Evolutionary Computation for Synthesis of Control System for Group of Robots and Optimum Choice of Trajectories for their Movement

Askhat I. Diveev
Federal Research Center
"Computer Science and Control"
of Russian Academy of Sciences
Vavilova, 44
119333 Moscow, Russia
aidiveev@mail.ru

Elizaveta Yu. Shmalko
Federal Research Center
"Computer Science and Control"
of Russian Academy of Sciences
Vavilova, 44
119333 Moscow, Russia
e.shmalko@gmail.com

Abstract

Challenges in control of groups of robots arise from dynamic constraints which assure the absence of collisions between robots. To solve the problem of optimal control for the group of robots we use a two-stage method of synthesis. At the first stage we solve a problem of stabilization of each robot in some point of the state space. At the second stage we search for optimal moving trajectories as sets of points of the state space at which each robot is stabilized. Optimization criteria contain conditions of absence of collisions and other phase constraints. For stabilization we use a symbolic regression method. At the second stage to search for optimal trajectories we use various evolutionary and gradient algorithms of nonlinear programming. In the example, we considered a group of three mobile robots. To search for points of optimal trajectories we used four methods: the genetic algorithm, the particle swarm optimization, the fast gradient descent and the bees algorithm.

1 Introduction

The classical approach to solving the applied problem of optimal control for a mobile robot is to solve successively two problems. The first one is a control synthesis problem or ensuring stability to the robot relative to some point of the state space. The second one is the parametric optimal control problem, which consists in finding points of the state space relative to which the control system, synthesized at the first stage, should ensure stability. And sequential switching of the detected points in the state space ensures the movement of the robot from the initial condition to the terminal one, taking into account the phase constraints and the optimal value of the given quality criterion.

The same approach we apply to the control of a group of robots. When solving problems at both stages, we use evolutionary computation methods, for the synthesis problem - the symbolic regression, for the problem of optimal parametric control - evolutionary algorithms.

In: Yu. G. Evtushenko, M. Yu. Khachay, O. V. Khamisov, Yu. A. Kochetov, V.U. Malkova, M.A. Posypkin (eds.): Proceedings of the OPTIMA-2017 Conference, Petrovac, Montenegro, 02-Oct-2017, published at http://ceur-ws.org

2 Problem Statement of Optimal Control of Group of Robots

Consider the problem of optimal control for a group of robots.

Given models of the plants

$$\dot{\mathbf{x}}^j = \mathbf{f}^j(\mathbf{x}^j, \mathbf{u}^j), \tag{1}$$

where $\mathbf{x}^j \in \mathbb{R}^{n_j}$, $\mathbf{x}^j = [x_1^j \dots x_{n_j}^j]^T$ is a state vector of the robot j, $\mathbf{u}^j \in \mathbf{U}_j \subseteq \mathbb{R}^{m_j}$, $\mathbf{u}^j = [u_1^j \dots u_{m_j}^j]^T$ is a control vector of the robot j, $\mathbf{U}_j$ is a compact set, $j = 1,\dots,N$, N is a number of robots.

Given initial conditions

$$\mathbf{x}^j(0) = \mathbf{x}^{0,j},\ j = 1, \dots, N. \tag{2}$$

Given terminal conditions

$$\varphi_k^j(\mathbf{x}^j(t_f)) = 0,\ k = 1, \dots, l_j,\ l_j \le n_j, \tag{3}$$

where

$$t_f = \begin{cases} t, & \text{if } \ t < t^+ \text{ and } \varphi_k^j(\mathbf{x}^j(t)) = 0,\ k = 1, \dots, l_j,\ j = 1, \dots N; \\ t^+, & \text{if } t = t^+, \end{cases} \tag{4}$$

t^+ is a given time limit of control.

