

Синтез универсальной вероятностной системы избегания столкновений в задаче перестроения группы динамических объектов

А. В. Доценко

Аннотация—Статья посвящена решению задачи синтеза системы управления, осуществляющей избегание столкновений в задаче перестроения группы динамических объектов. Проводится поиск замкнутой системы управления с вероятностным отображением вектора состояния в вектор оптимального управления. Одновременное столкновение более двух динамических объектов можно представить как совокупность отдельно взятых пар объектов, нарушающих динамические фазовые ограничения, поэтому поиск оптимальной системы управления реализуется как обучение специальной нейронной сети на парных взаимодействиях объектов. Новизна работы состоит в том, что искомая система управления реализована в виде функции плотности условной вероятности, где управление выступает в качестве случайной величины, зависимой от состояния объекта и его помехи. Рассматриваемая система управления является универсальной: она способна разрешать коллизии между динамическими объектами в неизвестной среде для любых начальных и терминальных состояний объектов. Более того, она способна учитывать альтернативные маршруты объезда препятствий. Представленная система разрешения коллизий является децентрализованной: она записывается на запоминающее устройство каждого объекта, за счет чего обеспечивается автономность объектов в принятии решений. Поскольку на сегодняшний день решение задачи управления группой роботов является востребованным в различных областях человеческой деятельности, и в то же время проблемы этой задачи не решены, в частности проблема избегания столкновений роботов, то это определяет актуальность работы.

Ключевые слова—Разрешение коллизий, перцептрон, синтез управления, эволюционные стратегии, смешанная плотность вероятностей.

I. ВВЕДЕНИЕ

Основным препятствием на пути создания системы управления группой динамических объектов является проблема разрешения коллизий или обеспечение отсутствия столкновений между объектами. В зависимости от конкретных постановок известны различные способы решения этой проблемы, в частности, используются планировщики маршрутов [1], методы потенциальных полей [3], [4] методы на основе координирования маршрутов [5], [6], реактивные подходы [7], [8] и т.д. Предыдущие работы в области коллективного управления подразумевают проведение вычислений, направленных на

поиск оптимального управления, в реальном времени. Планировщики маршрутов, осуществляющие поиск в совместном конфигурационном пространстве, в теории, являются методами, способными находить маршруты, которые являются компромиссными и оптимальными для всей группы. Однако они не способны находить оптимальные управления и маршруты для больших групп из-за проблемы проклятия размерности. Несвязные планировщики осуществляют прокладку маршрута независимо от других объектов управления группы. Они прокладывают оптимальный маршрут заранее, а в случае возникновения на пути объекта препятствий, корректируют его на ходу. Методы регулирования скоростей предъявляют высокие требования к данным, считываемым с датчиков, а также осуществляют вычисления на ходу и чувствительны к параметрам алгоритма поиска оптимальных скоростей. Методы на основе полей потенциалов приводят к колебательному движению динамических объектов при их нахождении в непосредственной близости с препятствиями и теряют в эффективности из-за проблемы локальных минимумов, когда объект попадает в тупиковую ситуацию.

В работах [9]-[13] приведено решение задачи синтеза управления группой роботов на основе современных эволюционных методов символьной регрессии. Здесь рассматриваются два основных направления: непосредственный синтез с учетом динамических фазовых ограничений, которые включены в функционал качества, и метод синтезированного управления, когда обеспечивается устойчивость объекта относительно точки пространства состояний с дальнейшим подбором положения точек таким образом, чтобы переместить объекты в терминальные состояния оптимально, согласно критерию качества, без столкновений между собой. Данные методы синтеза могут выполняться на этапе создания систем управления или требуют дополнительного решения оптимизационных задач на мощном бортовом компьютере. Это означает что, если терминальные условия изменяются, то, следовательно, изменяются и потенциально приводящие к столкновению положения, и задачу синтеза управления необходимо решать вновь.

В данной работе предлагается модульный подход к управлению группой динамических объектов, а именно, отделение системы избегания столкновений в самостоятельный орган управления. Представленная система

Статья получена 25 мая 2021.

Антон Викторович Доценко, инженер-программист АО «Корпорация «ВНИИЭМ» (anton.dozenko@gmail.com).

разрешения коллизий не осуществляет поиск оптимального управления в реальном времени, а представляет замкнутую по состоянию функцию управления, выдающую непрерывный сигнал управления как вероятностную величину, зависящую от собственного состояния и состояния объекта-помехи.

II. ПОСТАНОВКА ЗАДАЧИ

Избегание коллизий в постановке задачи синтеза системы управления задается как условие соблюдения динамических фазовых ограничений. Фазовые ограничения являются существенной проблемой при решении задачи оптимального управления. Известные подходы для решения задачи оптимального управления с фазовыми ограничениями вводят дополнительные переменные [14], нахождение значений которых представляет определенную вычислительную сложность. Более практический способ решения задачи соблюдения фазовых ограничений состоит во включении этих ограничений в виде штрафных функций в функционал качества. Но здесь возникает проблема определения весовых коэффициентов, которые могут существенно исказить значение функционала.

В задаче управления группой роботов возникает проблема соблюдения динамических фазовых ограничений. Это означает, что нарушение фазовых координат может происходить в любой точке пространства состояний, что существенно усложняет поиск решения поставленной задачи.

Задана математическая модель динамических объектов управления гомогенной группы

$$\dot{\mathbf{s}}^i = \mathbf{f}(\mathbf{s}^i, \mathbf{u}^i), \quad (1)$$

где $\mathbf{s}^i = [s_{i,1}, \dots, s_{i,n}]^T$ – вектор состояния объекта; $\mathbf{s}^i \in \mathbb{R}^n$, $\mathbf{u}^i = [u_{i,1}, \dots, u_{i,m}]^T$ – вектор управления; $\mathbf{u}^i \in U \subseteq \mathbb{R}^m$, U – ограниченное замкнутое множество, $i = 1, N$, N – количество динамических объектов в группе.

Задано множество начальных состояний

$$S_{0,i} = \{s_1^{0,i}, \dots, s_{k_i}^{0,i}\}, \quad (2)$$

множество терминальных состояний

$$S_{f,i} = \{s_1^{f,i}, \dots, s_{r_i}^{f,i}\}, \quad (3)$$

где k_i и r_i – количество начальных и терминальных состояний объекта i соответственно.

Подвектор $\mathbf{y}^i = [y_{i,1}, \dots, y_{i,p}]^T$ вектора \mathbf{s}^i определяет местоположение объекта i .

Заданы динамические фазовые ограничения в виде неравенства

$$R^2 - \sum_{c=1}^p (y_{i,c} - y_{j,c})^2 < 0, \quad (4)$$

где R^2 – радиус сферы, задающей габариты динамических объектов, $i, j = \{1, \dots, N\}$, $i \neq j$.

