



## УПРАВЛЕНИЕ МОБИЛЬНЫМ РОБОТОМ В УСЛОВИЯХ НЕПОЛНОЙ ИНФОРМАЦИИ: ИНТЕГРАЦИЯ МЕТОДОВ ПРОГРАММНОГО УПРАВЛЕНИЯ И АДАПТИВНОГО ВЫБОРА ВАРИАНТОВ

УДК 004.81:004.383.8

### ДОДОНОВ Вадим Александрович

научный сотрудник, Институт проблем регистрации информации НАНУ, г.Киев,

**Научные интересы:** базы данных и знаний, агентные технологии, робототехнические системы, .

**e-mail:** v.a.dodonov14@gmail.com

#### ВВЕДЕНИЕ

В современном мире широко используются информационные технологии в различных приложениях, связанных с мониторингом ситуаций [1]: экологических, экономических, техногенных, производственных. Особенно актуальны системы, в которых благодаря мониторингу достигается безопасность объектов [2]. Рассматриваются различные стратегии построения систем мониторинга сложных объектов [3]: от полностью централизованных до рассредоточенных. Комбинированный вариант, как наиболее экономичный с точки зрения затрат, заключается в том, что централизованная компонента системы мониторинга локализует состояние объекта мониторинга на достаточно общем уровне, а уточнение ситуации выполняют локальные подсистемы. Например, пожароопасная ситуация в помещении централизовано контролируется с помощью одного из датчиков (дымового, теплового, пламени и другие). В настоящее время становятся доступными системы, которые могут обнаруживать пожар более чем по одному признаку с целью лучшего распознавания между состоянием пожара и состоянием, когда его признаки отсутствуют. Такие системы могут иметь несколько чувствительных элементов или могут сочетать показания от нескольких разных пожарных извещателей [4]. Все это повышает надежность идентификации опасной ситуации, но ложные сигналы тревоги всё же могут вызываться другими причинами. Большой объем информации, поступающей от места лока-

лизации опасной ситуации, может значительно уменьшить количество ложных сигналов, возникающих из-за влияния окружающей среды [4]. Это справедливо и для охранных, и для многих других систем специального назначения. Так, для охранных систем централизованно регистрируется место несанкционированного проникновения (участок, на котором сработал инфракрасный датчик движения), а тип нарушителя, его идентификация требуют дополнительной информации по месту локализации. Эту информацию можно получить благодаря мобильным устройствам, которые оснащены развитыми сенсорными системами и способны переместиться к участку, где было локализовано нарушение.

Современные технологии делают доступными как по цене, так и по размерам разнообразные сенсорные и микропроцессорные компоненты, средства обмена информацией, такие как Wi-Fi. Все это способствует использованию принципа рассредоточенного мониторинга опасных ситуаций с применением информационных технологий «умного» дома, или «умных машин» [5].

Система для крупномасштабных объектов, организованная по принципу рассредоточенного мониторинга, включает локальные компоненты, которые реализованы на устройствах (роботах, дронах), перемещаемых в пространстве объекта мониторинга. Такой робот оснащен несколькими сенсорными системами, микропроцессором и системой беспроводной, например, Wi-Fi связи.

Функции, которые выполняет мобильный робот, заключаются в следующем:

1. Перемещение в пространстве объекта мониторинга с парковой позиции к указанной точке по маршруту, который загружен в память системы управления роботом. Управление роботом осуществляется в условиях помех [6].

2. Наблюдение за средой в точке, куда он прибыл. Наблюдение выполняется с помощью сенсорных систем, которыми робот оснащен. Процесс наблюдения является длительным во времени, что необходимо для вычисления динамических свойств среды, и связан с поисковыми маневрами в пространстве. Этот процесс не является заранее запрограммирован, а есть случайным поисковым.

3. Передача модели ситуации, которая построена на сенсорных данных и представляет нечеткое описание свойств локализованного окружения, к центральному компьютеру по беспроводной связи.

Функции, выполняемые на центральном компьютере, заключаются в следующем:

1. При обнаружении центральным компонентом системы мониторинга любых нарушений локализуется участок, где это нарушение произошло, и принимается решение о способе получения дополнительной информации о состоянии среды на участке. Формируется задача роботу (маршрут достижения участка, перечень свойств среды подлежащих контролю и способ получения этих свойств).