Given static phase constraints

$$\alpha_k(\mathbf{x}^j(t)) \le 0,\ k = 1, \dots, r. \tag{5}$$

To define dynamic phase constraints, we introduce in the set of robot numbers $I = \{1, 2, \dots, N\}$ a set of pairs

$$V = ((i_1, j_1), (i_2, j_2), \dots, (i_W, j_W)), \tag{6}$$

where $i_k, j_k \in I$, $i_k \ne j_k$.

Then the dynamic phase constraints determining the conditions of closure of pairs of objects are given by

$$\beta_r(\mathbf{x}^{i_k}(t), \mathbf{x}^{j_k}(t)) \le 0,\ k = 1, \dots, W,\ r = 1, \dots, R, \tag{7}$$

where W is the number of combinations of N elements taken 2 at a time, $W = N!/(2!(N-1)!) = N(N-1)/2$.

The dynamic phase constraints poses the property of vector commutativity

$$\beta_r(\mathbf{x}^{i_k}(t), \mathbf{x}^{j_k}(t)) = \beta_r(\mathbf{x}^{j_k}(t), \mathbf{x}^{i_k}(t)).$$

Given a quality functional

$$J = \int_0^{t_f} f_0(\mathbf{x}^1, \dots, \mathbf{x}^N, \mathbf{u}^1, \dots, \mathbf{u}^N)\, dt \to 0, \tag{8}$$

It is necessary to find a control function that ensures the movement of robots from the given initial states (2) to the terminal position (3) without violating the phase constraints (5), (7) with the optimal value of the quality criterion (8).

At the first stage we solve the problem of stabilization

$$\mathbf{u}^j = \mathbf{g}^j(\widetilde{\mathbf{x}}^j - \mathbf{x}^j), \tag{9}$$

where $\widetilde{\mathbf{x}}^j$ is some point of the state space $\mathbb{R}^{n_j}$.

To solve the problem, we use one of the methods of symbolic regression [Diveev, 2015a].

At the second stage we find the set of points of the state space and the parameters of switching between them

$$\widetilde{\mathbf{X}}^j = (\widetilde{\mathbf{x}}^{j,1}, \widetilde{\mathbf{x}}^{j,2}, \dots, \widetilde{\mathbf{x}}^{j,K}, \varepsilon_j),\ j = 1, \dots, N. \tag{10}$$

The points found (10) are stabilization points of robots

$$\mathbf{u}^j = \mathbf{g}^j(\widetilde{\mathbf{x}}^{j,p} - \mathbf{x}^j), \tag{11}$$

where the index p increases its value over given intervals of time

$$p \leftarrow p + (1 - \vartheta(t_j - t)),\ t_j = \leftarrow t_j + \delta t(1 - \vartheta(t_j - t)),$$

$t_1 = \delta t$, δt is a given interval of time,

$$\vartheta(A) = \begin{cases} 1, & \text{if} \quad A > 0\ ; \\ 0, & \text{otherwise.} \end{cases}$$

Coordinates of stabilization points are searched simultaneously for all robots, adding penalties to the target function for violation of phase constraints

$$\widetilde{J} = J + \omega_1 h_1 + \omega_2 h_2, \tag{12}$$

where

$$h_1 = \int_0^{t_f} \sum_{k=1}^{r} \sum_{j=1}^{N} \vartheta(\alpha_k(\mathbf{x}^i(t)))dt,\ h_2 = \int_0^{t_f} \sum_{k=1}^{s} \sum_{w=1}^{W} \vartheta(\beta_k(\mathbf{x}^{j_1}(t), \mathbf{x}^{j_2}(t)))dt.$$

3 Method of Variational Complete Binary Genetic Programming for Synthesis of Stabilization System

To solve the problem of control synthesis (9) is finding a multidimensional nonlinear function $\mathbf{u}^j = \mathbf{g}^j(\widetilde{\mathbf{x}}^j - \mathbf{x}^j)$ that ensures the stability of a system of differential equations

$$\mathbf{x}^j = \mathbf{f}^j(\mathbf{x}^j, \mathbf{g}^j(\widetilde{\mathbf{x}}^j - \mathbf{x}^j)) \tag{13}$$

relative to the point of the state space. We use one of the methods of symbolic regression, the method of variational complete binary genetic programming.