Задан функционал качества

$$J = \int_0^{\hat{t}_f} f_0(\mathbf{s}^1, \dots, \mathbf{s}^N, \mathbf{u}^1, \dots, \mathbf{u}^N) dt \rightarrow \min_{\mathbf{u} \in U}, \quad (5)$$

где $\hat{t}_f = \max \{t_{f,i}\}_{i=1}^N$ – ограниченное максимальное

время процесса управления, которое может быть задано или определяться по достижению терминальных условий.

Требуется найти адаптивное управление в форме функции от компонентов вектора состояний объекта

$$\mathbf{u}^i = \begin{cases} \mathbf{h}(\mathbf{s}^i), & \text{если } R_o^2 - \sum_{c=1}^p (y_{i,c} - y_{j,c})^2 \geq 0, \\ \mathbf{g}(\mathbf{s}^i, \mathbf{s}^j) & \text{иначе,} \end{cases} \quad (6)$$

где $\mathbf{h}(\mathbf{s}^i): \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{g}(\mathbf{s}^i, \mathbf{s}^j): \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m$,

$\mathbf{h}(\mathbf{s}^i), \mathbf{g}(\mathbf{s}^i, \mathbf{s}^j) \in U \subseteq \mathbb{R}^m$, $i, j = \{1, \dots, N\}$, $i \neq j$, R_o – радиус локальной окрестности, внутри которой динамические объекты способны обнаруживать другие объекты.

Подставив (6) в (1), получим модель объекта с функцией управления, зависящей от состояния

$$\dot{\mathbf{s}}^i = \begin{cases} \mathbf{f}(\mathbf{s}^i, \mathbf{h}(\mathbf{s}^i)), & \text{если } R_o^2 - \sum_{c=1}^p (y_{i,c} - y_{j,c})^2 \geq 0, \\ \mathbf{f}(\mathbf{s}^i, \mathbf{g}(\mathbf{s}^i, \mathbf{s}^j)) & \text{иначе.} \end{cases} \quad (7)$$

Требуется найти адаптивную функцию управления $\mathbf{u}^i = \mathbf{g}(\mathbf{s}^i, \mathbf{s}^j)$, позволяющую объектам управления соблюсти динамические фазовые ограничения (4) и удовлетворяющую функционалу качества (5) для всех возможных начальных состояний (2).

Будем считать, что система управления $\mathbf{u}^i = \mathbf{h}(\mathbf{s}^i)$, доставляющая объекты в терминальные состояния уже реализована [15] и функционирует на борту каждого динамического объекта.

III. СИНТЕЗ ВЕРОЯТНОСТНОЙ СИСТЕМЫ РАЗРЕШЕНИЯ КОЛЛИЗИЙ

Если в области видимости динамического объекта появляется другой объект, то эти объекты вступают во взаимодействие и управление каждым из них осуществляется системой разрешения коллизий. В задаче избегания столкновений необходимо учитывать не только собственную динамику, но и динамику объекта-помехи, столкновение с которым необходимо избежать. Если в области видимости динамического объекта находятся сразу несколько других объектов, то учитывается состояние ближайшего из них. В случае, когда несколько объектов находятся в области видимости рассматриваемого объекта на одинаковом расстоянии, то будем считать, что для выбора объекта-помехи, относительно которого необходимо совершить маневр, недостаточно информации и в таком случае выбираем объект, идущий первым в списке рассматриваемых объектов-помех.

По условию динамические объекты могут обнаруживать другие объекты в своей локальной области, но не могут обмениваться информацией друг с другом. Если в момент времени t одновременно выполняются условия, задаваемые в виде неравенств

$$\begin{aligned} R_o^2 - \sum_{c=1}^m (y_{i,c} - y_{j,c})^2 &\geq 0, \\ R^2 - \sum_{c=1}^m (y_{i,c} - y_{j,c})^2 &< 0, \end{aligned} \quad (8)$$

то будем говорить, что объекты i и j находятся во взаимодействии без столкновения, $i \neq j$, $R_o > R$. Если в момент времени t выполняется условие, задаваемое неравенством

$$R^2 - \sum_{c=1}^m (y_{i,c} - y_{j,c})^2 \geq 0, \quad (9)$$

то объекты i и j в момент времени t претерпевают столкновение.

Система разрешения коллизий, рассматриваемая в данной работе, основывается на взаимодействии двух динамических объектов, так как столкновение большого количества объектов является частным случаем столкновения пары. Для учета динамики изменения относительного состояния двух динамических объектов определим отклонение его состояния от состояния объекта-помехи в текущий момент времени

$$\Delta \hat{\mathbf{s}}^i(t) = \mathbf{s}^i(t) - \mathbf{s}^j(t), \quad (10)$$

где $i = \{1, 2\}$; и в предшествующий момент времени

$$\Delta \hat{\mathbf{s}}^i(t_p) = \mathbf{s}^i(t_p) - \mathbf{s}^j(t_p), \quad (11)$$

где j – номер ближайшего объекта, находящегося в локальной области рассматриваемого динамического объекта, \mathbf{s}^i и \mathbf{s}^j – состояния рассматриваемого динамического объекта и его *ближайшего* объекта-помехи соответственно, t и t_p – текущий и предшествующий моменты времени соответственно. Чтобы объекты были способны избегать столкновения и одновременно продвигаться к терминальным состояниям, определим вектор отклонения текущего состояния рассматриваемого объекта от его терминального состояния

$$\Delta \mathbf{s}^i(t) = \mathbf{s}^i(t) - \mathbf{s}_f^i, \quad (12)$$

где \mathbf{s}_f^i – терминальное состояние динамического объекта i .

Из невязок (10), (11) и (12) формируем вектор

$$\xi^i(t) = \left[\Delta \hat{\mathbf{s}}^i(t_p) \quad \Delta \hat{\mathbf{s}}^i(t) \quad \Delta \mathbf{s}^i(t) \right]^T, \quad (13)$$

который будем подавать на вход системы разрешения коллизий объекта i .

Множество двух упорядоченных наборов, состоящих из начальных и конечных состояний двух динамических объектов

$$b = \left\{ \left(\mathbf{s}^i(t_0), \mathbf{s}^i(t_f) \right), \left(\mathbf{s}^j(t_0), \mathbf{s}^j(t_f) \right) \right\}, \quad (14)$$

где $i, j = \{1, 2\}$, $i \neq j$, будем называть сценарием. Если для такой пары выполняются условия (8), то будем говорить о сценарии взаимодействия динамических объектов без столкновений. Если для сценария выполняется условие (9), то мы имеем дело со сценарием со столкновениями.