2. Прием по беспроводной связи от робота модели ситуации на участке.

3. Построение модели ситуации на общем уровне в виде абстрактного описания ситуации на основании сенсорных данных, поступивших от робота, и данных, которые были актуальны на предыдущее время. Эта задача реализуется на базе моделей и методов граничных вычислений [7].

В статье рассматривается первая задача, решаемая мобильным роботом – перемещение в пространстве объекта мониторинга по заданному маршруту. Пространство, в котором перемещается робот – помещения внутри здания. С целью снижения затрат на локальную компоненту системы мониторинга предлагается управление роботом реализовать на основании ультразвукового сенсора определения расстояния до

препятствия. Известны [8] проблемы, связанные с использованием этих сенсорных данных в управлении. Основная, среди них, состоит в следующем.

Движение по заранее заданной траектории возможно с использованием метода программного управления с обратными связями [6]. Если в качестве обратной связи используется информация от ультразвукового датчика, то на ориентиры в помещении, в котором перемещается робот, накладываются определённые требования [8]. Например, сплошные стены, как в соревнованиях роботов на прохождение лабиринта. Прерывистости в стенах, например, дверные проёмы, или препятствия в виде столов, стульев, отдельно стоящих колонн и так далее приводят к сбоям при отработке маршрута движения. Системы управления роботом, функционирующим в среде с препятствиями, разрабатываются, как правило, на основании эвристических алгоритмов [9]. Это приводит к тому, что масштабирование задач осуществляется путём использования библиотек и параметризованных программ управления обходом конкретных препятствий. Но, при этом, по-прежнему, остаётся нерешенной задача управления перемещением вдоль заранее заданного маршрута при нарушениях в маркировании маршрута.

Указанная проблема может быть преодолена с использованием модифицированного метода программного управления с обратными связями, дополненного системой *адаптивного выбора вариантов*.

## ПОСТАНОВКА ЗАДАЧИ

В работе описывается новый подход к управлению мобильными роботами, который представляет интеграцию метода программного управления с обратными связями [6] с методом адаптивного выбора вариантов [10] с целью расширения возможностей управления автономными роботами в неупорядоченных средах. Снятие ряда ограничений, накладываемых на среду функционирования автономного робота, связано со снижением затрат на упорядочение среды и приводит к расширению функциональности робота и возможности применения его в реальной обстановке.

Задача ставится следующим образом. Пусть роботу нужно проследовать по одному из множества возможных маршрутов, показанных на рис. 1. И, пусть, это будет маршрут из точки А в точку Е. При движении по



В предположении постоянной скорости вычисляются все требуемые интервалы времени: перемещения от точки  $A$  до точки  $B$ , что будем обозначать  $t_{A \rightarrow B}$ , от точки  $B$  до точки  $C$  ( $t_{B \rightarrow C}$ ) и другие  $t_{C \rightarrow D}$ ,  $t_{D \rightarrow E}$ . Аналогично экспериментальным путём определяются временные интервалы для поворотов направо и налево  $t_n, t_l$ . Перед началом выполнения управляющей программы робот должен находиться в определённом исходном состоянии: в точке  $A$  (рис. 1). Управляющая программа для метода прямых связей (классическое программное управление) приведена ниже.

- (1) *вперёд;*
- (2) *задержка  $t_{A \rightarrow B}$ ;*
- (3) *выкл\_вперёд;*
- (4) *влево;*
- (5) *задержка  $t_l$  ;*
- (6) *выкл\_влево;*
- (7) *вперёд;*
- (8) *задержка  $t_{B \rightarrow C}$ ;*
- (9) *выкл\_вперёд;*
- (10) *вправо;*
- (11) *задержка  $t_n$  ;*
- (12) *выкл\_вправо;*
- (13) *вперёд;*
- (14) *задержка  $t_{C \rightarrow D}$ ;*
- (15) *выкл\_вперёд;*
- (16) *влево;*
- (17) *задержка  $t_l$  ;*
- (18) *выкл\_влево;*
- (19) *вперёд;*
- (20) *задержка  $t_{D \rightarrow E}$ ;*
- (21) *выкл\_вперёд;*
- (22) *стоп.*