To build a code of binary genetic programming we use the following basic sets:

- a set of arguments of mathematical expression

$$F_0 = (q_1, \ldots, q_P, x_1, \ldots, x_N); \tag{14}$$

- a set of functions with one argument

$$F_1 = (f_{1,1}(z), f_{1,2}(z), \ldots, f_{1,R}(z)); \tag{15}$$

- a set of functions with two arguments

$$F_2 = (f_{2,1}(z_1, z_2), \ldots, f_{2,S}(z_1, z_2)); \tag{16}$$

- a set of unit elements for functions with two arguments

$$E_2 = (e_1, \ldots, e_M). \tag{17}$$

A set of functions with one argument must include the identity function

$$f_{1,1}(z) = z. \tag{18}$$

Every function with two arguments from (16) has a unit element from the set (17), $\forall f_{2,i}(z_1, z_2) \in F_2\, \exists e_j\ \in E_2$

$$f_{2,i}(e_j, z_2) = z_2,\ f_{2,i}(z_1, e_j) = z_1, \tag{19}$$

$i \in \{1, \ldots, S\}$, $j \in \{1, \ldots, M\}$.

To generate a code of binary genetic programming we combine the set of arguments (14) of mathematical expression and the set of unit elements (17) into one ordered set

$$F = (f_1 = q_1, \ldots, f_P = q_P, f_{P+1} = x_1, \ldots, f_{P+N} = x_N, f_{P+N+1} = e_1, \ldots, f_{P+N+M} = e_M). \tag{20}$$

We write down mathematical expression in the form of composition of nested functions and arguments of mathematical expression

$$y = f_{\alpha_1}(f_{\alpha_2}(\ldots f_{\alpha_K})\ldots) = f_{\alpha_1} \circ f_{\alpha_2} \circ \ldots \circ f_{\alpha_K}, \tag{21}$$

where $f_{\alpha_i} \in F \cup F_1 \cup F_2$, $i = 1, \ldots, K$.

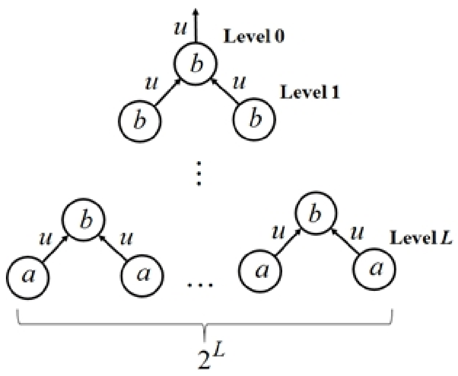

Figure 1: Complete binary computing tree

Define the inmost nesting depth of an element in the composition. Let the inmost depth is equal L. It means that we have binary computing tree with level L. It has 2^L leaves. We write down mathematical expression in the form of a complete binary computing tree (see Fig.1)

In the Fig.1 a is an item number from the set F (20), b is an item number from the set F_2 (16), u is an item number from the set F_1 (15).

Each level includes numbers of elements from the set F_1 and the same numbers of elements from the set F_2, and the last level includes numbers from the set F. If the tree has redundant nodes or edges, then we use a number of any function with two arguments for nodes, a number of identity function for edges and a number of unit element for this function with two arguments on the last level. Code of binary variational genetic programming is an ordered set of numbers of elements from the first, second and other levels of the tree

$$C = (u_1, b_1, u_{2,1}, u_{2,2}, b_{2,1}, b_{2,2}, \ldots, u_{L,1}, u_{L,2}, \ldots, u_{L,2^L}, a_{L,1}, a_{L,2}, \ldots, a_{L,2^L}). \tag{22}$$

Consider an example. Let we have the following mathematical expression

$$y = e^{-qx}\cos(qx + \sin(x)).$$

For this mathematical expression, we have sets

$$F_1 = (f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = e^z, f_{1,4}(z) = \cos(z), f_{1,5}(z) = \sin(z));$$

$$F_2 = (f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2);$$

$$F = (f_1 = x, f_2 = q, f_3 = 0, f_4 = 1).$$

Here $f_3 = 0$ is a unit element for addition and $f_4 = 1$ is a unit element for multiplication.

