

## Моделирование группового управления сельскохозяйственными роботами с использованием конечных автоматов и онтологий

М. А. Шереужев<sup>1</sup>, Ф. В. Девяткин<sup>1</sup>, Д. И. Арабаджиев<sup>1</sup>, М. А. Шереужев<sup>2,3</sup>

<sup>1</sup>Московский государственный технический университет им. Н. Э. Баумана  
105005, Россия, Москва, 2-я Бауманская улица, 5

<sup>2</sup>Институт информатики и проблем регионального управления –  
филиал Кабардино-Балкарского научного центра Российской академии наук  
360000, Россия, г. Нальчик, ул. И. Арманд, 37-а

<sup>3</sup>Кабардино-Балкарский государственный аграрный университет им. В. М. Кокова  
360030, Россия, г. Нальчик, пр. Ленина, 1в

**Аннотация.** В статье рассмотрен способ моделирования работы системы управления группой роботов. Предложена база знаний системы группового управления, представленная в виде онтологии. Описаны конечные автоматы ведущего и ведомого роботов. Рассмотрен алгоритм назначения задач отдельным роботам и сконфигурированы инструменты моделирования управления движением и распределением задач для обеспечения единой программной среды моделирования группового управления. Рассмотрен венгерский алгоритм в контексте назначения задач отдельным роботам, для оценки путей движения использовался планировщик A\* (Astar). Сконфигурированы инструменты моделирования управления движением и распределением задач для обеспечения единой программной среды моделирования группового управления. Полученные результаты применимы в задачах оптимизации взаимодействия многоагентных систем.

**Ключевые слова:** групповое управление, онтология, робототехническая система, конечный автомат

Поступила 24.11.2023, одобрена после рецензирования 04.12.2023, принята к публикации 08.12.2023

**Для цитирования.** Шереужев М. А., Девяткин Ф. В., Арабаджиев Д. И., Шереужев М. А. Моделирование группового управления сельскохозяйственными роботами с использованием конечных автоматов и онтологий // Известия Кабардино-Балкарского научного центра РАН. 2023. № 6(116). С. 247–263. DOI: 10.35330/1991-6639-2023-6-116-247-263

MSC: 68T40; 93C85

Original article

## Simulation of group management of agricultural robots using finite state machines and ontologies

M.A. Shereuzhev<sup>1</sup>, F.V. Devyatkin<sup>1</sup>, D.I. Arabadjiev<sup>1</sup>, M.A. Shereuzhev<sup>2,3</sup>

<sup>1</sup>Moscow State Technical University named after N. E. Bauman  
105005, Russia, Moscow, 2nd Baumanskaya street 5

<sup>2</sup>Institute of Computer Science and Problems of Regional Management –  
branch of Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences  
360000, Russia, Nalchik, 37-a I. Armand street

<sup>3</sup>Kabardino-Balkarian State Agrarian University named after V.M. Kokov  
360030, Russia, Nalchik, 1v Lenin avenue

**Abstract.** The article discusses a method for modeling the operation of a control system for a group of robots. A knowledge base of a group control system is proposed, presented in the form of ontology. The finite state machines of the leading and driven robots are described. An algorithm for assigning tasks to

individual robots is considered and tools for modeling motion control and task distribution are configured to provide a unified software environment for modeling group control. The Hungarian algorithm is considered in the context of assigning tasks to individual robots; the A\* (Astar) planner was used to estimate movement paths. Tools for modeling motion control and task distribution have been configured to provide a unified software environment for modeling group control. The results obtained are applicable in problems of optimizing the interaction of multi-agent systems.

**Keywords:** multirobot control system, ontology, robotic system, finite state machine

Submitted 24.11.2023,

approved after reviewing 04.12.2023,

accepted for publication 08.12.2023

**For citation.** Shereuzhev M.A., Devyatkin F.V., Arabadjiev D.I., Shereuzhev M.A. Simulation of group management of agricultural robots using finite state machines and ontologies *News of the Kabardino-Balkarian Scientific Center of RAS*. 2023. No. 6(116). Pp. 247–263. DOI: 10.35330/1991-6639-2023-6-116-247-263

## ВВЕДЕНИЕ

На сегодняшний момент наблюдается активное стремление к разработке средств автоматизации в областях человеческой деятельности, где прежде существовала только частичная механизация, как, например, в сельском хозяйстве, в части выращивания плодово-овощных культур [1]. Многообразие производственных задач в данной области и высокие издержки на трудовые ресурсы обуславливают актуальность разработки универсальных робототехнических платформ с модульным быстроразъемным технологическим оборудованием. При выполнении операций, связанных с выращиванием и сбором плодовых культур, необходимо решение как общих задач робототехники, так и специализированных задач. Общие операции робототехники включают, например, навигацию по местности, локализацию, и распознавание объектов и др. [2]. Специализированные операции включают такие задачи, как срыв плодов, планирование траектории и управление позиционированием распылителя для нанесения препаратов.

Для эффективного выполнения задач требуется реализация цепочек операций, включая как типичные для большинства роботов, так и специализированные. Это предполагает формирование знаний о типах задач, требованиях к роботам и их возможностях для решения различных операций. Для группового применения роботов необходимо наличие семантических знаний, а также обоснование последовательности действий каждого робота на этапе распределения задачи и формирования последовательности состояний на логическом уровне системы управления отдельного робота [3]. В этом контексте методы представления знаний на основе онтологий, а также их комбинирование с управлением состояниями роботов на основе конечных автоматов [4, 5], представляются подходящими инструментами, предоставляющими возможность для решения сложных задач в реальной среде.

