

**В. Э. Карпов**, канд. техн. наук, доц., нач. лаб. робототехники, [karprov\\_ve@mail.ru](mailto:karprov_ve@mail.ru),  
**В. В. Воробьев**, зам. нач. лаб. робототехники, [gatus86@mail.com](mailto:gatus86@mail.com),  
**М. А. Ровбо**, науч. сотр., [rovboma@gmail.ru](mailto:rovboma@gmail.ru),  
Национальный исследовательский центр "Курчатовский институт", г. Москва

## О некоторых аспектах применения автоматных моделей в групповом управлении

*Рассматриваются особенности применения дискретно-событийных моделей для управления согласованным поведением групп агентов и предлагается механизм организации управления поведением на основе иерархии конечных автоматов. Базовый уровень управления движением агентов использует непрерывную модель совместного движения — метод потенциалов, в котором направление движения определяется итоговой суммой сил "отталкивания", "притяжения" и "выравнивания курса". В работе описывается множество частных правил группового движения агентов — когезия, распределение, следование за лидером, движение вдоль цепочки ориентиров, преследование и т. д. При этом все они трактуются в терминах результатов работы автоматных моделей. Было проведено математическое моделирование процесса управления группой агентов с использованием частных правил группового движения, которое подтвердило их работоспособность. Также были проведены вычислительные эксперименты с помощью 3D-симулятора Gazebo. Кроме того, в работе описывается экспериментальная группировка подводных аппаратов в составе трех экземпляров, с помощью которых отработывались частные правила группового движения: когезия, распределение, движение за лидером. Связь между аппаратами осуществлялась с помощью системы инфракрасных приемопередатчиков, по псевдоаналоговому каналу. Она позволяет устойчиво принимать сигнал, который является закодированным числом в диапазоне от нуля до пятнадцати, в воде на расстоянии до сорока—пятидесяти сантиметров. Эксперименты проводились в лабораторном бассейне и завершились успешно.*

*Показано, что технология создания систем группового управления, основанная на иерархии автоматов (метаавтоматов), позволяет решать задачи управления не только на уровне абстрактных моделей и имитационного моделирования, но и на уровне создания реальных робототехнических комплексов.*

**Ключевые слова:** конечный автомат, групповое управление, модели движения, поведение, метод потенциалов, автоматное управление, иерархия в управлении

### Введение

Одним из классических методов управления в технических системах являются дискретно-событийные модели в виде конечных автоматов (КА) или эквивалентные им механизмы. