

УДК 004.896, 62.50

А.В. Доценко¹, А.И. Дивеев^{1,2}

РАЗРЕШЕНИЕ КОЛЛИЗИЙ В ЗАДАЧЕ ГРУППОВОГО ВЗАИМОДЕЙСТВИЯ МОБИЛЬНЫХ РОБОТОВ С ИЗМЕНЕНИЕМ НАБОРА АРГУМЕНТОВ ФУНКЦИИ УПРАВЛЕНИЯ

Российский университет дружбы народов¹
Федеральный исследовательский центр «Информатика и управление» РАН²

Рассматривается децентрализованный подход разрешения столкновений при групповом взаимодействии гусеничных мобильных роботов. Для каждого из них предусматривается отдельный аппроксиматор функции управления, осуществляющий поиск как в пространстве функций, так и в пространстве параметров. Предполагается, что роботы значительным образом ограничены в предоставляемой им информации о среде, а также о состоянии динамических и статических фазовых ограничений в каждый момент времени: можно обмениваться информацией о текущем состоянии с другими роботами, если те находятся в области видимости, задаваемой окружностью. Специфической чертой является наличие для функции управления робота возможности расширить набор аргументов, если в области видимости рассматриваемого робота появились другие, и сократить набор аргументов – если соседние роботы покинули его область видимости. Приведен вычислительный эксперимент, в котором для поиска функций управления использовался модифицированный метод сетевого оператора. Исходя из результатов эксперимента по перестроению группы, найденная для каждого робота функция управления доставила свой объект управления в терминальное состояние, позволив каждому роботу полностью избежать столкновений с другими роботами.

Ключевые слова: синтез системы управления, метод сетевого оператора, управление группой роботов, генетический алгоритм.

Введение

Задача разрешения коллизий не имеет унифицированной математической постановки в силу широкого набора технических решений, а также требований к входным данным. В рамках проводимых в данной области исследований сформировалась определенная иерархия подходов. В ряде работ рассматривается избегание столкновений благодаря планированию траекторий [1, 2]. Данная парадигма делится на два класса: сопряженный, где в целях избегания коллизий зачастую используется диаграмма координации [3, 4], и несвязный, который, в свою очередь, подразделяется на централизованный [5, 6] и децентрализованный [7]. Несвязное планирование также можно разделить на две категории:

- а) координирование путей [8-11];
- б) приоритетное планирование [12-14].

В настоящем исследовании акцент делается на задаче избегания коллизий при перестроении группы роботов в контексте постановки задачи синтеза системы управления [16]. Метод сетевого оператора [17, 18] был разработан специально для решения задачи синтеза системы управления. В отличие от других методов символьной регрессии (генетического программирования [19], грамматической эволюции [20], аналитического программирования [21] и недавно появившегося метода матриц разбора [22]), метод сетевого оператора производит поиск не только в пространстве функций, но и по параметрам, благодаря чему становится схожим с нейронными сетями. В экспериментальной части задействована модификация метода сетевого оператора [23], написанная на языке программирования Python.

Постановка задачи

Пусть задана некоторая система ОДУ, задающая динамику мобильных роботов (1):

$$\begin{cases} \dot{x}^i = \frac{(u_1^i + u_2^i) \cos(\theta^i)}{2} \\ \dot{y}^i = \frac{(u_1^i + u_2^i) \sin(\theta^i)}{2}, \\ \dot{\theta}^i = \frac{(u_1^i - u_2^i)}{2} \end{cases}, \quad (1)$$

где n – количество роботов, $\mathbf{u}^i \in U \subseteq \mathfrak{R}^m$, U – ограниченное замкнутое множество, $i = 1, \dots, n$.

При наложенных на управление ограничениях (2):

$$\mathbf{u}^i = [u_1^i, u_2^i] \in [-20.0, 20.0], \quad (2)$$

т.е. ограничение скорости до 1 см/с.