Write down the mathematical expression in the form of composition of functions from these sets

$$y = f_{1,1}(f_{2,2}(f_{1,3}(f_{1,2}(f_{2,2}(f_1, f_2))), f_{1,4}(f_{2,1}(f_{2,2}(f_1, f_2), f_{1,5}(f_1))))).$$

The nesting depth in our case is three. The code of the mathematical expression is

$$C = (1,\ 2,\ \ 3,4,\ 2,1,\ \ 1,2,1,5,\ 1,1,2,1,\ \ 1,1,1,1,1,1,1,1,\ 2,3,1,3,2,1,1,3).$$

Fig.2 shows complete binary computing tree for the equation. To calculation the mathematical expression by its code, we use an ordered set of real variables with the same cardinal number as the code.

$$Y = (y_1, \ldots, y_{30}).$$

Calculate mathematical expression from the end

$$Y = (y_1, \ldots, y_{14}, q, 0, x, 0, q, x, x, 0,\ \ q, 0, x, 0, q, x, x, 0);$$

$$Y = (y_1, \ldots, y_{10}, q + 0, x + 0, xq, x + 0,\ \ q, 0, x, 0, q, x, x, 0,\ \ q, 0, x, 0, q, x, x, 0);$$

$$Y = (y_1, \ldots, y_6, q, -x, qx, \sin(x),\ \ q, x, xq, x,\ \ q, 0, x, 0, q, x, x, 0, q, 0, x, 0, q, x, x, 0);$$

$$Y = (y_1, y_2, e^{-xq}, \cos(xq + sin(x)), \ldots);$$

$$Y = (e^{-xq}\cos(qx + \sin(x)), e^{-xq}\cos(qx + \sin(x)), \ldots).$$

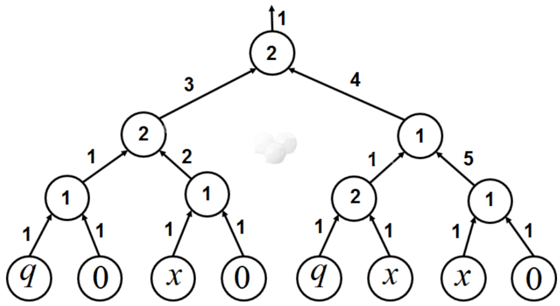

Figure 2: Complete binary computing tree for the mathematical expression

4 Variational Genetic Algorithm to Search for Optimal Solution

To search for optimal solution, we use a variational genetic algorithm. It uses principle of small variations of the basic solution [Diveev, 2015b].

Define a small variation of a code of binary analytic programming as an integer vector of two components

$$\mathbf{w} = [w_1 \ w_2]^T , \tag{23}$$

where w_1 is a position in a code, w_2 is a new value of the element of the code.

Let

$$C = (c_1, \ldots, c_K) \tag{24}$$

is a code of the mathematical expression for level L.

Then

$$K = 2^{L+2} - 2, \tag{25}$$

and we obtain after the variation (23) a new code

$$\mathbf{w} \circ C = (\overbrace{c_1, \ldots, w_2}^{w_1}, \ldots, c_K). \tag{26}$$

The small variation (23) satisfies the following conditions

$$w_1 \in \{1, \ldots, 2^{L+2} - 2\}, \tag{27}$$

$$w_1 \in \begin{cases} \{1, \ldots, |F_1|\}, & \text{if} \ \ 2^i - 1 \le w_1 \le 3 \cdot 2^{i-1} - 2 \ ; \\ \{1, \ldots, |F_2|\}, & \text{if} \ \ 3 \cdot 2^{i-1} - 1 \le w_1 \le 2^{i+1} - 2 \ ; \\ \{1, \ldots, |F|\}, & \text{if} \ \ 3 \cdot 2^L - 1 \le w_1 \le 2^{L+2} - 2. \end{cases} \tag{28}$$

where $i = 1, \ldots, L$.