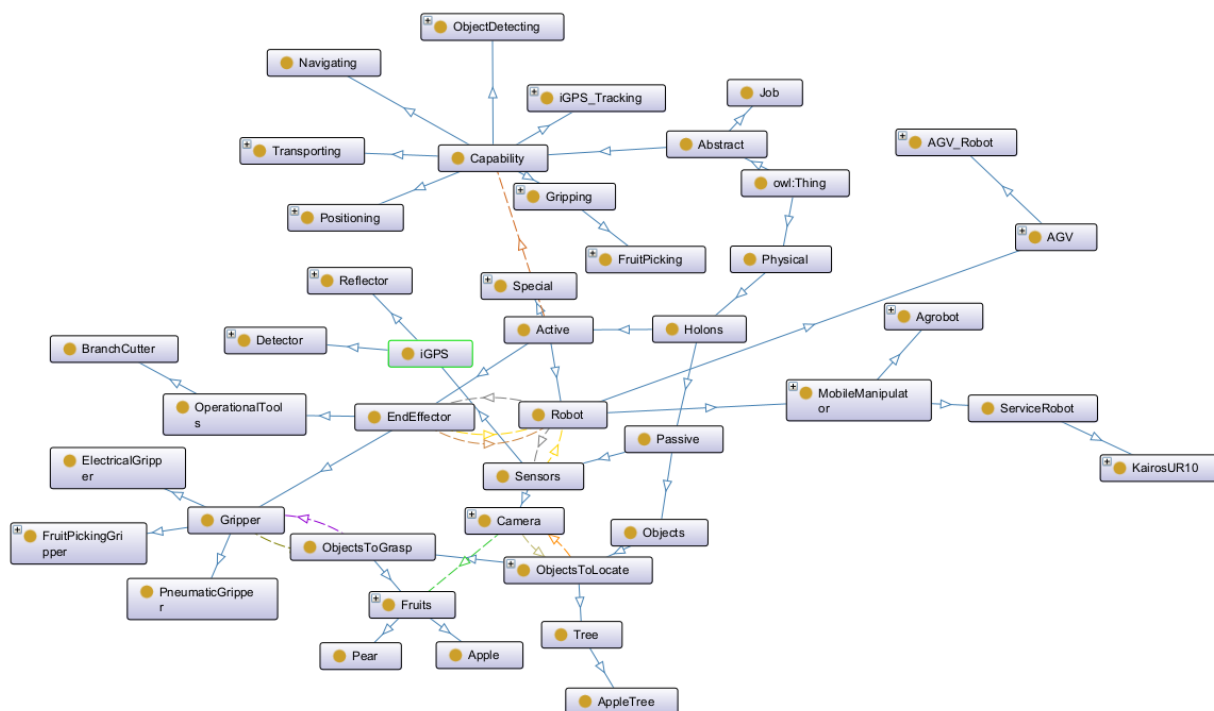
## ОНТОЛОГИЯ РОБОТОТЕХНИЧЕСКОЙ СИСТЕМЫ

Реализуем онтологию в составе алгоритма распределения задач для группы роботов, осуществляющих операцию сбора плодов в саду интенсивного выращивания. Определим набор классов для задач и роботов, таких как «СборПлодов», «ПеревозкаКоробок», «ЗарядкаАккумулятора», «РоботСборщик», «РоботЛогист» и «РоботЗарядка». Каждой задаче и роботу можно назначить набор атрибутов и возможностей, таких как «скорость\_сбора», «емкость\_хранилища», «объем\_перевозки», «расстояние\_до\_целевой\_точки» и «состояние\_зарядки». Также определим набор отношений между задачами и роботами, например, «совместимость\_с», «конфликт\_с», «зависит\_от» и «близость\_к» (табл. 1).

**Таблица 1.** Элементы онтологии группы роботов для сбора яблок**Table 1.** Ontology elements of a group of apple-picking robots

	<b>Задачи</b>	<b>Роботы</b>
<b>Классы</b>	СборПлодов, ПеревозкаКоробок, ЗарядкаАккумулятора	РоботСборщик РоботЛогист РоботЗарядка
<b>Атрибуты</b>	скорость_сбора, вместимость_хранилища, объем_перевозки, расстояние_до_целевой_точки, требуемый_заряд	состояние_зарядки остаток_заряда расстояние_до_целевой_точки
<b>Отношения</b>		совместимость_с конфликт_с зависит_от близость_к

Для реализации данной онтологии воспользуемся инструментом на базе OWL. Итоговая интерпретация онтологии, представленная на рисунке 1, учитывает наличие таких абстракций, как «способности», «рабочий орган», «объект манипулирования».



**Рис. 1.** Онтология для сельскохозяйственного робота с учетом возможных конфигураций  
(Источник: составлено авторами)

**Fig. 1.** Ontology for an agricultural robot taking into account possible configurations  
(Source: made by the authors)

В экземплярах классов данной онтологии будем хранить данные о требуемых параметрах выполнения задачи и данные о текущем состоянии роботов.

#### ПРИМЕР МЕТОДА РАСПРЕДЕЛЕНИЯ ЗАДАЧ

Основная цель заключается в координированном движении группы роботов, которая следует заданным маршрутом [6] и выполняет задачи по уборке урожая в определенном месте [7] в соответствии с заранее определенной логикой.

Используя данные о возможностях роботов и данные об актуальном состоянии, возможно сформировать алгоритм определения оптимального назначения задач [8–11].

Назначение задач гомогенным группам роботов является задачей о назначениях за полиномиальное время. Формализация данной задачи в виде графа предполагает, что в графе  $G = (S, T; E)$  есть  $n$  вершин, соответствующих гомогенным роботам в случае распределения на уровне верхнего вычислителя и отдельным агентам в случае гомогенной группы ( $S$ ), и  $n$  вершин, соответствующих видам работ  $T$ .

Пусть каждая задача  $T_i$  имеет набор требований  $req_i$  и набор ограничений  $con_i$ , а каждый робот  $r_j$  имеет набор возможностей  $cap_j$  и набор ограничений  $con_j$ . Решим проблему распределения задачи.

Для наглядности здесь и ниже полезность решения задачи  $t_i$  роботом  $r_j$  будем рассматривать в упрощенном виде как  $c_{i,j}$ , которая может быть рассчитана на основе совместимости требований задачи и возможностей робота, а также ограничений задачи. Например, если задача  $t_i$  требует возможности собирать яблоки, а робот  $r_j$  имеет возможность собирать яблоки, но срок службы батареи робота  $r_j$  низкий и он не может выполнить задачу без подзарядки, то стоимость назначения задачи  $t_i$  для робота  $r_j$  должна отражать как соответствие возможностей, так и ограничение батареи.

Для достижения оптимального режима работы РТК требуется найти совершенное или полное паросочетание с наименьшей стоимостью.

Венгерский алгоритм можно использовать для решения задачи о назначениях, находя такое задание, которое минимизирует общую стоимость [12]. Алгоритм сначала инициализирует матрицу  $M$  размера  $m \times n$ , где каждый элемент представляет стоимость назначения задачи  $t_i$  роботу  $r_j$ . Затем алгоритм итеративно сводит матрицу к квадратной матрице размера  $m \times m$ , где каждая строка и столбец содержат ровно один нулевой элемент, путем вычитания элемента минимальной стоимости каждой строки и столбца из всех остальных элементов. Затем алгоритм идентифицирует набор назначений, которые охватывают все нулевые элементы, и если количество назначений меньше  $m$ , алгоритм итеративно увеличивает набор назначений, находя новые назначения, которые охватывают непокрытые нулевые элементы, пока не будут назначены все задачи. Полное описание приведено на рисунке 2.

### 1. Инициализация

Пусть  $M$  – матрица размера  $m \times n$ , где  $M_{i,j}$  представляет стоимость назначения задачи  $t_i$  роботу  $r_j$ . Установим набор назначений  $A$  пустым.

#### Уменьшение матрицы

Для каждой строки  $i$  и каждого столбца  $j$  таблицы  $M$  необходимо вычесть минимальный элемент строки  $i$  и столбца  $j$  из всех элементов строки  $i$  и столбца  $j$  соответственно. Обозначим полученную матрицу как  $M'$ .

### 2. Определение задания

Пока размер  $A$  меньше  $m$ , сделать следующие шаги.

Найти нулевой элемент  $M'_{i,j}$ , который не покрывается никаким присваиванием в  $A$ .

Если существуют уникальная строка  $i$  и уникальный столбец  $j$  такие, что  $M'_{i,j} = 0$ , то назначаем задачу  $t_i$  роботу  $r_j$  и добавляем задание  $(t_i, r_j)$  в  $A$ .

В противном случае пусть  $S_i$  будет набором столбцов, в которых нет элементов в строке  $i$ , и пусть  $T_j$  будет набором строк, в которых нет элементов в столбце  $j$ . Пусть  $S$  – объединение  $S_i$  и  $T_j$ . Если  $S$  не пусто, выбрать любой столбец  $k$ .

Если  $S$  пусто, алгоритм завершается без действительного присваивания.

В противном случае выбрать любую строку  $p$  в  $S$  и добавить присваивание  $(t_p, r_k)$  к  $A$ . Удалить все назначения, которые охватывают любую строку или столбец, которые покрыты назначением  $(t_p, r_k)$ . Вернуться к шагу 2.

### 3. Выход

Набор заданий  $A$  представляет собой оптимальное распределение заданий, минимизирующее общую стоимость.

**Рис. 2.** Описание венгерского алгоритма для распределения задач

**Fig. 2.** Description of the Hungarian algorithm for task distribution

С учетом особенностей функционирования роботов предложим следующую структуру стоимости назначения задачи  $t_i$  для робота  $r_j$ :

$$c_{i,j} = w_1 \times d_{i,j} + w_2 \times (B - b_j) + w_3 \times (1 - c_j), \quad (1)$$

где  $d_{i,j}$  – расстояние между задачей  $t_i$  и роботом  $r_j$ ,  $B$  – необходимое время работы от батареи задачи  $t_i$ ,  $b_j$  – оставшееся время работы от батареи робота  $r_j$ ,  $c_j$  – количество задач, назначенных роботу  $r_j$ , а  $w_1$ ,  $w_2$  и  $w_3$  – весовые коэффициенты, отражающие относительную важность расстояния, времени автономной работы и балансировки рабочей нагрузки. Значения  $d_{i,j}$  и  $B$  можно получить из онтологии, а значения  $b_j$  и  $c_j$  можно динамически обновлять по мере назначения задач роботам.

#### КОНЕЧНО-АВТОМАТНОЕ УПРАВЛЕНИЕ

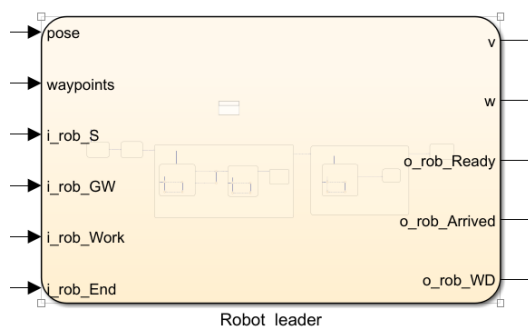
Модель управления состояниями робота-ведущего представлена на рисунке 3 (а) и (б). Эта модель включает в себя 6 входов и 5 выходов. Модель управления конечным автоматом ведомого робота представлена на рисунке 3 (в) и (г). Связь иллюстрированных автоматов между собой и с верхнеуровневым автоматом состояний всей системы, а также с объектами для публикации данных в ROS можно проследить на рисунке 4.

Структура и описание входных и выходных сигналов конечных автоматов роботов представлены в таблице 2.

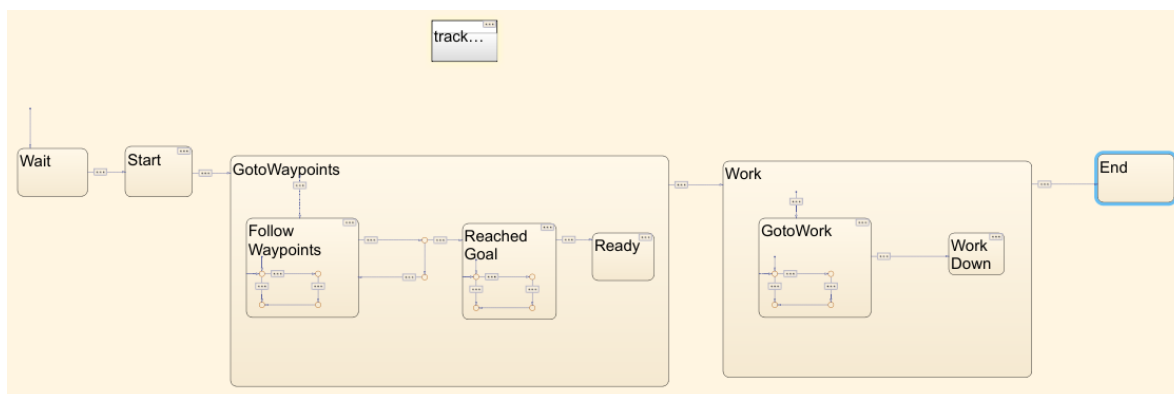
**Таблица 2.** Входы и выходы конечных автоматов ведущего и ведомого роботов

**Table 2.** Inputs and outputs of finite state machines of the leading and driven robots

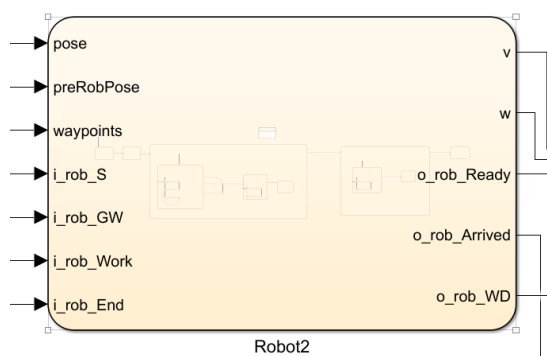
Показатель	Входы	Выходы
Ведущий робот	pose – текущее положение робота waypoints – начальная рабочая позиция робота i_rob_S – готовность робота к перемещению i_rob_GW – начинает ли робот движение к целевой точке i_rob_Work – начинает ли робот работу i_rob_End – завершил ли робот свою работу	v – желаемая линейная скорость робота w – желаемая угловая скорость робота o_rob_Ready – готов ли робот к перемещению o_rob_Arrived – достиг ли робот целевой точки o_rob_WD – завершилась ли работа с роботами
Ведомый робот	pose – текущее положение робота waypoints – начальная рабочая позиция робота i_rob_S – готовность робота к перемещению i_rob_GW – начинает ли робот движение к целевой точке i_rob_Work – начинает ли робот работу i_rob_End – завершил ли робот свою работу preRobPose – положение передового робота	v – желаемая линейная скорость робота w – желаемая угловая скорость робота o_rob_Ready – готов ли робот к перемещению o_rob_Arrived – достиг ли робот целевой точки o_rob_WD – завершилась ли работа с роботами



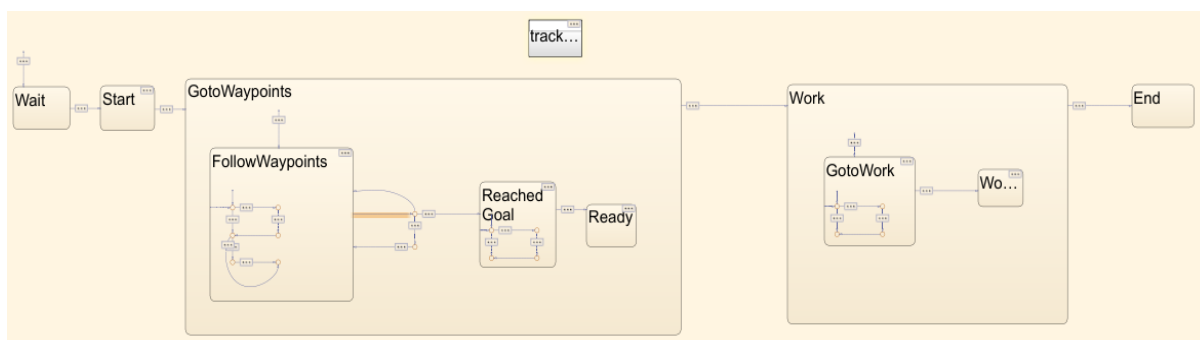
a)



б)



в)



г)

**Рис. 3.** Структура конечных автоматов роботов: а) – конечный автомат ведущего робота; б) – структура взаимодействия элементов ведомого робота; в) – конечный автомат ведомого робота; г) – структура взаимодействия элементов ведущего робота (Источник: составлено авторами)

