aimed at improving the dynamic stabilization system presented in the article using genetic algorithms.

Acknowledgments

The work was carried out within the framework of the state task of the Ministry of Education and Science, 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] Lan S and Woo P Y 1992 *Proc. IEEE Int. Conf. on Systems Engineering (Kobe, Japan)* (Hoboken, New Jersey: I.E.E.E. Press) pp 444-447
[2] Datar R, Gupta S and Langde R 2015 *IJISET* **2**(4) 948
[3] Dolezel P and Taufer I 2012 *Acta Montanistica Slovaca* **17**(3) 224
[4] Soltesz K, Mercader P and Baños A 2017 *Int. J. Robust Nonlin* **27**(11) 1857
[5] Cetin M and Iplikci S 2015 *ISA Trans.* **58** 292
[6] Yamamoto T, Fujii K and Kaneda M 1998 *IFAC Proc.* **31**(22) 59

[7] Lee C and Chen R 2015 *Sensors* **15** 11685
[8] He Y, Guo S, Shi L, Xing H, Chen Z and Su S 2017 *Proc. IEEE ICMA (Takamatsu, Japan)* (Hoboken, New Jersey: I.E.E.E. Press) p 632
[9] Niewiara L and Zawirski K 2013 *Comp. Appl. Electr. Eng.* **11** 230
[10] Güzelkaya M, Eksin I and Yeşil E 2003 *Eng. Appl. Artif. Intel.* **16** 227
[11] Coelho L S 2009 *Chaos, Solitons & Fractals* **39** 1504
[12] Ozana S and Docekal T 2016 *J. Electrical Engineering* **67** 160
[13] Carotenuto L, Pugliese P and Sergeyev Y 2006 *Control and Intelligent Systems* **34** 225
[14] Mirzal A, Shinichiro Y and Masashi 2006 *ISTECS Journal* **8** 34
[15] Sadasivarao M V and Chidambaram M 2006 *J. Indian Inst. Sci.* **86** 3434
[16] Jayachitra A and Vinodha R 2014 *Advances in Artificial Intelligence* **2014** 791230
[17] Hemami H and Golliday C L 1977 *Mathematical Biosciences* **34** 95
[18] McCall J 2005 *J. Comput. Appl. Math.* **184** 205
[19] Samakwong T and Wudhichai A 2016 *Procedia Computer Science* **86** 91
[20] Haupt R L and Haupt S E 2004 *Practical Genetic Algorithms* (John Wiley & Sons Inc.)
[21] Ryadchikov I, Sechenev S, ... and Nikulchev E 2017 *International Journal of Advanced Computer Science and Applications* **8**(9) 29
[22] Ryadchikov I, Nikulchev E, … and Feshin A 2018 *IOP Conference Series: Materials Science and Engineering* **312** 012021
[23] Ryadchikov I, Sechenev S, … and Vishnyakov R 2018 *Review of Automatic Control* **11**(4) 160