Пусть задано максимальное время моделирования t^+ , начальные условия

$$x_i(0) = x_i^0, y_i(0) = y_i^0, \theta_i(0) = \theta_i^0, \quad (3)$$

и терминальные условия (4):

$$x_i(t_f), y_i(t_f), \theta_i(t_f), \quad (4)$$

где t_f – время симуляции, $x_i^0, y_i^0, \theta_i^0 \in X_0 \subseteq \mathfrak{R}^n, i = 1, \dots, n$.

Условия отсутствия столкновения роботов записываем в виде динамических фазовых ограничений (5):

$$D_c = R_c - \sqrt{(x_i - x_{j \neq i})^2 + (y_i - y_{j \neq i})^2} \leq 0, \quad (5)$$

где R_c – радиус, задающий габариты робота (радиус столкновения), где i – индекс рассматриваемого робота, j – индекс ближайшего робота,

$$j = \arg \min \left\{ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right\}, i, j = 1, \dots, n, j \neq i.$$

Функционал качества задан в следующем виде (6):

$$J = t_f + \sum_{i=1}^n \sqrt{d_1^2 + d_2^2} + \sum_{i=1}^n \sum_{j=i+1}^n \int_0^{t_f} D_c dt \rightarrow \min, \quad (6)$$

где

$$d_1 = x_{i,f} - x_i,$$

$$d_2 = y_{i,f} - y_i,$$

$$d_3 = x_i - x_{j \neq i},$$

$$d_4 = y_i - y_{j \neq i},$$

$$t_f = \begin{cases} t, \text{ если } t < t^+ \text{ и } \sum_{i=1}^n \sqrt{d_1^2 + d_2^2} < \varepsilon \\ t^+ + \sum_{i=1}^n \sqrt{d_1^2 + d_2^2} - \text{иначе} \end{cases}.$$

Наконец, искомые управления представлены как функции, зависящие от состояния (7):

$$\dot{\mathbf{s}}^i = \mathbf{f}(\mathbf{s}^i, \mathbf{h}^i(\mathbf{s}^i)). \quad (7)$$

Целью является нахождение для каждого робота, описываемого системой (1), такой функции управления (7), удовлетворяющей условию оптимальности (6) для всех возможных начальных состояний (3), чтобы достиглось терминальное состояние (4) при наложенных ограничениях на управления (2) и динамических фазовых ограничениях (5).

Изменение правых частей дифференциальных уравнений

Специально для задачи разрешения коллизий в данной статье представлена модификация метода. Главной модификацией является ситуативное внедрение относительных координат двух ближайших роботов в правые части ОДУ. Таким образом, поиск оптимальной функции управления для взаимодействия соседних агентов встроен в процесс поиска оптимальной функции управления для достижения терминальных условий. Иными словами, на некотором шаге интегрирования агент определяет, входит ли ближайший к нему соседний агент в область взаимодействия. Если соседний робот находится достаточно близко, то относительные координаты этих двух агентов включаются в правые части ОДУ первого агента, а также второго, если они оба являются друг для друга ближайшими соседями.

Из этого следует необходимость введения гиперпараметра, благодаря которому определяется необходимость расширения вектора аргументов функции управления. Нельзя дать точный ответ на вопрос – какое значение должен принимать этот гиперпараметр. Можно лишь утверждать, что он, в первую очередь, зависит от технического оснащения агента, а также от среды и размера объекта. Также следует отметить, что параметр может изменяться в ходе работы алгоритма. Его поиск можно осуществлять наряду с аргументами и параметрами искомой функции, однако в этой работе было решено ограничиться фиксированным значением.

Будем считать, что если выполняется условие $D_i = R_i - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > 0$, где R_i – радиус видимости робота (радиус взаимодействия), где i – индекс рассматриваемого робота, j – индекс ближайшего робота, $j = \arg \min \left\{ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right\}, i, j = 1, \dots, n, j \neq i$, то два робота вступают во взаимодействие.