Из анализа программы управления по прямым связям видно, насколько метод требователен к предварительному упорядочиванию среды: обязательные начальные условия (строго определённое исходное состояние робота) и жесткие требования к скорости перемещения и мгновенные режимы разгона/торможения робота. Если первое требование может быть выполнено при определённой организации, то второе практически никогда не выполняется. В методе программного управления с обратными связями для идентификации события выхода робота на определённые

позиции используют информацию от датчиков, а не ориентируются на интервалы времени. Как отмечалось в постановке задачи, в управлении используется информация от УЗ датчика. Означим эти данные в виде  $L$  – расстояние до препятствия. Так как датчик установлен на поворотной платформе, то ещё доступны данные о направлении, по которому поступает информация о препятствии. Это данные  $\varphi_l, \varphi_r, \varphi_f$  – слева (left), справа (right) и прямо (forward) от робота. В управляющей программе с обратными связями в формат команды, например, *вперёд* вводится дополнительный параметр – условие завершения выполнения команды. Это условие задаётся конкретным значением сенсорных данных, например,  $L = 20$ . Интерпретация этой команды сводится к тому, что в автономном режиме САУ нижнего уровня, используя информацию от сенсора, выдаёт команду *выкл\_вперёд* когда появится значение  $L = 20$  и информирует верхний уровень о завершении выполнения команды. Команда *вперёд*  $L = 20$  в методе управления с обратными связями эквивалентна следующей последовательности команд метода управления с прямыми связями

- (1) *вперёд;*
- (2) *задержка пока  $L = 20$  ;*
- (3) *выкл\_вперёд*

Вторая команда в (2) задаёт задержку не на конкретный интервал времени  $t_{B \rightarrow C}$ , а до тех пор, пока не будет достигнута целевая позиция по показаниям сенсора  $L = 20$ .

Ниже приведена управляющая программа, аналогичная по функциональности (1) для системы управления с обратными связями.

- (1) *вперёд  $L_f = s_I$  ;*
- (2) *влево  $L_f = \infty$  и  $L_r = s_I$  и  $L_l = s_A$*
- (3) *вперёд  $L_f = s_G$  ;*
- (4) *вправо  $L_f = \infty$  и  $L_r = s_G$  и  $L_l = \infty$  ;*
- (5) *вперёд  $L_f = s_H$  ;*
- (6) *влево  $L_f = s_E$  и  $L_r = \infty$  и  $L_l = s_H$  ;*
- (7) *вперёд  $L_f = s_0$  ;*
- (8) *стоп*

В программе (3) в командах управления указаны условия достижения целевых позиций, например, в  $L_f = \infty$  и  $L_r = s_I$  и  $L_l = s_A$  записано условие: слева нет препятствия, слева на расстоянии  $s_A$  (от точки А до стены помещения П №1 на рис. 1) и справа – на расстоянии  $s_B$  находится стена помещения П №3.

Под управлением такой программы траектория робота уже не будет чувствительна к вариациям скорости и ускорения его движения, однако появление препятствий на участках маршрута между поворотами или во время разворотов по-прежнему, приведёт к нарушениям требуемой траектории передвижения. Для преодоления этой проблемы в статье предлагается в управляющую программу включить альтернативные условия, которые проверяются контроллером нижнего уровня на основе методов адаптивного выбора вариантов. Ниже приводится модифицированная (3) УП с альтернативными условиями.

- (1) *вперёд* ( $L_f = s_I$  или  $t > t_{A \rightarrow B}$ );
- (2) *влево* ( $(L_f = \infty$  и  $L_r = s_I$  и  $L_l = s_A)$  или  $t > t_n$ );
- (3) *вперёд* ( $L_f = s_G$  или  $t > t_{B \rightarrow C}$ );
- (4) *вправо* ( $(L_f = \infty$  и  $L_r = s_G$  и  $L_l = \infty)$  или  $t > t_n$ );
- (5) *вперёд* ( $L_f = s_H$  или  $t > t_{C \rightarrow D}$ );
- (6) *влево* ( $(L_f = s_E$  и  $L_r = \infty$  и  $L_l = s_H$  или  $t > t_n$ );
- (7) *вперёд* ( $L_f = s_0$  или  $t > t_{D \rightarrow E}$ );
- (8) *стоп*.

В (4) в качестве альтернативного условия использовано плановое время перемещения между двумя позициями или время поворота. С такой альтернативой УП представляет гибрид программного управления по прямым и обратным связям, то есть (1) и (3). В качестве альтернативы может выступить расстояние, если робот оснащен спидометром или имеется GPS связь и другие. Альтернатива обрабатывается контроллером нижнего уровня – модулем адаптивного выбора вариантов принимается решение либо о достижении целевой позиции и выдается одна из команд *выкл\_вправо*, *выкл\_влево*, *выкл\_вперёд*, *выкл\_назад*, либо об устранении препятствие и продолжении отработки той же команды УП.

### МЕТОД АДАПТИВНОГО ВЫБОРА ВАРИАНТОВ

Достижение заданной позиции в условиях неопределённости возможно на основе применения адаптивного подхода [10,11], смысл которого состоит в использовании текущей информации, получаемой в результате отдельных действий выбора, что позволяет компенсировать недостаток априорной информации и реализовать оптимальную на классе систем стратегию управления.

Рассмотрим общую постановку задачи адаптивного выбора вариантов, представленную на рис. 2.

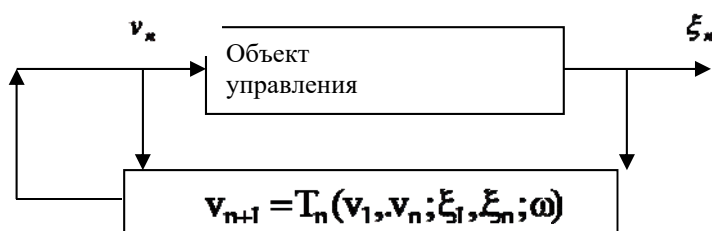
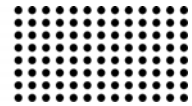
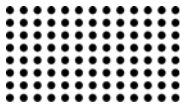


Рис. 2. Схема адаптивного выбора вариантов

В результате произведенного выбора управления потери системы  $\xi_n$  представляют собой случайную величину – функцию элементарного исхода  $\omega$ , и зависят от  $v_n$  и, возможно, состояний системы. Реализуемая при этом последовательность вариантов  $\{v_n\}$  должна быть такой, чтобы достигалась заданная цель,

формулируемая в терминах предельных значений текущих средних потерь.

Наличие априорной неопределенности, состоящей в отсутствии точной информации о потерях системы и ее характеристиках, приводит к тому, что формирование последовательности вариантов  $\{v_n\}$ , обеспечивающей достижение целевого условия решаемой задачи, следу-



ет осуществлять в соответствии с адаптивным подходом. При этом выбор очередного варианта  $v_{n+1}$  производится на основе полученной к данному моменту времени последовательности потерь  $\xi_1, \xi_2, \dots, \xi_n$ , соответствующей реализованной последовательности

вариантов  $v_1, v_2, \dots, v_n$ . Это значит, что  $v_{n+1}$  является функцией от  $v_1, v_2, \dots, v_n, \xi_1, \xi_2, \dots, \xi_n$  и, возможно, от момента времени  $n$  и элементарного исхода  $\omega$ . Таким образом:

$$v_{n+1} = T_n(v_1, v_2, \dots, v_n; \xi_1, \xi_2, \dots, \xi_n; \omega), n = 1, 2, \dots \quad (5)$$

где  $\xi_n$  в зависимости от задачи – либо скаляр, либо вектор.

Функцию  $T_n$  называют правилом выбора варианта  $v_{n+1}$ . Эта функция может быть как детерминированной, так и случайной (рандомизированной). Последовательность  $\{T_n\}$  правил выбора определяет стратегию выбора вариантов или стратегию управления.

стратегия всегда принадлежит классу детерминированных стратегий:

$$v_{n+1} = T_n(\omega), n = 1, 2, \dots \quad (6)$$

Детерминированные стратегии выбора составляют предмет изучения теории поведения детерминированных автоматов в случайных средах [12]. Эти стратегии допускают возможность простой реализации с помощью детерминированных конечных автоматов. Они в основном ориентированы на задачи с бинарными потерями. В условиях полной информации оптимальная

Наличие априорной неопределенности приводит к необходимости использовать более сложные рандомизированные стратегии. В теории поведения автоматов таким стратегиям соответствуют стохастические автоматы с переменной структурой. Большинство из них реализуют рандомизированные правила выбора следующего вида:

$$p_{n+1} = R_n(v_1, v_2, \dots, v_n; p_1, p_2, \dots, p_n; \xi_1, \xi_2, \dots, \xi_n), n = 1, 2, \dots, \quad (7)$$

где  $R_n$  – вектор-функция со значениями в симплексе  $S^n$ ,

$p_n$  – вектор условных вероятностей выбора вариантов  $v(1), v(2), \dots, v(N)$  в момент времени  $t_n$ .

Среди рандомизированных правил выбора (7) известны марковские правила:

$$p_{n+1} = Q_n(v_n, p_n, \xi_n), n = 1, 2, \dots \quad (8)$$

Рандомизированные стратегии, определяемые последовательностью правил вида (8) относятся к классу рекуррентных алгоритмов адаптивного выбора вариантов. Эти алгоритмы достаточно просто реализуются, поскольку они на каждом шаге  $n$  используют минимальную информацию о предыстории процесса.

В результате проведенного анализа методов адаптивного выбора вариантов при оптимизации систем с дискретным временем, оказалось возможным предложить следующую классификацию классов-моделей

системы «информационная среда – программируемый робот».

#### Класс А.

Тип состояния информационной системы: полная информация

Функция потерь:  $\xi_n = \{1, 0\}$  бинарная

Стратегия поведения:  $v_{n+1} = T_n(\omega)$

Модель поведения: автоматная модель поведения,

Тип автомата: детерминированный, с постоянной структурой

#### Класс С.

Тип состояния информационной системы: априорная неопределенность

Функция потерь:  $\xi_n = \{1, 0\}$  бинарная

Стратегия поведения:  $v_{n+1} = T_n(v_1, v_2, \dots, v_n; \xi_1, \xi_2, \dots, \xi_n; \omega)$

Модель поведения: автоматная модель поведения,

Тип автомата: детерминированный,

#### Класс Е.

Тип состояния информационной системы: априорная неопределенность

Функция потерь:  $\xi_n = \{1,0\}$  бинарная

Стратегия поведения:

$$P_{n+1} = R_n(v_1, v_2, \dots, v_n; p_1, p_2, \dots, p_n; \xi_1, \xi_2, \dots, \xi_n)$$

Модель поведения: автоматная модель поведения,

Тип автомата: детерминированный, стохастический с переменной структурой

Задачу о поведении автомата, иллюстрирующую смысл этого подхода, можно сформулировать следующим образом. Рассматривается автомат [13,14] (система управления роботом) способный в каждый момент времени  $t = 1, 2, \dots$  воспринимать конечное число сигналов  $s \in (s_1, s_2, \dots, s_n)$  и изменять в зависимости от них свое внутреннее состояние; автомат может производить конечное число действий  $u \in (u_1, u_2, \dots, u_n)$ ; выбор действия определяется внутренним состоянием автомата; автомат имеет конечное число внутренних состояний  $\psi \in (\psi_1, \psi_2, \dots, \psi_m)$ , где  $m$  – емкость памяти автомата.

Предполагается, что автомат находится в некоторой среде, и что действия  $u$  автомата вызывают ответные реакции  $S$  среды. Эти реакции, в свою очередь, являются для автомата входными сигналами; автомат, так сказать, использует их для принятия решения о дальнейших действиях (рис. 2).

Рассмотрим случай, когда все возможные реакции среды  $s \in (s_1, s_2, \dots, s_n)$  воспринимаются автоматом как относящиеся к одному из двух классов – классу благоприятных реакций (выигрыш  $s=0$ ) и классу реакций неблагоприятных (проигрыш,  $s=1$ ). Целесообразность поведения автомата в некоторой среде заключается в увеличении числа благоприятных реакций и уменьшении числа реакций неблагоприятных. Для рассматриваемой в статье задачи программного управления выбраны детерминированные и стохастические автоматы.

Автомат задается уравнением  $u(t) = F(\psi(t))$ , указывающим зависимость действия  $u(t)$  от его состояния  $\psi(t)$ , и стохастической матрицей  $\|a_{ij}(s)\|, i, j = 1, 2, \dots, m$ . При этом  $a_{ij}(s)$  равно вероятности перехода состояния  $\psi(t) = \psi_i$  в состояние

$\psi(t+1) = \psi_j$  под воздействием входа  $s(t+1)$ . Для детерминированных автоматов матрицы  $\|a_{ij}(s)\|$  состоят из нулей и единиц. Так как рассматриваются автоматы, воспринимающие лишь два сигнала  $s=0$  и  $s=1$ , то достаточно задать две такие матрицы  $\|a_{ij}(0)\|$  и  $\|a_{ij}(1)\|$ . Таким образом, детерминированный автомат  $U$  может быть задан каноническими уравнениями:

$$\begin{aligned} \psi(t+1) &= \Phi(\psi(t), s(t+1)), \\ u(t) &= F(\psi(t)) \end{aligned} \quad (9)$$

Матрица состояний детерминированного автомата является простой: каждая ее строка при любом фиксированном значении  $s$  содержит один элемент, равный единице, а остальные элементы равны нулю. Переходы состояний детерминированного автомата определяются следующим образом: если в момент  $t$  автомат находится в состоянии  $\psi_i$  то в момент  $t+1$  он перейдет в такое состояние  $\psi_j$ , для которого  $a_{ij}(s(t+1)) = 1$ .

Стохастический автомат также имеет конечное число состояний  $\psi \in (\psi_1, \psi_2, \dots, \psi_m)$  и конечное число действий  $u \in (u_1, u_2, \dots, u_n)$ . Действия стохастического автомата однозначно определяются его состоянием:  $u(t) = F(\psi(t))$ , а матрицы состояний  $\|a_{ij}(s)\|, s \in \{0,1\}$  являются стохастическими. При этом  $a_{ij}(s)$  имеет смысл вероятности перехода из  $i$ -го состояния в  $j$ -е при заданном значении входной переменной  $s$ .

Пусть в момент  $t$  автомат находится в состоянии  $\psi_i, i = 1, 2, \dots, m$ , которому соответствует действие  $u_r = F(\psi_i)$ . Тогда вероятность  $p_{ij}$  перехода автомата из состояния  $\psi_i$  в состояние  $\psi_j$  определяется формулой:

$$p_{ij} = p_r a_{ij}(1) + q_r a_{ij}(0), \quad i, j = 1, 2, \dots, m. \quad (10)$$

В работе [13] показано, что поведение автомата в стационарной случайной среде описывается конечной цепью Маркова. Существуют финальные вероятности

состояний, и можно определить математическое ожидание выигрыша автомата, не зависящее от начального состояния.

Обозначим через  $\tau_i$  финальную вероятность состояния  $\psi_i$  автомата, находящегося в стационарной случайной среде  $C$ , а через  $\delta_r$  – сумму финальных вероятностей таких состояний  $\psi_i$ , которым соответствует действие  $u_r$ . Величины  $\delta_r$  имеют смысл вероятностей действия  $u_r$  автомата  $U$  в среде  $C$ .

Математическое ожидание  $W(U, C)$  выигрыша для автомата  $U$  в среде  $C$  определяется выражением:

$$W(U, C) = \sum_{r=1}^n \delta_r a_r \quad (11)$$

В работах по асимптотически оптимальным конечным симметрическим автоматам, например, [15] доказана оптимальность автоматов с линейной тактикой в некоторых стационарных случайных средах.