A variational genetic algorithm consists of the following stages.

Set a code of some basic solution

$$C_0 = ({c^0}_1, \ldots, {c^0}_K).$$

Generate a set of ordered sets of small variations

$$\Omega = \{W_1, \ldots, W_H\},$$

where W_i is an ordered set of variations (23)

$$W_i = \left[\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,l}\right],$$

$\mathbf{w}^{i,j} = \left[w_1^{i,j} \ w_2^{i,j}\right]^T$, $i = 1, \ldots, H$, $\ j = 1, \ldots, l$, l is a set of numbers of variations.

We carry out crossover and mutation on the sets of variations. Select two parents $W_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,l})$ and $W_j = (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,l})$, and randomly define a point of crossover $k \in \{1, \ldots, l\}$ and exchange tails of the parents

$$\widetilde{W}_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,k-1}, \mathbf{w}^{j,k}, \ldots, \mathbf{w}^{j,l}), \ \widetilde{W}_j = (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,k-1}, \mathbf{w}^{i,k}, \ldots, \mathbf{w}^{i,l}).$$

Define randomly a mutation point $\mu \in \{1, \ldots, l\}$ and generate new variation in this point $\mu \cdot \mathbf{w}^{i,\mu} = \left[w_1^{i,\mu}\ w_2^{i,\mu}\right]^T$ for both new solutions.

To solve the problem of parametric optimal control, searching for points of the state space $\widetilde{\mathbf{X}}^j = (\widetilde{\mathbf{x}}^{j,1}, \widetilde{\mathbf{x}}^{j,2}, \ldots, \widetilde{\mathbf{x}}^{j,K_j}, \varepsilon_j)$, we use evolutionary algorithms that work much better than gradient methods for non-unimodal objective function as we show on the experiment. In this paper, we use three popular evolutionary algorithms that are most suitable for solving the parametric optimal control problem: the genetic algorithm [Goldberg, 1989], the particle swarm optimization [Kennedy & Eberhart, 1995]. [Karpenko & Seliverstov, 2010], the bees algorithm [Pham et al., 2006], and the grey wolf algorithm [Mirjalili et al., 2014].

All evolutionary algorithms use a set of possible solutions with a given number H of elements in which the evolution is performed at each iteration a predetermined number W of times to produce new possible solutions. The solution of the problem is considered to be the best possible solution according to the value of the objective function in the resulting set.

5 Computational Experiment

Consider the task of optimal control of a group of three mobile robots.

Mathematical models of the control objects are the following

$$\dot{x}^j = u_1^j \cos\theta^j,\ \dot{y}^j = u_1^j \sin\theta^j,\ \dot{\theta}^j = u_2^j, \tag{29}$$

where $[x^j\ y^j\ \theta^j]^T$ is a vector of state of the robot j, $[u_1^j\ u_2^j]^T$ is a control vector of the robot j, $j = 1, 2, 3$.

Given the initial conditions

$$x^j(0) = x^{j,0}, y^j(0) = y^{j,0}, \theta^j(0) = \theta^{j,0}, j = 1, 2, 3. \tag{30}$$

Given the following constraint on control

$$u_i^- \leq u_i^j \leq u_i^+, i = 1, 2,\ j = 1, 2, 3. \tag{31}$$

Given the phase constraint

$$r^* - \sqrt{(x^j - x^*)^2 + (y^j - y^*)^2} \leq 0, j = 1, 2, 3, \tag{32}$$

where x^*, y^*, r^* are parameters of the phase constraint, $r^* > 0$.

Given the terminal conditions

$$x^j(t_f) = x^{j,f},\ y^j(t_f) = y^{j,f},\ \theta^j(t_f) = \theta^{j,f},\ j = 1, 2, 3, \tag{33}$$

where

$$t_f = \begin{cases} t, & \text{if} \quad t < t^+ \text{ and } \sum\limits_{j=1}^{3} \sqrt{(x^j(t) - x^{j,f})^2 + (y^j(t) - y^{j,f})^2 + (\theta^j(t) - \theta^{j,f})^2} \leq \varepsilon; \\ t^+, & \text{otherwise} \end{cases} \tag{34}$$

where t^+ and ε are given positive values.

Given the dynamic phase constraints

$$\tilde{r} - \sqrt{(x^1 - x^2)^2 + (y^1 - y^2)^2} \leq 0,\ \tilde{r} - \sqrt{(x^1 - x^3)^2 + (y^1 - y^3)^2} \leq 0,\ \tilde{r} - \sqrt{(x^2 - x^3)^2 + (y^2 - y^3)^2} \leq 0. \tag{35}$$

Given the following quality functional

$$J = t_f + \sum_{j=1}^{3} \sqrt{(x^j(t) - x^{j,f})^2 + (y^j(t) - y^{j,f})^2 + (\theta^j(t) - \theta^{j,f})^2} \to \min. \tag{36}$$

Taking into account the phase constraints, the quality functional has the following form

$$\begin{gathered} \widetilde{J} = t_f + \sum_{j=1}^{3} \sqrt{(x^j(t) - x^{j,f})^2 + (y^j(t) - y^{j,f})^2 + (\theta^j(t) - \theta^{j,f})^2} + \\ \int\limits_0^{t_f} \sum_{j=1}^{3} \vartheta(r^* - \sqrt{(x^j - x^*)^2 + (y^j - y^*)^2}) + \vartheta(\tilde{r} - \sqrt{(x^1 - x^2)^2 + (y^1 - y^2)^2}) + \\ +\vartheta(\tilde{r} - \sqrt{(x^1 - x^3)^2 + (y^1 - y^3)^2}) + \vartheta(\tilde{r} - \sqrt{(x^2 - x^3)^2 + (y^2 - y^3)^2}\, dt \to \min. \end{gathered} \tag{37}$$

In the overall functional (37), the error in fulfilling the terminal conditions and the time period, in which any phase constraints (32), (35) are violated, are added to the time of the control process.

In the computational experiment the parameters of the problem had the following values: $x^{(1)}(0) = 10$, $y^{(1)}(0) = 10$, $\theta^{(1)}(0) = 0$, $x^{(2)}(0) = 0$, $y^{(2)}(0) = 10$, $\theta^{(2)}(0) = 0$, $x^{(3)}(0) = 5$, $y^{(3)}(0) = 10$, $\theta^{(3)}(0) = 0$, $u_1^- = -10$, $u_2^- = -10$, $u_1^+ = 10$, $u_2^+ = 10$, $x^{1,f} = 0$, $y^{1,f} = 0$, $\theta^{1,f} = 0$, $x^{2,f} = 10$, $y^{2,f} = 0$, $\theta^{2,f} = 0$, $x^{3,f} = 5$, $y^{3,f} = 0$, $\theta^{3,f} = 0$, $x^* = 5$, $y^* = 5$, $r^* = 3$, $\tilde{r} = 2$, $t^+ = 2.8$, $\varepsilon = 0.01$, $\delta t = 0.7$.

At the first stage, the problem of synthesis of control by the method of variational complete binary genetic programming for one robot was solved. As a result, the following control was obtained

$$u_i = \begin{cases} u_i^-, & \text{if } \widetilde{u}_i < u_i^- ; \\ u_i^+, & \text{if } \widetilde{u}_i > u_i^+ ; \quad i = 1, 2, \\ \widetilde{u}_i, & \text{otherwise,} \end{cases} \tag{38}$$

where

$$\widetilde{u}_1 = \operatorname{sgn}(\operatorname{sgn}(A)(exp|A| - 1)) \ln(|\exp|A| - 1| + 1),\ \widetilde{u}_2 = B + C,$$

$$A = \operatorname{sgn}(\ln|q_3^3(q_2 - q_2^3)| + \sqrt{1 + \exp(q_3)}) \times (\exp|ln|q_3^3(q_2 - q_2^3)| + \sqrt{1 + \exp(q_3)}) \times$$

$$\times \operatorname{sgn}(\Delta x - \operatorname{sgn}(\Delta y)(\exp|\Delta y| - 1)\operatorname{sgn}(\Delta\theta)) \times \ln(\Delta x - \operatorname{sgn}(\Delta y)(\exp|\Delta y| - 1)\operatorname{sgn}(\Delta\theta)| + 1),$$

$$B = \operatorname{sgn}(q_3\Delta\theta)(\exp|q_3\Delta\theta| - 1) + \left(\mu(\Delta\theta) + \frac{1 - \exp(\Delta y/\Delta x)}{1 + \exp(\Delta y/\Delta x)}\right)^{-1},$$

$$C = \operatorname{sgn}(q_2^3\Delta y(\Delta x + \Delta y))\sqrt{|q_2^3\Delta y(\Delta x + \Delta y)|} \times \operatorname{sgn}(\operatorname{sgn}(\Delta\theta)(\exp|\Delta\theta| - 1)(q_3 - q_3^3)) \times$$

$$\times (\ln|(\exp|\Delta\theta| - 1)(q_3 - q_3^3)| + 1) + \mu(\Delta x \Delta y),$$

$$\Delta x = x^f - x, \Delta y = y^f - y, \Delta\theta = \theta^f - \theta, q_1 = 0.84180, q_2 = 0.65527, q_3 = 2.45020.$$

The obtained control (38) is a nonlinear function $u_i^j = g_i^j(x^{j,1} - x^j,\ y^{j,1} - y^j, \theta^{j,1} - \theta^j)$, $i = 1, 2,\ j = 1, 2, 3$, that depends on the coordinates of the vector of the state of the robot j, $j = 1, 2, 3$, and ensures the stability of the system of differential equations (29) with respect to the point $[x^{j,1}\ y^{j,1}\ \theta^{j,1}]^T$.

At the second stage we solve the problem of parametric optimal control by evolutionary algorithms. It is necessary to find the coordinates of the points $[x^{j,1}\ y^{j,1}\ \theta^{j,1}]^T$, $[x^{j,2}\ y^{j,2}\ \theta^{j,2}]^T$, $j = 1, 2, 3$, which are together with the terminal point $[x^{j,f}\ y^{j,f}\ \theta^{j,f}]^T$ the stabilization points of the system (29). Switching from points $[x^{j,1}\ y^{j,1}\ \theta^{j,1}]^T$ to points $[x^{j,2}\ y^{j,2}\ \theta^{j,2}]^T$ and from points $[x^{j,2}\ y^{j,2}\ \theta^{j,2}]^T$ to points $[x^{j,f}\ y^{j,f}\ \theta^{j,f}]^T$ is performed at specified time moments $t_1 = 1s$, $t_2 = 2s$ and the points found should provide the minimum value of the functional (37).

When solving the problem of parametric optimal control, the parameters of evolutionary algorithms were chosen so that the total number of calculations of the objective function was approximately the same. The results of the calculations are given in Table 1. For comparison, the problem was also solved by the algorithm of multipoint fast gradient descent, and the best solution of 48 starts was taken. The table shows values of the functional (37) for each of the 10 tests and the number of calculations of the functional in each test. The table contains: GA – genetic algorithm, PSO – particle swarm optimization, BA – bee algorithm, GW – the grey wolf algorithm, MFGD – multipoint fast gradient descent. The last line of the table shows the average values for ten tests.

From the results of the computational experiments it follows that the tested evolutionary algorithms find solutions with approximately the same values of the objective function, and much better than the solutions found by the fast gradient descent. The best solution found is $\widetilde{J} = 1.9241$:

$$[x^{1,1}\ y^{1,1}\ \theta^{1,1}]^T = [0.08539\ 3.1936\ \ -0.0469]^T,\ \ [x^{2,1}\ y^{2,1}\ \theta^{2,1}]^T = [5.7851\ 7.8697\ 0.5353]^T,$$

$$[x^{3,1}\ y^{3,1}\ \theta^{3,1}]^T = [0.0015\ 1.4369\ 0.1729]^T,\ \ [x^{1,2}\ y^{1,2}\ \theta^{1,2}]^T = [0.7282\ 1.5034\ \ -0.0294]^T,$$

$$[x^{2,2}\ y^{2,2}\ \theta^{2,2}]^T = [10.0000\ 0.5002\ 0.0927]^T,\ \ [x^{3,2}\ y^{3,2}\ \theta^{3,2}]^T = [9.4238\ 3.1071\ \ -0.7933]^T.$$

Acknowledgements

This work was supported by Russian Foundation for Basic Research, Projects 16-29-04224-ofim, 17-08-01203-a.

Table 1: Results of Computational Experiments

GA		PSO		MFGD		BA		GW	
2.2117	64600	2.8918	65602	4.9758	86626	2.8746	65838	2.0115	65026
2.2424	64616	2.3665	65602	6.1991	82530	2.8751	65838	2.0028	65026
2.4058	65314	2.8514	65602	6.5167	74338	2.8744	65838	2.1668	65026
2.5942	65350	2.8702	65602	5.7414	94818	2.8740	65838	2.2147	65026
2.8727	65458	2.2850	65602	5.7305	80482	2.8720	65838	1.9924	65026
2.5462	65332	2.8459	65602	4.7807	96866	2.8965	65838	2.0120	65026
2.3855	65204	2.8459	65602	5.8009	84578	2.8681	65838	2.0333	65026
2.3068	65214	2.4346	65602	6.1118	86626	2.8681	65838	2.1706	65026
2.1633	65602	2.8943	65602	6.2203	84578	2.8729	65838	2.1312	65026
2.6650	65556	2.3971	65602	5.8471	90722	2.8689	65838	1.9241	65026
2.4394	65225	2.6733	65602	5.7924	86216.4	2.8745	65838	2.0659	65026

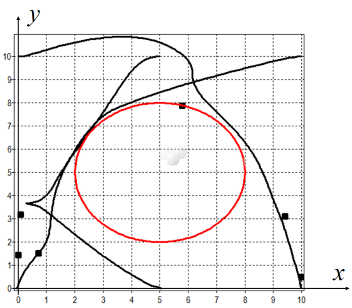

Figure 3: Optimal trajectories of movement of robots

References

[Diveev, 2015a] Diveev, A.I., (2015). *Approximate methods for solving the optimal control synthesis problem* Moscow: CC RAS [in Russian].

[Diveev, 2015b] Diveev, A.I., (2015). Small Variations of Basic Solution Method for Non-numerical Optimization *Proceedings of 16th IFAC Workshop on Control Applications of Optimization, CAO 2015.* October 6th-9th, 2015, Garmisch-Partenkirchen, pp. 28-33.

[Goldberg, 1989] Goldberg, D.E., (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[Kennedy & Eberhart, 1995] Kennedy, J., & Eberhart, R., (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks IV, 1995.*. pp. 1942-1948.

[Karpenko & Seliverstov, 2010] Karpenko, A.P., & Seliverstov, E.Yu., (2010). Global optimization by the particle swarm method. Overview. *Information technologies, 2010 (2).*, pp. 25-34. [in Russian].

[Pham et al., 2006] Pham, D.T., et al. (2006). The Bees Algorithm - A Novel Tool for Complex Optimisation Problems *Intelligent Production Machines and Systems - 2nd I*PROMS Virtual International Conference 3-14 July 2006.* Elsevier Ltd, pp. 25-34.

[Mirjalili et al., 2014] Mirjalili S.A., Mirjalili S.M., Lewis A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software Vol.69* , pp. 46-61.