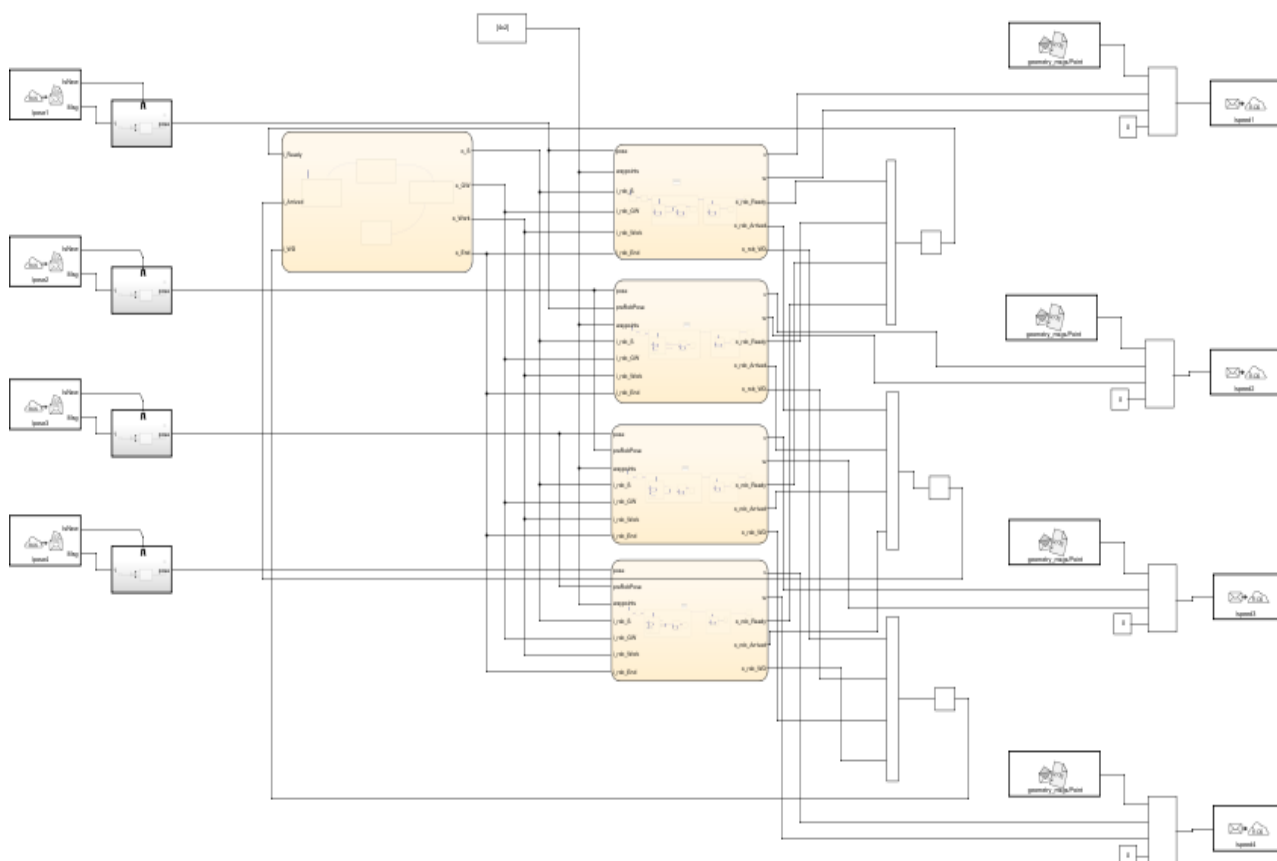
**Fig. 3.** Structure of finite state machines of robots: а) – finite state machine of the leading robot; б) – structure of interaction between elements of the driven robot; в) – finite state machine of the driven robot; г) – structure of interaction between the elements of the leading robot (Source: made by the authors)

Логика работы конечного автомата ведущего робота описывается следующим образом:

- Инициализация робота, проверка готовности.
- Робот движется к целевой точке, непрерывно оценивая угол в процессе работы и корректируя угловую скорость, если отклонение больше  $\pi/24$ .
- По достижении целевой точки робот продолжает корректировать установку, выставляя угловую скорость, и когда все установки верны, отправляет команду о достижении цели.
- Робот начинает работу и следует по траектории.
- Робот завершает свою работу и прекращает движение.
- Функция  $\text{trackGoal}(\text{goal})$  используется для вычисления угла и расстояния до целевой точки.

Логика работы конечного автомата ведомого робота описывается следующим образом:

- Инициализация робота, проверка его готовности.
- Робот движется к целевой точке, непрерывно оценивая угол в процессе работы и корректируя угловую скорость, если отклонение больше  $\pi/24$ .
- Оценка расстояния до робота, являющегося ведущим для текущего, и снижение скорости, если расстояние слишком мало.
- По достижении целевой точки робот продолжает корректировать установку, выставляя угловую скорость, и когда все установки верны, отправляет команду о достижении цели.
- Робот начинает работу и следует по траектории.
- Робот завершает свою работу и прекращает движение.



**Рис. 4.** Структура управляющего сетевого конечного автомата  
(Источник: составлено авторами)

**Fig. 4.** Structure of the control network finite state machine  
(Source: made by the authors)

## МОДЕЛИРОВАНИЕ ПРОЦЕССА

Группа роботов состоит из робота-ведущего и нескольких ведомых. Процесс работы развертывается следующим образом: сначала роботы последовательно прибывают в заранее назначенное место, после чего они одновременно приступают к выполнению задачи по уборке урожая.

В основе системы лежат следующие инструменты: MATLAB, ROS, CoppeliaSim [13, 14]. MATLAB (настроенный на функцию минимизации `ode15s`): логическое управление с использованием конечных автоматов для координации действий роботов. ROS: обеспечивает связь между MATLAB и CoppeliaSim для эффективного взаимодействия между различными компонентами системы. CoppeliaSim: реализует моделирование рабочего процесса робота, обеспечивая визуализацию и анализ выполнения задач. Для обмена сообщениями между перечисленными инструментами были использованы темы, приведенные на рисунке 5. Топики с корнем `pose` наполнялись данными из CoppeliaSim и считывались в MATLAB, а с корнем `speed` с точностью до наоборот, `simTime` необходима для корректной работы регуляторов на стороне MATLAB, выполняет задачу согласования времени между этими инструментами.

В создании распределенной робототехнической системы в рамках данной работы использован инструмент Mobile Robotics Simulation Toolbox в среде MATLAB. Такое сочетание инструментов обеспечивает эффективное функционирование и взаимодействие в рамках робототехнической системы.

```

/pose1
/pose2
/pose3
/pose4
/pose5
/rosout
/rosout_agg
/simTime
/speed1
/speed2
/speed3
/speed4
/speed5

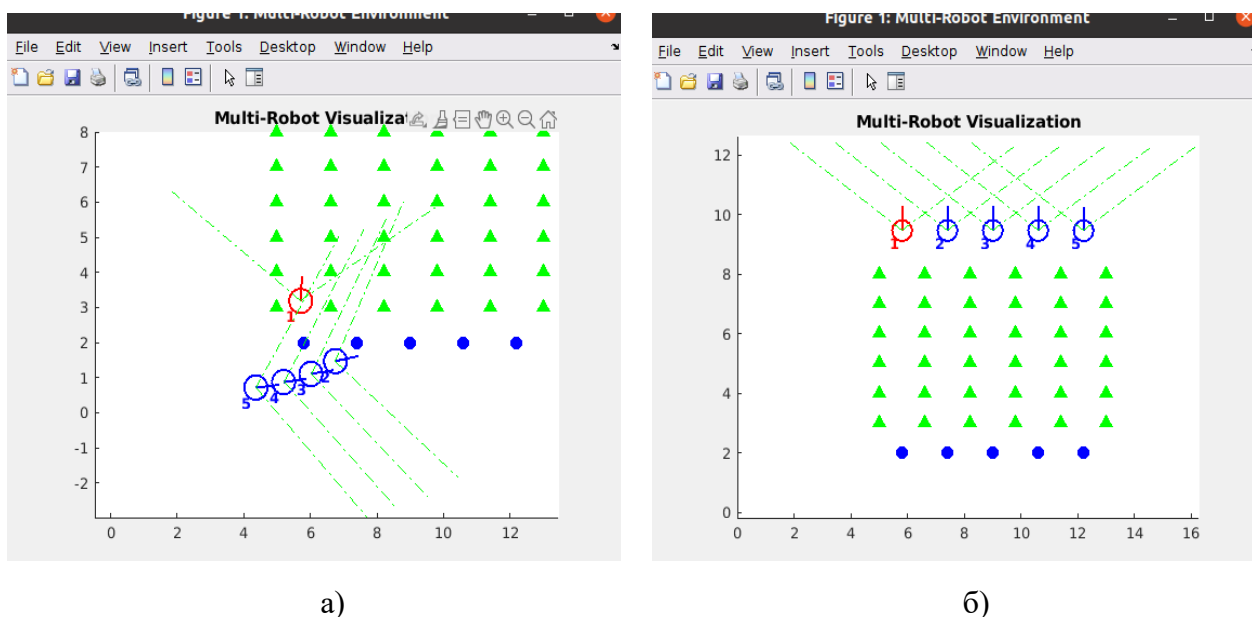
```

**Рис. 5.** Активные топики (Источник: составлено авторами)

**Fig. 5.** Active topics (Source: made by the authors)

В процессе управления на вход робота поступают желаемые значения линейной скорости и угла поворота, а на выходе формируются скорости колес робота. Робот изображен в виде прототипа, где красным цветом выделен робот-ведущий, а синим – робот-ведомый. Весь рабочий процесс был успешно смоделирован в среде MATLAB. На рисунке 6 на карте представлено 5 маршрутов между насаждениями. Начальная точка обозначена синей меткой. Робот начинает движение по рабочей траектории после достижения заданной начальной точки. Состояние робота в начале распределения между рядами деревьев проиллюстрировано на рисунке 6 (а), а конечное состояние – на рисунке 6 (б).