Используем непрерывную динамическую систему стабилизации [15] в качестве системы управления, доставляющую динамические объекты в их терминальные

состояния и генерируем множество сценариев, приводящих объекты к столкновениям или вхождению во взаимодействие без столкновений. Использование сценариев взаимодействия без столкновений обусловлено тем, что искомая система должна учитывать оптимальность управления, поступающего от системы стабилизации. Поэтому, необходимо присутствие сценариев взаимодействия без столкновений, чтобы система разрешения коллизий также училась воспроизводить управление, снимаемое с выхода системы стабилизации.

На основе множеств (14) формируем множество сценариев

$$\mathbf{Y} = \left\{ b_q \right\}_{q=1}^L, \quad (15)$$

которое будем использовать для поиска замкнутой функции $\mathbf{u}^i = \mathbf{g}(\cdot)$ в системе (6).

В ходе функционирования группы динамических объектов возможны ситуации, в которых существует несколько возможных маршрутов объезда препятствия, а иногда эти маршруты вовсе могут быть равноценными с точки зрения функционала качества. Например, подобная ситуация может возникнуть для двух гусеничных роботов, движущихся напрямую навстречу друг другу и задача которых – попасть в начальное состояние движущегося навстречу робота.

Для реализации возможности учета системой разрешения коллизий альтернативных маршрутов объезда препятствий, предлагается использовать специальную нейронную сеть, порождающую смешанное вероятностное распределение [16]. Особенность этой нейронной сети состоит в том, что она позволяет моделировать любую функцию распределения условной вероятности. Функция плотности распределения на выходе нейронной сети формируется посредством линейной комбинации ядерных функций

$$p(\mathbf{u}^i | \mathbf{s}^i) = \sum_{g=1}^k \alpha_g(\mathbf{s}^i) \phi_g(\mathbf{u}^i | \mathbf{s}^i), \quad (16)$$

где k – количество ядерных функций (функций распределения), составляющих смешанное распределение, \mathbf{u}^i – вектор управления, \mathbf{s}^i – состояние динамического объекта с номером $i = \{1, 2\}$, $g = \overline{1, k}$. Параметры $\alpha_g(\mathbf{s}^i)$ представляют коэффициенты смешивания и являются априорными вероятностями выборки вектора управления \mathbf{u}^i из g -ой функции распределения. Функции $\phi_g(\mathbf{u}^i | \mathbf{s}^i)$ представляют плотности условной вероятности управления \mathbf{u}^i для g -го ядра. В качестве ядерных функций используем нормальное распределение в форме

$$\phi_g(\mathbf{u}^i | \mathbf{s}^i) = \frac{1}{\zeta} \exp \left\{ - \frac{\| \mathbf{u}^i - \boldsymbol{\mu}_g(\mathbf{s}^i) \|_2^2}{2 \boldsymbol{\sigma}_g(\mathbf{s}^i)^2} \right\}, \quad (17)$$

где $\zeta = (2\pi)^{m/2} \boldsymbol{\sigma}_g(\mathbf{s}^i)^m$, вектор $\boldsymbol{\mu}_g(\mathbf{s}^i)$ – математическое ожидание g -го ядра для i -го динамического объекта, $m = \dim \{ \mathbf{u}^i \}$, $i = \{1, 2\}$.

Функция (16) способна генерировать оптимальную плотность условной вероятности (17) для любого подаваемого на вход состояния \mathbf{s}^i . Для этого необходимо задать параметры смешанной модели: априорные коэффициенты смешивания $\alpha_g(\mathbf{s}^i)$, средние значения $\boldsymbol{\mu}_g(\mathbf{s}^i)$ и среднеквадратичные отклонения $\boldsymbol{\sigma}_g(\mathbf{s}^i)$ в виде функций от состояния \mathbf{s}^i . Эти зависимости можно получить на выходе многослойного персептрона, принимающего на вход состояние \mathbf{s}^i . Общее число выходов такой нейронной сети составляет

$$O = k(m + 2),$$

где k – количество ядерных функций. Достаточное число ядерных функций и многослойных персептрон с достаточным числом нейронов в скрытых слоях формируют нейронную сеть, которая может аппроксимировать сколь угодно точно любую функцию условной плотности вероятностей [16]. При этом априорные коэффициенты смешивания $\alpha_g(\mathbf{s}^i)$ должны удовлетворять равенству

$$\sum_{g=1}^k \alpha_g(\mathbf{s}^i) = 1,$$

что достигается посредством нормализации соответствующих выходов при помощи *softmax*-слоя [17]:

$$\alpha_g = \frac{\exp(z_g^\alpha)}{\sum_{d=1}^k \exp(z_d^\alpha)}. \quad (18)$$

В целях поиска оптимальных весовых коэффициентов, инициализируем нейронную сеть с начальными весовыми коэффициентами $\boldsymbol{\Omega}^0 \in \mathbb{R}^{a \times l \times w}$, a – число слоев аппроксиматора; l, w – число входов и выходов слоя $\omega = \overline{1, a}$ соответственно. Будем осуществлять поиск оптимальных функциональных связей $\mathbf{u}(\boldsymbol{\xi}_q^i)$ на множестве сценариев (15), обеспечивающих минимум функционала качества

$$J = \hat{t}_f + \sum_{q=1}^L \left(\|\Delta \mathbf{s}_q^i\|_2 + \|\Delta \mathbf{s}_q^j\|_2 + \int_0^{\hat{t}_f} \vartheta(\boldsymbol{\xi}_q^i) dt \right), \quad (19)$$

$$J \rightarrow \min_{\boldsymbol{\Omega}}.$$

где $i = 1, j = 2, \hat{t}_f = \max \{t_{f,i}\}_{i=1}^2$ – максимальное ограниченное время процесса управления, которое может быть задано или определяться по достижению терминальных условий,

$$\vartheta(\boldsymbol{\xi}_q^i) = \begin{cases} 1, & \text{если } R_o^2 - \|\mathbf{y}_q^i - \mathbf{y}_q^j\|_2^2 \geq 0, \\ 0, & \text{иначе,} \end{cases}$$

– функция Хэвисайда, накладывающая штраф за столкновение на функционал (19), $\mathbf{y}_q^i, \mathbf{y}_q^j$ – подвекторы состояний \mathbf{s}_q^i и \mathbf{s}_q^j соответственно.

Для поиска оптимальных весовых коэффициентов $\boldsymbol{\Omega}^o$ аппроксиматора используем метод эволюционных стратегий [18]. На каждой эпохе генерируем множество случайных, нормально распределенных векторов возмущения

$$\Phi = (\boldsymbol{\varepsilon}_1 \dots \boldsymbol{\varepsilon}_\lambda), \quad \boldsymbol{\varepsilon}_p = [\mathbf{X}_1, \dots, \mathbf{X}_a]^T. \quad (20)$$

Здесь

$$\mathbf{X}_y = N(0, I), \quad \mathbf{X}_y \in \square^{l \times w}, \quad (21)$$

где $z = \overline{1, \lambda}, y = \overline{1, a}$.