Далее, в случае, если ближайший робот находится во взаимодействии с рассматриваемым роботом, осуществляется шаг интегрирования с расширенным вектором аргументов, иначе – в качестве аргумента остается собственный вектор состояния (8).

$$\mathbf{u}^i = \begin{cases} \mathbf{h}^i(\mathbf{s}^i, \mathbf{s}^k), & \text{если } D_i > 0 \\ \mathbf{h}^i(\mathbf{s}^i) - \text{иначе.} \end{cases} \quad (8)$$

Вычислительный эксперимент

В вычислительном эксперименте демонстрируется перестроение группы из восемнадцати мобильных роботов. Начальное расположение группы можно видеть на рис. 2. Желаемые и фактические терминальные состояния роботов изображены на рис. 3. В качестве оптимизатора использовался модифицированный метод сетевого оператора [23], реализованный на языке программирования Python. В процессе моделирования использовались следующие значения гиперпараметров: время моделирования, $t = 3$, шаг интегрирования, $dt = 0.03$, радиус взаимодействия, $R_i = 4$, радиус коллизии, или габарит, $R_c = 1$. Гиперпараметры сетевого оператора следующие: количество индивидов в популяции, $P = 6000$, количество поколений, $G = 100$, минимальное значение параметров, $q_{\min} = -20$, максимальное значение параметров, $q_{\max} = 20$, число вариаций, $l = 8$, количество скрещиваний, $R = 256$, количество потомков, $C = 8$.

Для отображения расположения роботов на плоскости используется график с черными точками, где каждая отдельная точка отображает состояние определенного робота в некий момент времени. Для отслеживания динамики на каждом шаге интегрирования сохранялось состояние всех роботов в виде черных точек. В данной статье представлены самые информативные с точки зрения избегания коллизий, а также достижения терминальных состояний, графики из коллекции. Исходя из рис. 3, можно сделать вывод, что роботы успешно перегруппировались, заняв желаемые терминальные состояния в пределах заданной в ходе моделирования точности.

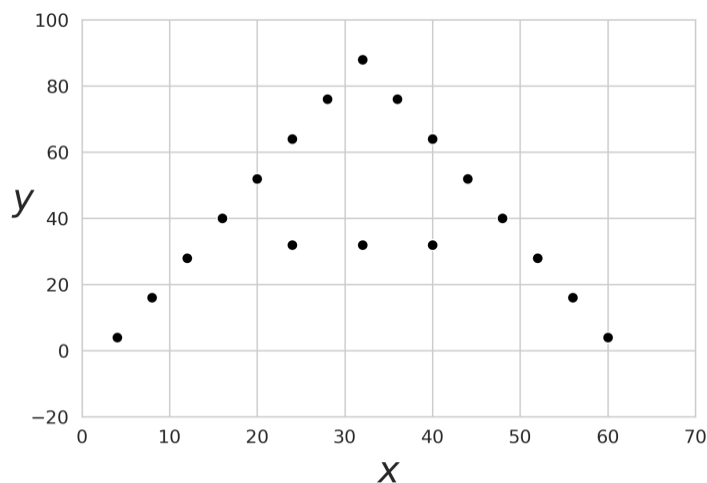


Рис. 2. Начальное расположение группы роботов

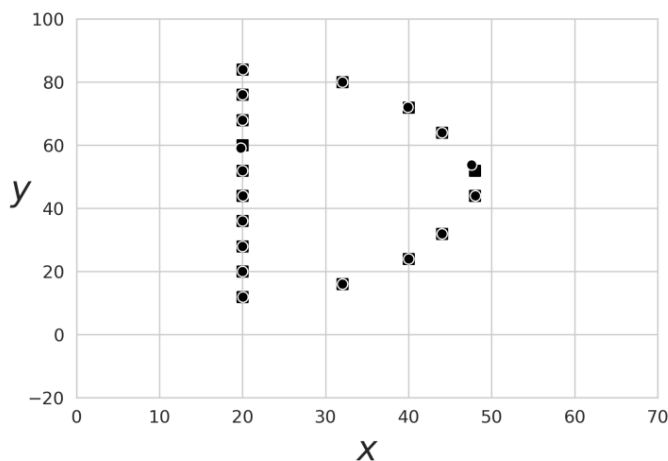


Рис. 3. Желаемое расположение роботов (квадраты) и фактическое (точки)

Чтобы продемонстрировать процесс взаимодействия двух ближайших роботов с изменением набора аргументов функции управления, был выбран момент времени, в который было зафиксировано резкое уклонение от столкновения. Стоит отметить, что это всего лишь один из нескольких зафиксированных случаев избегания столкновений между роботами в ходе данного перестроения. Ради наглядности в этой работе использовалась ситуация с самой резкой реакцией роботов.

Сопоставляя рис. 4 и рис. 5, можно увидеть взаимодействие двух роботов, отмеченное черным овалом.

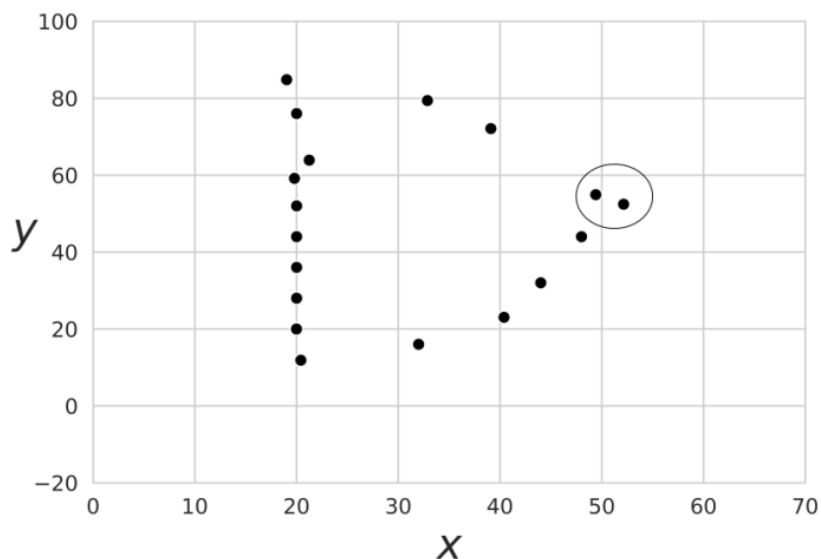


Рис. 4. Взаимодействие роботов в некоторый момент времени

На рис. 4, в ситуации, когда почти все роботы заняли свои терминальные состояния, два робота внутри овала сближаются, причем один из них находится достаточно далеко от своего терминального состояния.