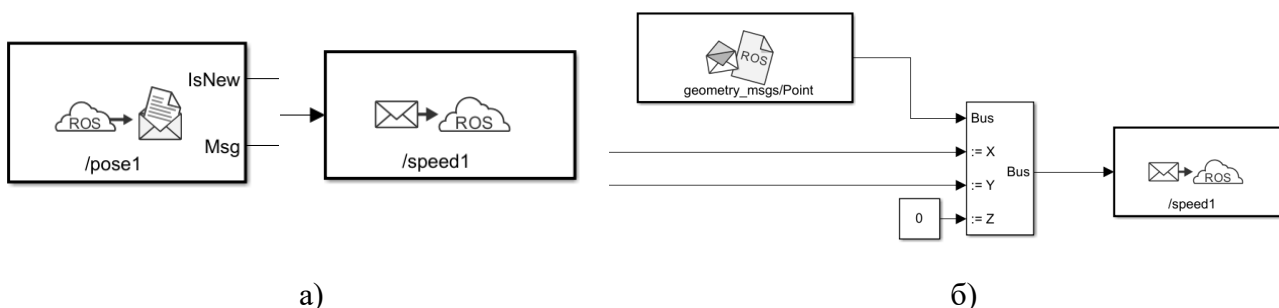




**Рис. 6.** Моделирование группы роботов при выполнении назначенных задач:  
 а) – начало процесса выполнения задачи; б) – конец прохода ряда  
 (Источник: составлено авторами)

**Fig. 6.** Simulation of a group of robots performing assigned tasks:  
 a) – the beginning of the task execution process; b) – end of row passage  
 (Source: made by the authors)

Для публикации и подписки на сообщения в Simulink используется ROS Toolbox, модуль темы представлен на рисунке 7.



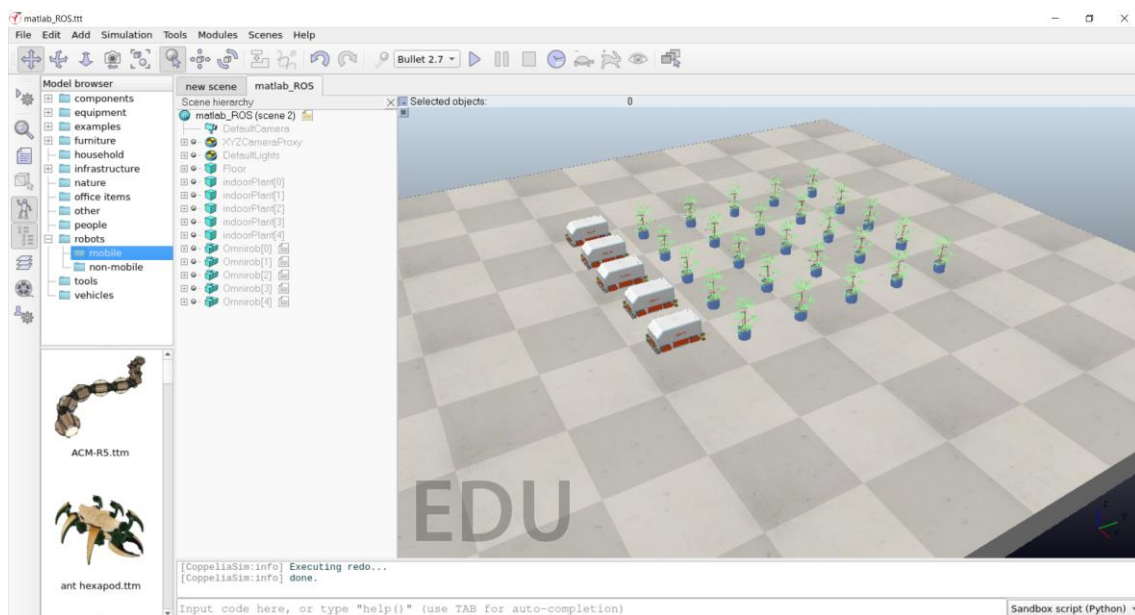
**Рис. 7.** а) – блоки модуля ROS для подписки и публикации;  
 б) – блок-схема MATLAB для публикации  
 (Источник: составлено авторами)

**Fig. 7.** a) – ROS module blocks for subscription and publication;  
 b) – MATLAB block diagram for publication  
 (Source: made by the authors)

Модель робота и карты в CoppeliaSim представлены на рисунке 8. В данной модели сельскохозяйственные культуры заменены на стеновые блоки с маркерами, указывающими направление движения вперед.

Используя предложенную оценку стоимости назначения задач (1), составим матрицу затрат и применим венгерский алгоритм, чтобы найти оптимальное распределение задач для системы с несколькими роботами, состоящей из ведущего и ведомых роботов.

Для этого сначала происходит загрузка изображения карты и создание бинарной карты занятости. Количество задач и роботов определено переменными numTasks и numRobots.



**Рис. 8.** Моделирование роботов и карт в CoppeliaSim  
(Источник: составлено авторами)

**Fig. 8.** Simulation of robots and maps in CoppeliaSim  
(Source: made by the authors)

Далее задача разбивается на 4 подзадачи разной стоимости. Стоимость задачи определяется суммой всех подзадач, в нее входящих. Стоимость подзадач определяется по расходу заряда аккумулятора у робота для выполнения подзадачи. Транспортные перемещения до начала ряда требуют меньше всего заряда. Движение вдоль ряда требует большего расхода заряда, так как робот не только движется, но и собирает плоды, следовательно, стоимость подзадачи больше.

Затем используется алгоритм  $A^*$  для планирования путей от каждого робота к каждой задаче. Результаты сохраняются в матрицу pathArray, а стоимость каждой задачи сохраняется в матрицу costMatrix.

После этого происходит распределение задач между роботами с использованием венгерского алгоритма и вывод результатов.

В таблицах 3–5 представлены матрицы стоимости, где номера столбцов соответствуют номерам роботов, а номера строк – номерам задач. На зеленом фоне оптимальное распределение задач по роботам.

**Таблица 3.** Матрица стоимости и оптимальное распределение для 4 роботов и 4 задач

**Table 3.** Cost matrix and optimal allocation for 4 robots and 4 tasks

№ зад.\№ роб.	1	2	3	4
1	268	73	153	280
2	95	252	110	105
3	219	139	87	229
4	315	92	216	327

**Таблица 4.** Матрица стоимости и оптимальное распределение для 5 роботов и 5 задач**Table 4.** Cost matrix and optimal allocation for 5 robots and 5 tasks

№ зад.\№ роб.	1	2	3	4	5
1	27	233	272	182	371
2	132	262	301	211	252
3	270	328	289	331	184
4	200	36	75	125	181
5	247	200	151	286	141

**Таблица 5.** Матрица стоимости и оптимальное распределение для 6 роботов и 6 задач**Table 5.** Cost matrix and optimal allocation for 6 robots and 6 tasks

№ зад.\№ роб.	1	2	3	4	5	6
1	230	146	183	226	257	157
2	235	145	188	225	256	156
3	130	261	148	107	73	178
4	322	38	275	156	187	61
5	15	345	34	195	209	290
6	259	158	277	98	58	75