Суммируем матрицы возмущения (21) из множества (20) с текущей матрицей весовых коэффициентов $\boldsymbol{\Omega}^e$ и получаем матрицы возмущенных весовых коэффициентов

$$\tilde{\boldsymbol{\Omega}}_p^e = \boldsymbol{\Omega}^e + \sigma_{\boldsymbol{\Omega}} \boldsymbol{\varepsilon}_p,$$

где $\sigma_{\boldsymbol{\Omega}}$ – среднеквадратическое отклонение нормального распределения, из которого осуществляется выборка возмущений. Формируем множество возмущенных весовых коэффициентов

$$\mathbf{T} = (\tilde{\boldsymbol{\Omega}}_1^e \dots \tilde{\boldsymbol{\Omega}}_\lambda^e). \quad (22)$$

На множестве (22) проводим оценку значений функционала (19) на множестве сценариев (15) и получаем множество

$$\mathbf{A} = (J_1 \dots J_\lambda). \quad (23)$$

Проводим упорядочивание функционалов (23) по возрастанию:

$$\tilde{\mathbf{A}} = (\tilde{J}_1 \leq \dots \leq \tilde{J}_\lambda). \quad (24)$$

Далее, осуществляем ранговое преобразование значений функционала [19], подставляя порядковые номера функционалов в функцию полезности

$$\mathbf{F} = (f_1 \leq \dots \leq f_\lambda).$$

Здесь

$$f(z) = \frac{\max \left(0, \ln \left(\frac{\lambda}{2} + 1 \right) - \ln z \right)}{\sum_{p=1}^{\lambda} \max \left(0, \ln \left(\frac{\lambda}{2} + 1 \right) - \ln p \right)}, \quad (25)$$

где z – номер вектора возмущения, упорядоченного в соответствии с (24). График применяемой функции полезности изображен на рис. 1.

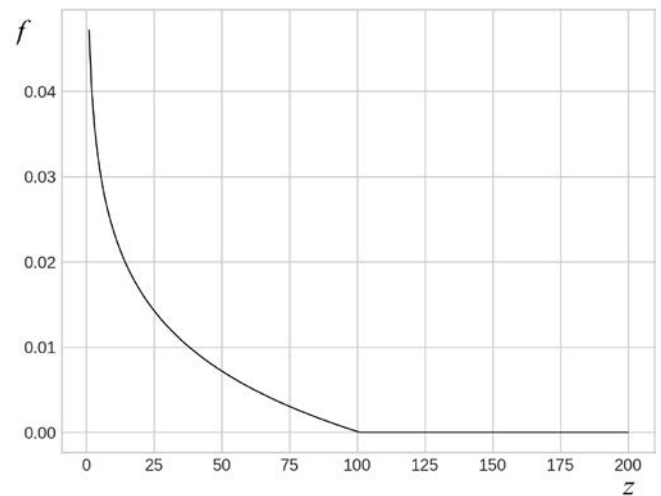


Рис. 1. Функция полезности (25).

Переход от значений оценок функционала к значениям функции полезности позволяет уменьшить влияние

некачественных векторов в популяции и устраняет проблему раннего схождения алгоритма в точке локального минимума [18]. Из рис. 1 можно сделать вывод, что функция полезности (25) отсеивает половину векторов возмущения, отдавая предпочтение более пригодным векторам возмущения.

Получаем новые весовые коэффициенты Ω^{e+1} :

$$\Omega^{e+1} \leftarrow \Omega^e - \frac{\alpha_\Omega}{\lambda \sigma_\Omega} \sum_{p=1}^{\lambda} f_p \varepsilon_p,$$

где α_Ω – шаг (скорость) обучения.

По завершении обучения получаем нейронную сеть, реализующую замкнутую вероятностную функцию управления

$$\mathbf{u}^i = \mathbf{h}(\xi^i(t), \Omega^o), \quad (26)$$

где Ω^o – обученные весовые коэффициенты нейронной сети.

При одновременном нахождении в локальной области видимости рассматриваемого динамического объекта (1) нескольких объектов-помех, определяется ближайший из них

$$\gamma_i(t) = \min_j \left\{ \|\mathbf{s}^i - \mathbf{s}^j\|_2 : i, j = \overline{1, N} \right\},$$

$$\beta_i(t) = \arg \min_j \left\{ \|\mathbf{s}^i - \mathbf{s}^j\|_2 : i, j = \overline{1, N} \right\},$$

где N – число динамических объектов в группе, $i \neq j$. Если несколько объектов-помех находятся на одном расстоянии от рассматриваемого объекта, то учитываем состояние объекта, идущего первым в списке рассматриваемых объектов-помех.

IV. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

В качестве объектов управления гомогенной группы рассматриваем гусеничных мобильных роботов [20], чья динамика описывается системой

$$\begin{cases} \dot{x}_i = \frac{(u_{i,1} + u_{i,2}) \cos(\theta_i)}{2}, \\ \dot{y}_i = \frac{(u_{i,1} + u_{i,2}) \sin(\theta_i)}{2}, \\ \dot{\theta}_i = \frac{(u_{i,1} - u_{i,2})}{2}, \end{cases} \quad (27)$$

где x_i, y_i – координаты центра масс роботов; θ_i – угол между осью симметрии роботов и осью абсцисс инерциальной системы координат Oxy ; $u_{i,1}, u_{i,2} \in [-1, 1]$ – управления гусеницами роботов, $i = \{1, 2\}$.

Определим множество начальных

$$\mathbf{S}_{0,i} = \left\{ \left(\mathbf{s}_q^{0,i}, \mathbf{s}_q^{0,j} \right) \right\}_{q=1}^L \quad (28)$$

и терминальных состояний

$$\mathbf{S}_{f,i} = \left\{ \left(\mathbf{s}_q^{f,i}, \mathbf{s}_q^{f,j} \right) \right\}_{q=1}^L \quad (29)$$

пар роботов, приводящих роботов к коллизиям и взаимодействиям без коллизий в пропорции 1:1, где

$$\mathbf{s}_q^{0,i} = \begin{bmatrix} x_{0,i_q} & y_{0,i_q} & \theta_{0,i_q} \end{bmatrix}^T, \quad \mathbf{s}_q^{0,j} = \begin{bmatrix} x_{0,j_q} & y_{0,j_q} & \theta_{0,j_q} \end{bmatrix}^T,$$

$$\mathbf{s}_q^{f,i} = \begin{bmatrix} x_{f,i_q} & y_{f,i_q} & \theta_{f,i_q} \end{bmatrix}^T, \quad \mathbf{s}_q^{f,j} = \begin{bmatrix} x_{f,j_q} & y_{f,j_q} & \theta_{f,j_q} \end{bmatrix}^T.$$

Здесь $L = 200$, $i, j = \{1, 2\}$, $i \neq j$. Величины

$\theta_* \in [0, 2\pi]$, $x_* \in [-10, 10]$, $y_* \in [-10, 10]$ равномерно случайно распределены в указанных интервалах.