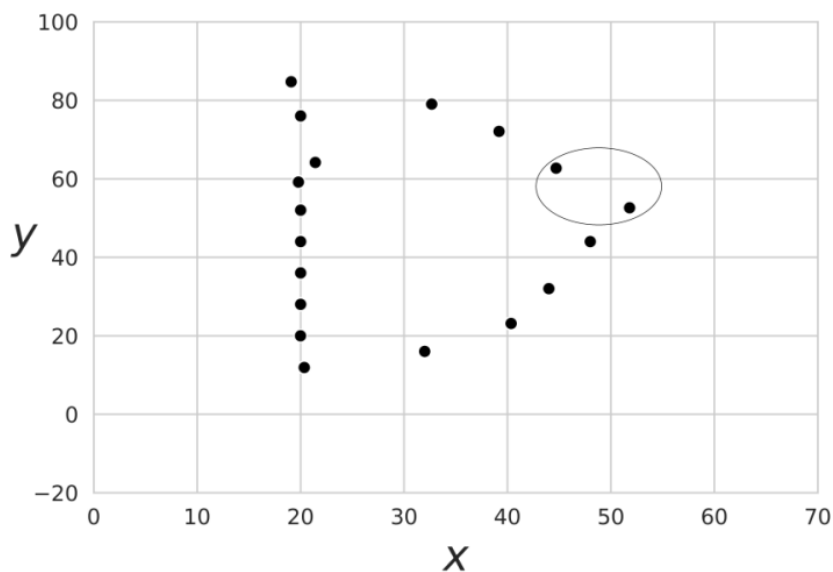


Рис. 5. Взаимодействие роботов в последующий момент времени

Далее, на рис. 5 изображен следующий момент времени, когда один из роботов отстраняется от другого в направлении терминального состояния, посредством включения относительных координат соседа в свою функцию управления. Обратим внимание, что на рис. 3 роботы, достигшие своих терминальных состояний, не находятся во взаимодействии. Однако роботы, достигшие терминальных состояний, все равно прекратят движение, даже если они продолжают находиться во взаимодействии.

Сетевой оператор робота с самым большим значением по оси ординат на рис. 2 будет иметь следующий вид:

$$R = [[(0,0), (6,0), (0,2), (12,3)],$$

$$[(1,0), (7,0), (13,1)],$$

$$[(2,0), (8,0), (14,1)],$$

$$[(3,0), (9,0), (15,1)],$$

$$[(4,0), (10,0), (16,1)],$$

$$[(5,0), (11,1), (2,12), (2,12), (17,1)],$$

$$[(12,0), (13,0), (14,0), (15,0), (16,0), (17,0), (6,7), (18,0)],$$

$$[(12,0), (13,0), (14,0), (15,0), (16,0), (17,0), (5,5), (5,5), (5,5), (5,5), (19,0)]]].$$

Полученная структура является несокращенной, о чем свидетельствует, например, наличие в последней строке унарной операции под номером 5 четыре раза. Его вектор параметров имеет следующие значения: $p = [1.00, 1.00, 2.05, 18.56, -16.50, -17.21]$.

Унарные и бинарные операции, используемые в ходе поиска оптимальных функций управления, приведены в табл. 1 и табл. 2 соответственно.

Таблица 1

Множество унарных операций

Номер операции	Операция
0	$f(x) = x$
1	$f(x) = -x$
2	$f(x) = x^2$
3	$f(x) = \sin(x)$
4	$f(x) = \begin{cases} 1, & \text{если } x \geq 0 \\ 0 & \text{иначе.} \end{cases}$
5	$f(x) = \cos(x)$
6	$f(x) = \arctan(x)$
7	$f(x) = e^x$
8	$f(x) = \ln x$
9	$f(x) = \operatorname{sgn}(x)\sqrt{ x }$
10	$f(x) = x^3$
11	$f(x) = x^{-1}$
12	$f(x) = \sqrt[3]{x}$
13	$f(x) = e^x - 1$
14	$f(x) = 2^x$
15	$f(x) = \operatorname{sgn}(x)$
16	$f(x) = \lg x$
17	$f(x) = \log_2 x$
18	$f(x) = \ln(1 + x)$
19	$f(x) = x $
20	$f(x) = \tan x$
21	$f(x) = \tanh x$

Таблица 2

Множество бинарных операций

Номер операции	Операция
0	$f(x, y) = x + y$
1	$f(x, y) = xy$
2	$f(x, y) = \max(x, y)$
3	$f(x, y) = \min(x, y)$

Используя индексы входных и выходных узлов, можно перевести закодированную запись в математическое выражение. В случае самого верхнего (по оси ординат) на рис. 2 робота, получим следующие функции управления:

$$u_1 = \min(1, x - x^f) + (y - y^f) + 2.05(\theta - \theta^f) + 18.56(x - x_n) - 16.50(y - y_n) - 17.21(-\theta + \theta_n) + e^{(x-x^f)},$$

$$u_2 = \min(1, x - x^f) + (y - y^f) + 2.05(\theta - \theta^f) + 18.56(x - x_n) - 16.50(y - y_n) - 17.21(-\theta + \theta_n) + 4\cos(-17.21).$$

Отметим, что приведенные функции управления были сокращены вручную. Возможность сокращать полученные математические выражения автоматически является предметом исследования последующих работ по тематике методов символьной регрессии.