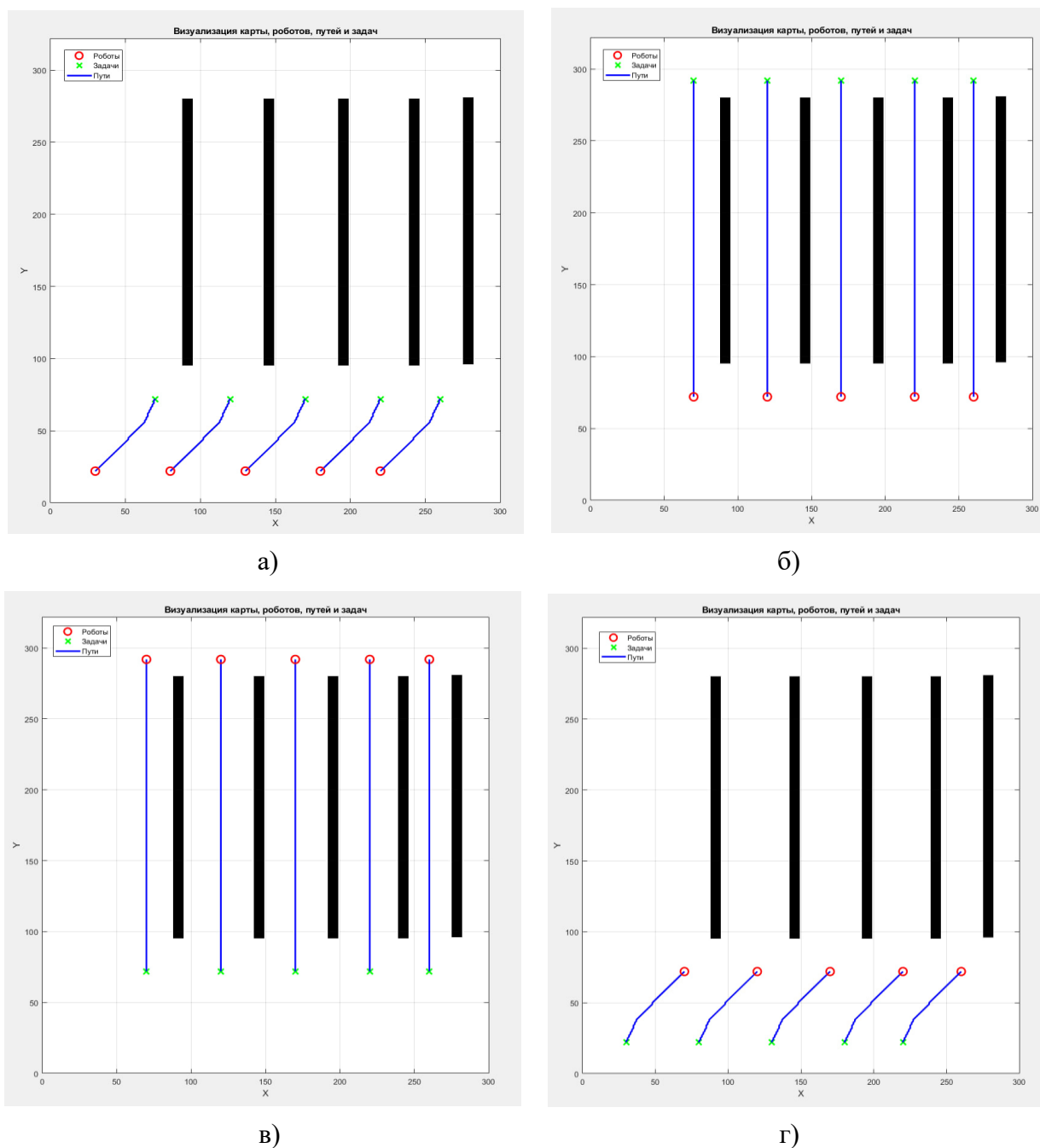
Реализация венгерского алгоритма, используемого для оптимального распределения задач между роботами в соответствии с матрицей стоимости costMatrix, следующая:

1. Инициализация:
  - 1.1. Получить размеры матрицы стоимости (numRows и numCols).
  - 1.2. Вычислить минимальные значения по строкам и столбцам и вычесть их из соответствующих строк и столбцов матрицы стоимости.
  - 1.3. Инициализировать переменные и массивы для отслеживания распределения задач: coveredRows, coveredCols и assignment.
2. Поиск и покрытие полей:
  - 2.1. Поочередно находить ноли в матрице стоимости, которые еще не были покрыты и не использовались для распределения задач.
  - 2.2. Если найден ноль в текущих строке и столбце, обновить массивы coveredRows и coveredCols и прервать вложенный цикл 3.
3. Итерации для покрытия всех строк:
  - 3.1. Запустить цикл, который будет повторяться, пока есть непокрытые строки.
  - 3.2. Если невозможно найти ноль в текущей строке, уменьшить все непокрытые значения матрицы на минимальное непокрытое значение.
4. Возврат оптимального распределения:
  - 4.1. По завершении цикла вернуть оптимальное распределение задач в виде матрицы assignment.

На рисунке 9 представлен результат оптимального распределения задач между роботами и траекторий движения для их достижения. Критерием оптимизации в данном случае выступает расстояние до точки выполнения задачи.

Сценарий робота был отредактирован в CoppeliaSim для преобразования полученных из топики ROS скоростей в скорости всех четырех колес в соответствии с кинематической моделью платформы с меканум колесами.

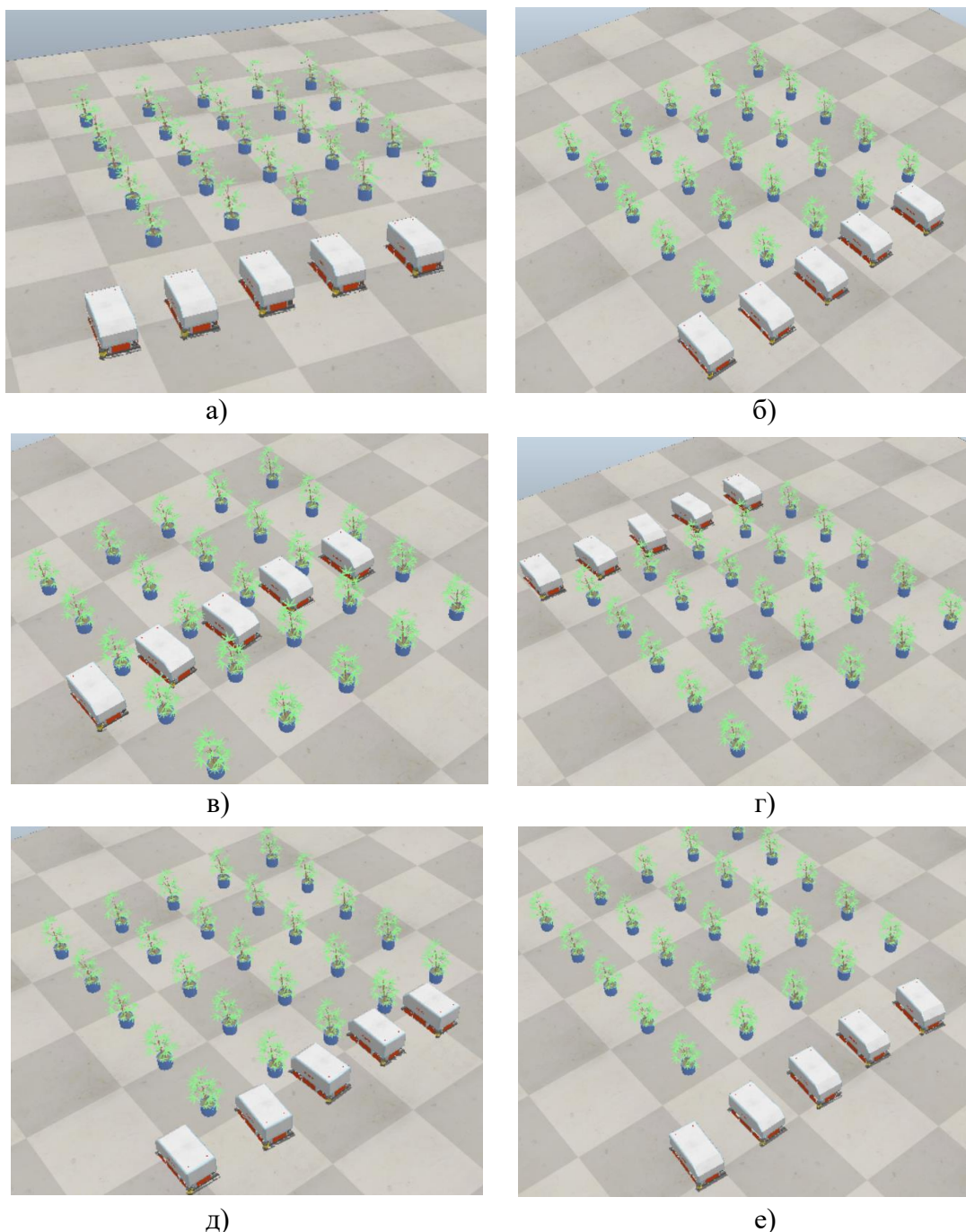
Движение роботов к рядам происходит в соответствии с вышеуказанным распределением, порядок роботов в строю регулируется результатом работы венгерского алгоритма, выполненного для начальных позиций роботов.