Из (28) и (29) формируем множество сценариев обучения

$$\mathbf{Y} = \{b_q\}_{q=1}^L, \quad (30)$$

где $b_q = \left\{ \left(\mathbf{s}_q^{0,i}, \mathbf{s}_q^{0,j} \right), \left(\mathbf{s}_q^{f,i}, \mathbf{s}_q^{f,j} \right) \right\}$.

Вектор

$$\Delta \mathbf{s}_q^i(t) = \mathbf{s}_{q,f}^i - \mathbf{s}_{q,i}^i = \begin{bmatrix} x_{q,f,i} - x_{q,i} & y_{q,f,i} - y_{q,i} & \theta_{q,f,i} - \theta_{q,i} \end{bmatrix}^T \quad (31)$$

определяет отклонение текущего состояния i -го робота от его терминального состояния \mathbf{s}_f^i . Вектор

$$\Delta \hat{\mathbf{s}}_q^i(t) = \mathbf{s}_q^i - \mathbf{s}_q^j = \begin{bmatrix} x_{q,i} - x_{q,j} & y_{q,i} - y_{q,j} \end{bmatrix}^T \quad (32)$$

задает отклонение состояния i -го робота от состояния объекта-помехи j в момент времени $t \in [0, \hat{t}_f]$. Вектор

$$\Delta \hat{\mathbf{s}}_q^i(t_p) = \mathbf{s}_q^i(t_p) - \mathbf{s}_q^j(t_p) = \begin{bmatrix} x_{q,i} - x_{q,j} & y_{q,i} - y_{q,j} \end{bmatrix}^T \quad (33)$$

определяет отклонение i -го робота от состояния объекта-помехи в предшествующий момент времени $t_p \in [0, \hat{t}_f]$.

Из совокупности векторов (31), (32) и (33) формируем вектор

$$\xi_q^i(t) = \left[\Delta \hat{\mathbf{s}}_q^i, \Delta \hat{\mathbf{s}}_q^i(t_p), \Delta \mathbf{s}_q^i(t) \right]^T,$$

который будем подавать на вход системы разрешения коллизий.

Подлежащий минимизации функционал имеет вид

$$J = \hat{t}_f + \sum_{q=1}^{200} \left(\left\| \Delta \mathbf{s}_q^i \right\|_2 + \left\| \Delta \mathbf{s}_q^j \right\|_2 + \sum_{t=0}^{\hat{t}_f} \vartheta(\xi_q^i) \right), \quad (34)$$

где $i = 1, j = 2$, $t \in [0, \hat{t}_f]$, $t^+ = 5$ с – максимальное время симуляции движения пары роботов,

$$\hat{t}_f = \begin{cases} t, & \text{если } \max \{t_{f,i}\}_{i=1}^2 < t^+, \\ t^+ & \text{иначе.} \end{cases}$$

Здесь

$$\vartheta(\xi_q^i) = \begin{cases} 1, & \text{если } R^2 - (\square x_{ij})^2 - (\square y_{ij})^2 > 0, \\ 0 & \text{иначе,} \end{cases}$$

– функция Хэвисайда, накладывающая штраф за столкновение на функционал (34), где $\square x_{ij} = x_{q,i} - x_{q,j}$,

$\square y_{ij} = y_{q,i} - y_{q,j}$. Если выполняется условие

$$\left\| \mathbf{s}_{q,f}^i - \mathbf{s}_q^i \right\|_2 < \delta,$$

то полагаем, что i -ый робот (27) достиг терминального состояния. Здесь $\mathbf{s}_q^i(t) = [x_{q,i}(t) \ y_{q,i}(t) \ \theta_{q,i}(t)]^T$, $\delta = 0,01$ – допустимая точность попадания объекта в терминальное состояние.

В качестве аппроксиматора используем многослойный перцептрон с двумя скрытыми слоями по 80 нейронов в каждом. Мобильный робот функционирует на плоскости и способен объезжать препятствия с двух сторон, поэтому минимальное число ядерных функций должно быть не менее двух. В данном эксперименте используем $k = 4$.

Алгоритм функционирования первого слоя нейронной сети можно записать в виде

$$\mathbf{y}_1^i = \begin{bmatrix} y_{i,1}^1 \\ \vdots \\ y_{i,1}^{80} \end{bmatrix} = f(\mathbf{W}_1 \xi^i) = \begin{bmatrix} f\left(\left(\mathbf{w}_1^1\right)^T \xi^i\right) \\ \vdots \\ f\left(\left(\mathbf{w}_1^{80}\right)^T \xi^i\right) \end{bmatrix}, \quad (35)$$

$$\mathbf{W}_1 = \begin{bmatrix} \mathbf{w}_1^1 \\ \vdots \\ \mathbf{w}_1^{80} \end{bmatrix} = \begin{bmatrix} w_{1,1,1} & \cdots & w_{1,1,7} \\ \vdots & \ddots & \vdots \\ w_{1,80,1} & \cdots & w_{1,80,7} \end{bmatrix},$$

где \mathbf{W}_1 – матрица весовых коэффициентов первого слоя, \mathbf{y}_1 – выходной вектор первого слоя. Выходной вектор этого слоя является входным вектором для второго слоя, так что аналогично имеем:

$$\mathbf{y}_2^i = \begin{bmatrix} y_{i,2}^1 \\ \vdots \\ y_{i,2}^{80} \end{bmatrix} = f(\mathbf{W}_2 \mathbf{y}_1^i) = \begin{bmatrix} f\left(\left(\mathbf{w}_2^1\right)^T \mathbf{y}_1^i\right) \\ \vdots \\ f\left(\left(\mathbf{w}_2^{80}\right)^T \mathbf{y}_1^i\right) \end{bmatrix}, \quad (36)$$

$$\mathbf{W}_2 = \begin{bmatrix} \mathbf{w}_2^1 \\ \vdots \\ \mathbf{w}_2^{80} \end{bmatrix} = \begin{bmatrix} w_{2,1,1} & \cdots & w_{2,1,80} \\ \vdots & \ddots & \vdots \\ w_{2,80,1} & \cdots & w_{2,80,80} \end{bmatrix}.$$

В выражениях (35), (36) функция $f(\cdot)$ – функция активации. Каждая компонента функции $f(\cdot)$ имеет вид $\varphi(x) = \max(0, x)$. Функционирование последнего слоя отличается тем, что компоненты выходного вектора подаются на разные функции активации. В результате скалярного произведения матрицы весовых коэффициентов предпоследнего слоя на выходной вектор слоя (36) получаем вектор

$$\tilde{\mathbf{v}}^i = \mathbf{W}_3 \mathbf{y}_2^i = \begin{bmatrix} \left(\mathbf{w}_3^1\right)^T \mathbf{y}_2^i \\ \vdots \\ \left(\mathbf{w}_3^c\right)^T \mathbf{y}_2^i \end{bmatrix},$$

$$\mathbf{W}_3 = \begin{bmatrix} w_{3,1,1} & \cdots & w_{3,1,80} \\ \vdots & \ddots & \vdots \\ w_{3,0,1} & \cdots & w_{3,0,80} \end{bmatrix},$$

или

$$\tilde{\mathbf{v}}^i = [\tilde{\alpha}_{i,1} \dots \tilde{\alpha}_{i,k} \ \tilde{\mu}_{i,k+1} \dots \tilde{\mu}_{i,c} \ \tilde{\sigma}_{i,c+1} \dots \tilde{\sigma}_{i,o}]^T, \quad (37)$$

где $c = k + 1 + mk$.

В соответствии с уравнением (18) проводим нормализацию первых k компонент вектора (37) и получаем априорные коэффициенты смешивания

$$\mathbf{a}^i = [\alpha_{i,1} \dots \alpha_{i,k}]^T.$$

Следующие k компонент подаем на вход гиперболического тангенса и получаем математические ожидания ядерных функций

$$\mathbf{m}^i = [\mu_{i,1} \dots \mu_{i,m \times k}]^T.$$

$$\text{Здесь } \mu_{i,y} = \frac{\exp(\tilde{\mu}_{i,y}) - \exp(-\tilde{\mu}_{i,y})}{\exp(\tilde{\mu}_{i,y}) + \exp(-\tilde{\mu}_{i,y})}, \quad y = \overline{1, m \times k}.$$

Последние k компонент вектора (37) подаем на сигмоидальную функцию и получаем среднеквадратичные отклонения плотностей распределения вероятностей

$$\mathbf{d}^i = [\sigma_{i,1} \dots \sigma_{i,k}]^T.$$

$$\text{Здесь } \sigma_{i,y} = \frac{\exp(\tilde{\sigma}_{i,y})}{1 + \exp(\tilde{\sigma}_{i,y})}, \quad y = \overline{1, k}.$$

Для инициализации алгоритма поиска оптимальных весовых коэффициентов Ω^0 задаем скорость обучения $\alpha_\Omega = 0,01$, стандартное отклонение $\sigma_\Omega = 0,2$, количество эпох обучения $E = 60$, количество векторов возмущения $\lambda = 1000$, количество тренировочных $c = 200$ и тестовых $m = 50$ сценариев, начальные весовые коэффициенты Ω_0 . Генерируем множества тестовых сценариев с коллизиями и с взаимодействиями пар динамических объектов без коллизий $D \leftarrow \{\mathbf{d}_l : l \in \{1, \dots, m\}\}$, $Z \leftarrow \{\mathbf{z}_l : l \in \{1, \dots, m\}\}$. На каждой эпохе выполняем следующие 9 шагов:

1. Генерируем смешанное множество тренировочных сценариев $Y \leftarrow \{\mathbf{y}_l : l \in \{1, \dots, c\}\}$, половина которых задают условия столкновения пары и другая половина – взаимодействие без столкновений;

2. Генерируем векторы возмущений $\Phi = (\boldsymbol{\varepsilon}^1, \dots, \boldsymbol{\varepsilon}^\lambda)$;

3. Вычисляем на тренировочных сценариях значения функционала (34) для возмущенных весовых коэффициентов $A = \left(J_Y^1(\tilde{\Omega}_1^e) \dots J_Y^\lambda(\tilde{\Omega}_p^e) \right)$, где

$$\tilde{\Omega}_p^e = \Omega^e + \sigma_\Omega \boldsymbol{\varepsilon}_p;$$

4. Упорядочиваем значения функционалов

$$\mathbb{A} = \left(J_Y^\eta \leq \dots \leq J_Y^\zeta \right);$$

5. Вычисляем значения функции полезности

$$F = \left(\varphi^1 \leq \dots \leq \varphi^\lambda \right), \quad \text{где } \varphi^1 = f(\eta), \quad \varphi^\lambda = f(\zeta);$$

6. Оставляем γ лучших возмущений $\tilde{\Phi} = (\tilde{\boldsymbol{\varepsilon}}^1 \dots \tilde{\boldsymbol{\varepsilon}}^\gamma)$;

7. Корректируем весовые коэффициенты

$$\Omega^{e+1} \leftarrow \Omega^e - \alpha \frac{1}{\lambda \sigma_\Omega} \sum_{q=1}^\gamma \varphi^q \tilde{\boldsymbol{\varepsilon}}^q.$$

В завершении процесса обучения получаем обученные весовые коэффициенты нейронной сети Ω^0 .