Заключение

Рассмотрена задача разрешения коллизий. Экспериментально подтверждено, что можно осуществлять поиск функций управления с ситуативным изменением набора аргументов этой функции. Найденные функции одновременно позволяют роботам избегать коллизии и попадать в терминальные состояния.

Избегание столкновений в постановке задачи синтеза системы управления представляется иной, выбивающейся из существующей иерархии методов группового взаимодействия, перспективой задачи при управлении группой роботов. Однако в данной работе осуществлялся поиск отдельной для каждого робота функции, что лишь подтверждает способность функций с переменным числом аргументов как доставлять объект управления в терминальное состояние, так и избегать столкновений с динамическими фазовыми ограничениями. В продолжение данной работы, в будущем планируется провести ряд экспериментов по синтезу функции управления с изменением правых частей дифференциальных уравнений для многочисленной группы роботов с использованием единого аппроксиматора.

Библиографический список

1. **Tian, J.** Dynamic collision avoidance path planning for mobile robot based on multi-sensor data fusion by support vector machine / J. Tian, M. Gao, E. Lu // 2007 international conference on mechatronics and automation. IEEE, 2007. – P. 2779-2783.
2. **Fujimori, A.** Adaptive navigation of mobile robots with obstacle avoidance / A. Fujimori, P.N. Nikiforuk, M.M. Gupta // IEEE Transactions on Robotics and Automation. – 1997. – Vol. 13. – N 4. – P. 596-601.
3. **Barraquand, J.** Robot motion planning: A distributed representation approach / J. Barraquand, J.C. Latombe // The International Journal of Robotics Research, 1991. – Vol. 10. – № 6. – P. 628-649.