**Рис. 9.** а) – движение на исходные позиции для работы; б) – движение вдоль рядов; в) – возврат вдоль рядов на исходные позиции; г) – возвращение домой  
(Источник: составлено авторами)

**Fig. 9.** a) – movement to the starting positions for work; b) – movement along the rows; c) – return along the rows to the starting positions; d) – return to home  
(Source: made by the authors)

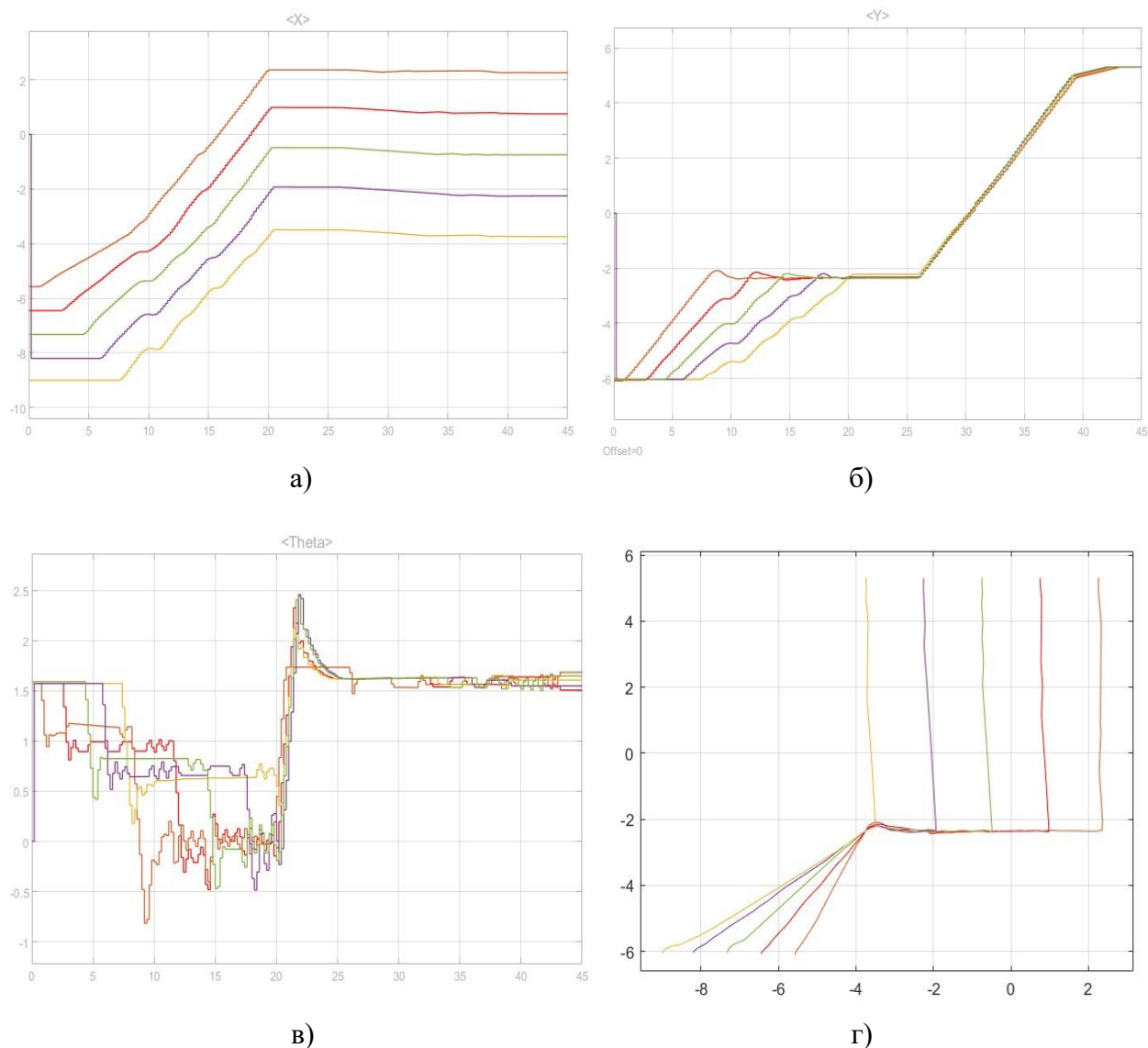
На рисунке 10 отображен процесс моделирования в CoppeliaSim движения мобильных роботов по траекториям, сформированным из задачи распределения.



**Рис. 10.** а) – движение на исходные позиции для работы; б) – начало движения вдоль рядов; в) – движение вдоль рядов; г) – возврат вдоль рядов из конечной точки на исходные позиции; д) – достижение исходной позиции; е) – в стартовой позиции  
(Источник: составлено авторами)

**Fig. 10.** a) – movement to the starting positions for work; б) – movement began along the rows; в) – movement along the rows; г) – return along the rows from the end point to the starting positions; д) – reaching the starting position; е) – in the starting position  
(Source: made by the authors)

На рисунке 11 приведена динамика изменения координат каждого робота на этапе подхода к исходной позиции и движения вдоль ряда. Ступенчатость значений обоснована имеющей значение частотой отсылки сообщений из симулятора в ROS.



**Рис. 11.** а) – изменение  $x$  координаты роботов; б) – изменение  $y$  координаты роботов; в) – изменение угла поворота роботов; г) – зависимости  $x$  от  $y$  (Источник: составлено авторами)

**Fig. 11.** a) – change in the  $x$  coordinates of the robots; б) – change in the coordinates of the robots; в) – change in the angle of rotation of the robots; г) – dependence of  $x$  on  $y$  (Source: made by the authors)

#### ЗАКЛЮЧЕНИЕ

Разработана методика моделирования назначения задач для систем, управляемых конечными автоматами, с оптимизацией по критерию эффективности. Разработана модель задачи роботизированного сбора урожая группой роботов в среде Simulink с применением Stateflow. Для обеспечения совместного моделирования в реальном времени между MATLAB и CoppeliaSim был использован ROS для эффективной организации коммуникации между системами. Этот подход позволяет успешно интегрировать и координировать действия роботов в симуляционной среде, обеспечивая более точное и надежное моделирование задачи сбора урожая.

## REFERENCES

1. Нагоев З. В., Шуганов В. М., Бжихатлов К. Ч. и др. Перспективы повышения производительности и эффективности сельскохозяйственного производства с применением интеллектуальной интегрированной среды // Известия Кабардино-Балкарского научного центра РАН. 2021. № 6(104). С. 155–165.