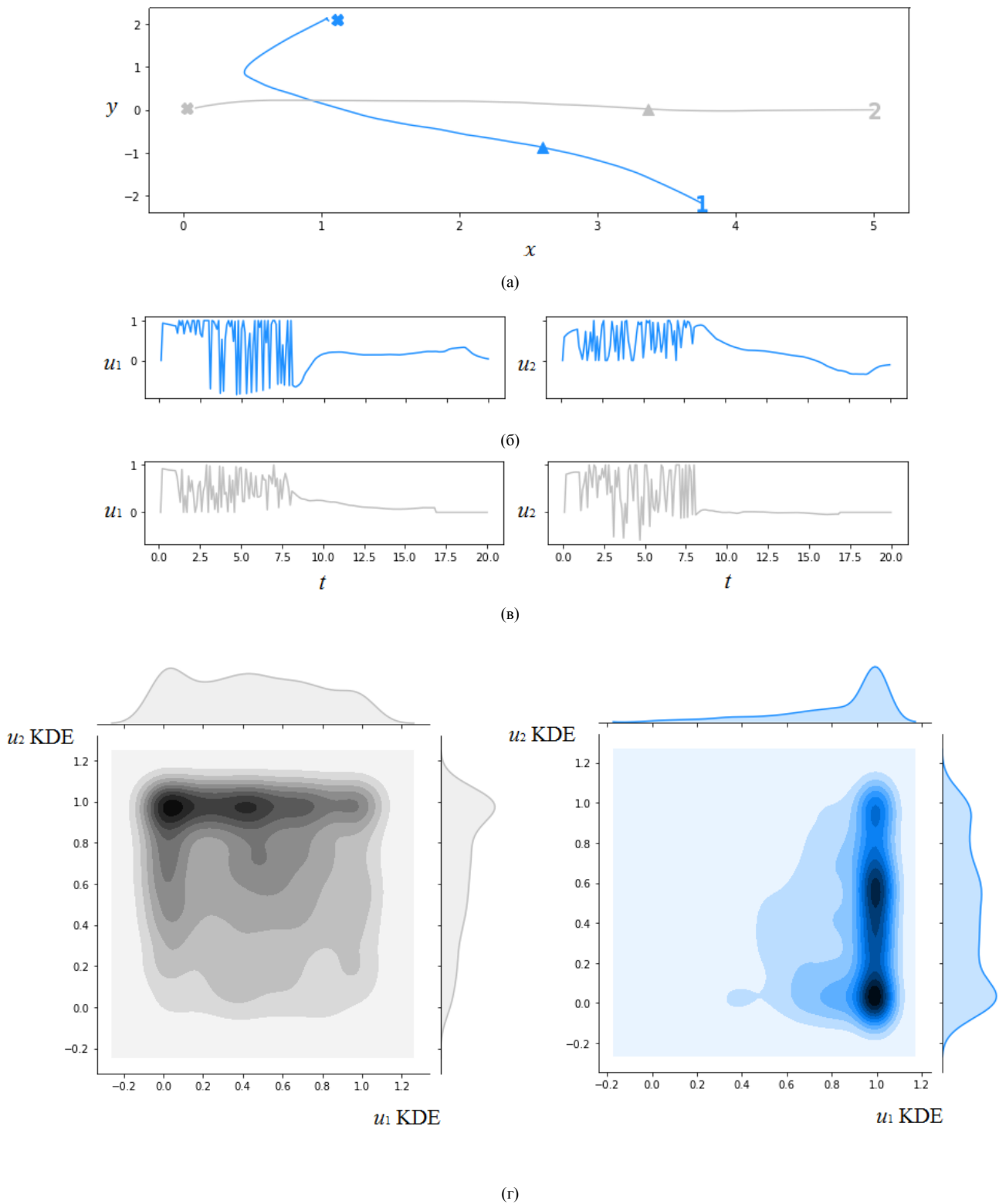


Рис. 2. Сценарий с нарушением динамических фазовых ограничений: (а) траектории роботов; (б) управления на гусеницах первого робота; (в) управления на гусеницах второго робота; (г) ядерные оценки условных плотностей вероятностей управляющих воздействий.

Рис. 2 отражает тестовый сценарий коллизии двух роботов, не входящий в обучающую выборку Y . На рис. 2 номерами 1 и 2 обозначены начальные состояния роботов, крестами – их конечные состояния. Пунктирной линией обозначены участки траекторий, где роботы сталкиваются. Треугольником на графиках а) обозначены начальные моменты времени, когда роботы вступают во взаимодействие. Для соответствующего момента времени, на графике (г) отмечены оценочные плотности вероятностей выходных управлений, где KDE – kernel density estimate (ядерная оценка плотности).

Для получения ядерных оценок плотностей вероятностей выборка управлений проводилась нейронной сетью 10000 раз. Из рис. 2 (а) можно сделать вывод, что робот под номером 1 совершает маневр, отклоняющий его от маршрута, без которого произошло бы столкновение с роботом под номером 2. На рис. 2 (б), (в) можно видеть, что функции управления не имеют скачков. Двумерные ядерные оценки плотности условных вероятностей демонстрируют вероятностный характер управлений, зависящих от текущего состояния объекта и его помехи.

Найденная система разрешения коллизий является универсальной по двум причинам: во-первых, она реализована в виде замкнутой функции, во-вторых, она учитывает все возможные варианты решения задачи разрешения коллизий. Несмотря на то, что данная система обучается на сценариях парного взаимодействия роботов, она может эксплуатироваться в группах, состоящих из $N > 2$ объектов. Совместное ее использование с системой стабилизации динамической системы [15] формирует полноценную систему управления группой динамических объектов.

I. ЗАКЛЮЧЕНИЕ

Интеллектуальные системы управления должны проектироваться с учетом возможного попадания динамического объекта в неоднозначные ситуации. В статье реализована и протестирована система разрешения коллизий на основе архитектуры нейронной сети, осуществляющей вероятностное отображение состояния в управляющее воздействие.

Предложенная двухконтурная система управления избавляет роботов от вычисления оптимальных управлений в реальном масштабе времени, из-за чего отсутствует задержка между моментом времени получения роботом информации о среде и моментом времени, когда робот совершает действие.

В качестве дальнейшей работы планируется обучение с разными значениями радиусов взаимодействия. Также планируется добавить дополнительное условие выхода динамических объектов из взаимодействия. Взаимное отдаление роботов может служить дополнительным

условием для прекращения управления роботами системой разрешения коллизий.

БИБЛИОГРАФИЯ

- [1] Barraquand J., Latombe J. C. *Robot motion planning: A distributed representation approach* // The International Journal of Robotics Research. – 1991. – Vol. 10, no. 6. – P. 628–649.
- [2] LaValle S. M. *Planning algorithms* / – Cambridge university press, 2006. – 842 p.
- [3] Kapadia M., Singh S., Hewlett W., Reinman G., Faloutsos, P. *Parallelized egocentric fields for autonomous navigation* // The Visual Computer. – 2012. – Vol. 28, no. 12. – P. 1209–1227.
- [4] Park J. W., Kwak H. J., Kang Y. C., Kim D. W. *Advanced fuzzy potential field method for mobile robot obstacle avoidance* // Computational intelligence and neuroscience. – 2016. – Vol. 2016.
- [5] O'Donnell P. A., Lozano-Pérez T. *Deadlock-free and collision-free coordination of two robot manipulators* // ICRA. Vol. 89. – 1989. – P. 484–489.
- [6] Jansen M. R., Sturtevant N. R. *Direction Maps for Cooperative Pathfinding*. // AIIDE. – 2008. – Vol. 2008. – P. 185.
- [7] Hennes D., Claes D., Meeussen W., Tuyls, K. *Multi-robot collision avoidance with localization uncertainty* // AAMAS. – 2012. – P. 147–154.
- [8] Alejo D., Cobano J. A., Heredia G. *Reactive method for collision avoidance in industrial environments* // Journal of Intelligent & Robotic Systems. – 2016. – Vol. 84, no. 1–4. – P. 745–758
- [9] Diveev A. I., Shmalko E. Y. *Self-adjusting control for multi robot team by the network operator method* // 2015 European Control Conference (ECC). – IEEE. 2015. – P. 709–714.
- [10] Diveev A. I., Shmalko E. Y. *Optimal motion control for multi-robot system by multilayer network operator* // 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). – IEEE. 2016. – P. 2164–2169.
- [11] Diveev A., Shmalko E. *Hybrid evolutionary algorithm for synthesized optimal control problem for group of interacting robots* // 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT). – IEEE. 2019. – P. 876–881.
- [12] Diveev, A., Hussein, O., Shmalko, E., Sofronova, E. *Synthesis of Control System for Quad-Rotor Helicopter by the Network Operator Method* // Proceedings of SAI Intelligent Systems Conference. — Springer. 2020. – P. 246–263.
- [13] Diveev. A. *Cartesian Genetic Programming for Synthesis of Control System for Group of Robots* // 2020 28th Mediterranean Conference on Control and Automation (MED). – IEEE. 2020. – P. 972–977.
- [14] Дубовицкий А. Я., Милотин А. А. *Задачи на экстремум при наличии ограничений* // Журнал вычислительной математики и математической физики. – 1965. – Т. 5, no. 3. – С. 395–453.
- [15] Доценко А. В. *Автоматический синтез непрерывной динамической системы стабилизации на основе искусственных нейронных сетей* // Вестник МГТУ им. Н. Э. Баумана. Серия Приборостроение. – 2020. – no.3. – С. 66-83.
- [16] Bishop C. M. *Mixture density networks*. – 1994.
- [17] Wang M., Lu S., Zhu D., Lin J., Wang Z. *A high-speed and low-complexity architecture for softmax function in deep learning* // 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). – IEEE, 2018. – С. 223–226.
- [18] Salimans T., Ho J., Chen X., Sidor S., Sutskever I. *Evolution strategies as a scalable alternative to reinforcement learning* // arXiv preprint arXiv:1703.03864. – 2017.
- [19] Wierstra D. et al. *Natural evolution strategies* // The Journal of Machine Learning Research. – 2014. – Vol. 15, no. 1. – P. 949–980.
- [20] Dubins L. E. *On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents* // Amer. J. Math., 1957, Vol. 79. P. 497-516.