Рассмотрим поведение классического конечного автомата с линейной тактикой. Принцип функционирования такого автомата напоминает «ромашку». Число лепестков «ромашки» равно числу действий автомата (возможных управлений робота). На рис. 3 показан вариант, когда число таких действий равно трем (вперёд, вправо и влево). В каждом лепестке выделено четыре устойчивых состояния, в которых может находиться автомат. В любом из состояний, образующих лепесток ромашки, устройство выдает в среду сигнал действия, приписанный этому лепестку. Смена состояний происходит с учетом сигналов оценок за действие, поступающих от внешней среды (расстояние до препятствия впереди, справа и слева). Как было рассмотрено выше, это двоичные сигналы {1,0}. При поступлении сигнала 0, «не штраф», автомат меняет свое состояние и переходит в более глубокое состояние. Автомат как бы переходит к внешнему краю лепестка, а когда достигает последнего, то остается в нем.

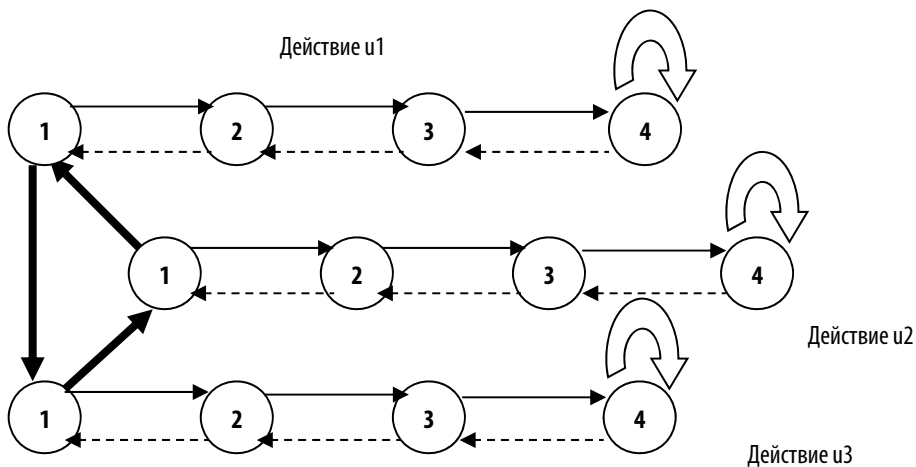


Рис. 3. Структура автомата с линейной тактикой

Если же на вход автомата приходит сигнал 1, «штраф», то состояния меняются в соответствии с пунктирными стрелками. В этом случае автомат идет внутрь лепестка и в какой-то момент под влиянием сигнала штрафа переходит на другой лепесток ромашки и происходит смена действий автомата. Смена лепестков, а, соответственно, и действий происходит поочередно.

Структура матриц переходов  $\|a_{ij}(s)\|$ ,  $s=0,1$  для действия  $u_2$ .

Для  $s=0$   $\|a_{ij}(s)\| =$

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
|          | $\psi_1$ | $\psi_2$ | $\psi_3$ | $\psi_4$ |
| $\psi_1$ |          | 1        |          |          |
| $\psi_2$ |          |          | 1        |          |
| $\psi_3$ |          |          |          | 1        |
| $\psi_4$ |          |          |          | 1        |



Для  $s=1$  $\|a_{ij}(s)\| =$ 

|          | $\psi_1$ | $\psi_2$ | $\psi_3$ | $\psi_4$ |
|----------|----------|----------|----------|----------|
| $\psi_1$ | 1*       |          |          |          |
| $\psi_2$ | 1        |          |          |          |
| $\psi_3$ |          | 1        |          |          |
| $\psi_4$ |          |          | 1        |          |

Особенность представленной выше матрицы состоит в следующем (символ 1\*), если автомат выполнял действие  $u_2$  и находился в состоянии  $\psi_1$ , при этом реакция среды  $s=1$ , то он должен поменять тип действия  $u_2$  и перейти в состояние  $\psi_1$ .

### ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

Технология управления перемещением мобильного устройства вдоль заранее заданной траектории с помехами (возможны препятствия на маршруте и нарушения ориентирующих маркеров) на основе метода, интегрирующего программное управление с обратными связями и адаптивный выбор вариантов, проверена на

системе управления роботом на платформе Arduino MotorShield. Система организована в виде двух уровней. На верхнем уровне интерпретатор команд управляющей программы обеспечивает управление перемещением робота вдоль заданного маршрута, ориентируясь на маркеры, в качестве которых выступают сенсорные данные ультразвукового датчика расстояния. На нижнем уровне модуль адаптивного выбора вариантов поддерживает управление роботом при появлении помех: получение дополнительной информации, необходимой для выбора варианта обхода помехи без отклонения от маршрута. Натурные эксперименты показали, что метод адаптивного выбора вариантов обеспечивает обход препятствий и восстановление движения робота вдоль траектории маршрута, а также восстановление маршрута при повреждении маркера маршрута, если погрешность альтернативного параметра лежит в пределах интервала перемещений, связанный с глубиной памяти автомата.

### ЛИТЕРАТУРА:

1. Baklanov A.I. Sistemi nabludenija i monitoringa. M.: Binom, – 2009. – 240 s.
2. Printsipi postroenija avtomatizirovannix sistem operativnogo kontrolja sostojanija potentsialno opasnich obektov / Makarov M.I., Korolev A.N., Pavlov S.V. // Bezopasnost giznedejatelnosti – 2004. – №12. – S.44-46.
3. Frolova M.V. Primenenie veb-technologij pri razrabotke raspredelennyh sistem monitoringa // Izvestija JFU. Technicheskie nauki – 2011. – №5. – S.24-31.
4. Sistemi protipoghegnogo zachistu DBN B.2.5-56:2014. K.: Minregion Ukraini, – 2015, –131s.
5. Nikolaev P.L. Arhitektura integrirovannoy v oblachnuyu sredu sistemi upravleniya umnym domom // Programnie produkti i sistemi – 2015. – №2 (110). – S.110-118.
6. Gradetskij V. G., Veshnikov V.B., Kalinichenko S. V., Kravchuk L. N. Upravljaemoe dvigienie mobilnich robotov po proizvolno orientirovannym v prostranstve poverxnostjam. M.: Nauka, – 2001. – 360 s.
7. Kargin A.O., Timchuk O.S., Isaenkov K.O., Galich G.B. Model sensohoji pamjati intelektualnoji mashini z mehanizmom uzagalnennja ta abstraguvannja // Sistemi ozbroennja ta vijskova tehnika / HUPS im. I. Kosheduba MOU, Kharkiv, – 2015. – №3(43). – S.85-88.
8. Ilin V.A. Algoritmi planirovanija povedenija integralnih robotov v uslovijah nepolnoj informatsii o structure vneshnej sredi. Tomsk: Izd-vo Tomsk. un-ta, – 1990. – 270 s.
9. Shevchuk B.M. The systems approach to problem solving calculations optimization objective means of sensor networks / B.M. Shevchuk // Information Technology and Computer Engineering. – 2013. - №1. – pp. 88-95.
10. Nazin A.V., Poznjak A.S. Adaptavnij vikor variantov: rekursivnie algoritmi. M.: Nauka, – 1986. – 288 s
11. Lujbchik L.M, Poznjak A.S. Obuchajuchie avtomati v zadachax upravlenija stohastichesrimi obektami // AiT – 1974. – №5. – S.95-109.
12. Trahtenbrot B.A., Barzdin J.M. Konetchnie avtomati (povedenie i sintez). M.: Mir, – 1970. – 400 s.
13. Kudrjavtsev V.B., Aleshin C.V., Podkolzin A.S. Vvedenie v teoriju avtomatov. M.: Nauka, –1985. –464 s.
14. Tsetlin M. L. Issledovanija po teorii avtomatovi modelirovanij biologicheskix sistem. . M.: Nauka, – 1969. – 316 s.
15. Ponomarev V.A. Ob odnoj konstruksii kontchnogo avtomata, asimptoticheski optimalnogo v statsionarnoj srede // Biofizika. – 1973. – t.9. – Vip. 1. – pp. 104-110.

**Рецензент:** д.т.н., проф. Филатов В.А.,  
заведующий кафедрой искусственного интеллекта,  
Харьковский национальный университет радиоэлектроники