Nagoev Z.V., Shuganov V.M., Bzhikhatlov K.Ch. et al. Prospects for increasing the productivity and efficiency of agricultural production using an intelligent integrated environment. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2021. No. 6(104). Pp. 155–165. (In Russian)

2. Зенкевич С. Л., Чжу Хуа, Хо Цзяньвень, Движение группы мобильных роботов в строю типа «конвой» – теория, моделирование и эксперимент // IV Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта», 5–6 октября 2017 г., Казань: труды семинара, под ред. Е. А. Магида, В. Е. Павловского, К. С. Яковлева. Казань: ЦИТ, 2017. С. 136–147.

Zenkevich S.L., Zhu Hua, Ho Jianwen. *Dvizhenie gruppy mobil'nyh robotov v stroju tipa «konvoj» – teorija, modelirovanie i eksperiment* [Movement of a group of mobile robots in a convoy-type formation – theory, modeling and experiment]. *IV Vserossiyskiy nauchno-prakticheskiy seminar «Bespilotnyye transportnyye sredstva s elementami iskusstvennogo intellekta», 5–6.10.2017, Kazan: trudy seminar, eds. E.A. Magid, V.E. Pavlovsky, K.S. Yakovlev. Kazan: Tsentr innovatsionnykh tekhnologiy, 2017. Pp. 136–147. (In Russian)*

3. Shereuzhev M., Mostakov N., Vorotnikov S. Development of the Elements of a Control System for a Mobile Agricultural Robot Operating in a Group. *Modeling in Engineering*. Moscow, 2021. С. 11–18.

4. Нагоев З. В., Нагоева О. В. Извлечение знаний из многомодальных потоков неструктурированных данных на основе самоорганизации мультиагентной когнитивной архитектуры мобильного робота // Известия Кабардино-Балкарского научного центра РАН. 2015. № 6(68). Т. 2. С. 145–152.

Nagoev Z.V., Nagoeva O.V. Extracting knowledge from multimodal streams of unstructured data based on the self-organization of a multi-agent cognitive architecture of a mobile robot. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2015. No. 6–2(68). Pp. 145–152. (In Russian)

5. Nazarova A.V., Huo J., Zenkevich S.L. Dynamic switching of multi-agent formation in unknown obstacle environment. *Studies in systems, decision and control*. 2020. No. 261. Pp. 73–87.

6. Носков В. П., Рубцов И. В. Опыт решения задачи автономного управления движением мобильных роботов // Мехатроника, автоматизация, управление. 2005. № 12. С. 21–24.

Noskov V.P., Rubtsov I.V. Experience in solving the problem of autonomous control of the movement of mobile robots. *Mekhatronika, avtomatizatsiya, upravleniye* [Mechatronics, automation, control]. 2005. No. 12. Pp. 21–24. (In Russian)

7. Hohimer C. Design and Field Evaluation of a Robotic Apple Harvesting System with a 3D-Printed Soft-Robotic End-Effector. *Transactions of the ASABE*. 2019. No. 62. Pp. 405–414. DOI: 10.13031/trans.12986.

8. Yang S.H., Lai W.H. Multi-robot task allocation with fuzzy constraint satisfaction. *International Journal of Advanced Robotic Systems*. 2012. No. 9(3). Pp. 107–116.

9. Julian B.J., Gerkey B.P. Scalable multirobot task allocation for complex missions. *Proceedings of the International Conference on Intelligent Robots and Systems*. 2007. Pp. 2730–2736.

10. Lacroix S., Nourbakhsh I.R., Tomatis N. Multi-robot task allocation in uncertain environments using negotiation. *Robotics and Autonomous Systems*. 2001. No. 37(2–3). Pp. 163–179.
11. Shi Z., Yu H., Zhang Q., Cheng Y. A hybrid algorithm for task allocation in multi-robot systems based on improved Hungarian algorithm and particle swarm optimization. *International Journal of Advanced Robotic Systems*. 2015. No. 12(8). Pp. 1–13.
12. Martinelli S., Giordani M., Lujak F. A Distributed Algorithm for the Multi-Robot Task Allocation Problem. IEA/AIE 2010: *Trends in Applied Intelligent Systems*. Cordoba, Spain: Springer, 2010.
13. Галин Р.Р. Виртуальный полигон для эффективного взаимодействия роботов в многоагентной робототехнической системе // Известия Кабардино-Балкарского научного центра РАН. 2018. № 6–2(86). С. 108–113.
- Galín R.R. Virtual training ground for effective interaction of robots in a multi-agent robotic system. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2018. No. 6–2(86). Pp. 108–113. (In Russian)
14. Shamshiri R.R., Hameed I.A., Karkee M., Weltzien C. Robotic Harvesting of Fruiting Vegetables: A Simulation Approach in V-REP, ROS and MATLAB. *Automation in Agriculture – Securing Food Supplies for Future Generations*. 2018. URL: <http://dx.doi.org/10.5772/intechopen.73861> (access date: 15.08.2019).

### Информация об авторах

**Шереужев Мадин Артурович**, ст. преподаватель кафедры «Робототехнические системы и мехатроника», Московский государственный технический университет им. Н. Э. Баумана;

105005, Россия, Москва, 2-я Бауманская улица, 5, корп. 1;  
shereuzhev@bmstu.ru, ORCID: <https://orcid.org/0000-0003-2352-992X>

**Арабаджиев Денис Игоревич**, студент-магистр кафедры «Робототехнические системы и мехатроника», Московский государственный технический университет им. Н. Э. Баумана;

105005, Россия, Москва, 2-я Бауманская улица, 5, корп. 1;  
arabadzhievdi@student.bmstu.ru

**Девяткин Федор Владимирович**, студент-магистр кафедры «Робототехнические системы и мехатроника», Московский государственный технический университет им. Н. Э. Баумана;

105005, Россия, Москва, 2-я Бауманская улица, 5, корп. 1;  
feodor-dev@ya.ru

**Шереужев Марат Артурович**, аспирант кафедры «Агрономия», Кабардино-Балкарский государственный аграрный университет им. В. М. Кокова;

360030, Россия, г. Нальчик, пр-т Ленина, 1в;

стажер-исследователь лаборатории «Интеллектуальные среды обитания», Институт информатики и проблем регионального управления – филиал Кабардино-Балкарского научного центра РАН;

360000, Россия, Нальчик, ул. И. Арманд, 37-а;

marat.shereuzhev07@mail.ru, ORCID: <https://orcid.org/0000-0001-7368-4691>

### Information about the authors

**Shereuzhev Madin Arturovich**, Senior Teacher of the Department of “Robotic systems and mechatronics”, Moscow State Technical University named after N.E. Bauman;

105005, Russia, Moscow, 2nd Baumanskaya street 5, building 1;  
shereuzhev@bmstu.ru, ORCID: <https://orcid.org/0000-0003-2352-992X>



**Arabadzhiev Denis Igorevich**, Graduate Student of “Robotics and mechatronics” Department, Moscow State Technical University named after N.E. Bauman; 105005, Russia, Moscow, 2nd Baumanskaya street 5, building 1; arabadzhievdi@student.bmstu.ru

**Devyatkin Fedor Vladimirovich**, Graduate Student of “Robotics and mechatronics” Department, Moscow State Technical University named after N.E. Bauman; 105005, Russia, Moscow, 2nd Baumanskaya street 5, building 1; feodor-dev@ya.ru

**Shereuzhev Marat Arturovich**, Postgraduate Student of “Agronomy” Department, Kabardino-Balkarian State Agrarian University named after V.M. Kokov; 360030, Russia, Nalchik, 1v Lenin avenue;

Trainee researcher of the Laboratory “Intellectual Habitats” of the Institute of Computer Science and Problems of Regional Management – branch of Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences;

360000, Russia, Nalchik, 37-a I. Armand street;  
marat.shereuzhev07@mail.ru, ORCID: <https://orcid.org/0000-0001-7368-4691>