Synthesis of the probabilistic collision avoidance system in the domain of dynamic objects motion

A. V. Dotsenko

Abstract—The article is devoted to the control system synthesis problem considering collision avoidance system in the task of collective regrouping of dynamic objects. Our goal is to find the closed loop control system with probabilistic mapping of the current state vector to the optimal control vector. We propose to consider the control of pairs of dynamic objects, as simultaneous collision of more than 2 multiple objects can be decomposed to build a set of individual pairs of objects for which the violation of phase dynamic constraints is taking place. Control system is realized as a conditional density function with control as a random variable conditioned on the states of both dynamic object and moving obstacle, which reflects the novelty of the research. Proposed control system is universal in a sense that it is able to resolve collisions between dynamic objects even in the unknown environment for arbitrary initial and terminal states given for objects. Moreover, it is able to consider alternate obstacle avoidance routes through the probability density function of control. The control system also assumes the control of the group is decentralized. Individual object will be acting as a standalone system after the control system is uploaded in the memory.

Keywords – collision avoidance, perceptron, control synthesis, evolution strategies, mixture probability density.

REFERENCES

- [1] Barraquand J., Latombe J. C. *Robot motion planning: A distributed representation approach* // The International Journal of Robotics Research. – 1991. – Vol. 10, no. 6. – P. 628–649.
- [2] LaValle S. M. *Planning algorithms* / – Cambridge university press, 2006. – 842 p.
- [3] Kapadia M., Singh S., Hewlett W., Reinman G., Faloutsos, P. *Parallelized egocentric fields for autonomous navigation* // The Visual Computer. – 2012. – Vol. 28, no. 12. – P. 1209–1227.
- [4] Park J. W., Kwak H. J., Kang Y. C., Kim D. W. *Advanced fuzzy potential field method for mobile robot obstacle avoidance* // Computational intelligence and neuroscience. – 2016. – Vol. 2016.
- [5] O'Donnell P. A., Lozano-Pérez T. *Deadlock-free and collision-free coordination of two robot manipulators* // ICRA. Vol. 89. – 1989. – P. 484–489.
- [6] Jansen M. R., Sturtevant N. R. *Direction Maps for Cooperative Pathfinding*. // AIIDE. – 2008. – Vol. 2008. – P. 185.
- [7] Hennes D. et al. *Multi-robot collision avoidance with localization uncertainty* // AAMAS. – 2012. – P. 147–154.
- [8] Alejo D., Cobano J. A. *Heredia G. Reactive method for collision avoidance in industrial environments* // Journal of Intelligent & Robotic Systems. – 2016. – Vol. 84, no. 1–4. – P. 745–758
- [9] Diveev A. I., Shmalko E. Y. *Self-adjusting control for multi robot team by the network operator method* // 2015 European Control Conference (ECC). – IEEE. 2015. – P. 709–714.
- [10] Diveev A. I., Shmalko E. Y. *Optimal motion control for multi-robot system by multilayer network operator* // 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). – IEEE. 2016. – P. 2164–2169.
- [11] Diveev A., Shmalko E. *Hybrid evolutionary algorithm for synthesized optimal control problem for group of interacting robots* // 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT). – IEEE. 2019. – P. 876–881.
- [12] Diveev, A., Hussein, O., Shmalko, E., Sofronova, E. *Synthesis of Control System for Quad-Rotor Helicopter by the Network Operator Method* // Proceedings of SAI Intelligent Systems Conference. — Springer. 2020. – P. 246–263.
- [13] Diveev. A. *Cartesian Genetic Programming for Synthesis of Control System for Group of Robots* // 2020 28th Mediterranean Conference on Control and Automation (MED). – IEEE. 2020. – P. 972–977.
- [14] Dubovitskiy A. Y., Milyutin A. A. *Extremum problems in the presence of constraints* // Journal of numerical mathematics and mathematical physics. – 1965. – Vol. 5, no. 3. – C. 395–453.
- [15] Dotsenko A. V. *Automatic synthesis of a continuous dynamic stabilization system based on artificial neural networks* // Herald of the Bauman Moscow state technical university, Instrument Engineering. – 2020. – no.3. – C. 66-83.
- [16] Bishop C. M. *Mixture density networks*. – 1994.
- [17] Wang M., Lu S., Zhu D., Lin J., Wang Z. *A high-speed and low-complexity architecture for softmax function in deep learning* // 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). – IEEE, 2018. – C. 223–226.
- [18] Salimans T., Ho J., Chen X., Sidor S., Sutskever I. *Evolution strategies as a scalable alternative to reinforcement learning* // arXiv preprint arXiv:1703.03864. – 2017.
- [19] Wierstra D. et al. *Natural evolution strategies* // *The Journal of Machine Learning Research*. – 2014. – Vol. 15, no. 1. – P. 949–980.
- [20] Dubins L. E. *On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents* // *Amer. J. Math.*, 1957, Vol. 79. P. 497-516.