4. **Sanchez, G.** Using a PRM planner to compare centralized and decoupled planning for multi-robot systems / G. Sanchez, J.C. Latombe // Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). IEEE, 2002. – Vol. 2. – P. 2112-2119.
5. **Oh, J.H.** Centralized decoupled path planning algorithm for multiple robots using the temporary goal configurations / J.H. Oh, J.H. Park, J.T. Lim // 2011 Third International Conference on Computational Intelligence, Modelling and Simulation. IEEE, 2011. – P. 206-209.
6. **Den Berg, J.** Centralized path planning for multiple robots: Optimal decoupling into sequential plans / J. Den Berg // Robotics: Science and systems, 2009. – Vol. 2. – № 2(5). – P. 2-3.
7. **Devasia, S.** Decoupled conflict-resolution procedures for decentralized air traffic control / S. Devasia // IEEE Transactions on intelligent transportation systems, 2011. – Vol. 12. – № 2. – P. 422-437.
8. **O'Donnell, P.A.** Deadlock-free and collision-free coordination of two robot manipulators / P.A. O'Donnell, T. Lozano-Pérez // ICRA, 1989. – Vol. 89. – P. 484-489.
9. **Griswold, N.C.** Control for mobile robots in the presence of moving objects / N.C. Griswold, J. Eem // IEEE Transactions on Robotics and Automation, 1990. – Vol. 6. – № 2. – P. 263-268.
10. **Pan, T.J.** Motion planning for mobile robots in a dynamic environment with moving obstacles / T.J. Pan, R.C. Luo // Proceedings, IEEE International Conference on Robotics and Automation. IEEE, 1990. – P. 578-583.
11. **Rude, M.** Collision avoidance by using space-time representations of motion processes / M. Rude // Autonomous Robots, 1997. – Vol. 4. – № 1. – P. 101-119.
12. **Erdmann, M.** On multiple moving objects / M. Erdmann, T. Lozano-Perez // Algorithmica, 1987. – Vol. 2. – № 1-4. – P. 477.
13. **Bennewitz, M.** Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots / M. Bennewitz, W. Burgard, S. Thrun // Robotics and autonomous systems, 2002. – Vol. 41. – № 2-3. – P. 89-99.
14. **Clark, C.M.** Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems / C.M. Clark, T. Bretl, S. Rock // Proceedings, IEEE Aerospace Conference. IEEE, 2002. – Vol. 7. – P. 7.
15. **Diveev, A.I.** Application of network operator method for synthesis of optimal structure and parameters of automatic control system / A.I. Diveev, E.A. Sofronova // IFAC Proceedings Volumes. 2008. – Vol. 41. – № 2. – P. 6106-6113.
16. **Diveev, A.I.** Self-adjusting control for multi robot team by the network operator method / A.I. Diveev, E.Y. Shmalko // 2015 European Control Conference (ECC). IEEE, 2015. – P. 709-714.
17. **Diveev, A.I.** Numerical method of network operator for multiobjective synthesis of optimal control system / A.I. Diveev, E.A. Sofronova // 2009 IEEE International Conference on Control and Automation. IEEE, 2009. – P. 701-708.
18. **Diveev, A.** The Network operator numerical method for solving the problem of control system synthesis with undefined initial conditions / A. Diveev // Izvestiya RAS Theory and control systems. 2012. – Vol. 2. – P. 63-78.
19. **Koza, J.R.** Genetic programming: on the programming of computers by means of natural selection / J.R. Koza, J.R. Koza // MIT press. – 1992. – Vol. 1.
20. **O'Neill, M.** Grammatical evolution / M. O'Neill, C. Ryan // IEEE Transactions on Evolutionary Computation. 2001. – Vol. 5. – № 4. – P. 349-358.
21. **Zelinka, I.** Analytic programming by means of soma algorithm / I. Zelinka // Proceedings of the 8th International Conference on Soft Computing, Mendel. 2002. – Vol. 2. – P. 93-101.
22. **Luo, C.** Parse-matrix evolution for symbolic regression / C. Luo, S.L. Zhang // Engineering Applications of Artificial Intelligence. 2012. – Vol. 25. – № 6. – P. 1182-1193.
23. **Dotsenko, A.V.** Python package for the network operator implementation and its application for group control system synthesis problem / A.V. Dotsenko // Procedia Computer Science. 2019. – Vol. 150. – P. 679-686.

Дата поступления

в редакцию: 16.02.2029

A.V. Dotsenko¹, A.I. Diveev^{1,2}

COLLISION AVOIDANCE AT SWARM REGROUPING WITH VARIABLE SET OF ARGUMENTS OF CONTROL FUNCTION

Peoples Friendship University of Russia¹

Federal Research Center Computer Science and Control of the Russian Academy of Sciences²

Purpose: To study the possibility of applying control functions with variable set of arguments in collision avoidance problem and to propose a method which is aimed to find such control functions.

Approach: It is suggested to equip each robot with its own optimizer which is able to search in the space of functions and parameters as well. Robots pursue not only to reach terminal states but also to avoid collisions with another robots. In order to accomplish both goals we introduce a new approach which assumes that right parts of differential equations of control object can be modified.

Findings: It was experimentally confirmed that situational extension or contraction of control function argument vector is a viable approach to resolving collisions between robots.

Research limitations: The present study provides a starting-point for further research of multi-robot interaction domain which is formulated here as a synthesis problem but here we assume each robot to have its own optimizer when the only control function is needed.

Originality: There is a plenty of methods addressing collision avoidance problem. Proposed in this study approach is somewhat special because it fully relies on calculations and its key feature of situational argument vector modification is unique.

Key words: synthesis of control system, modified network operator method, group of robots, genetic algorithm.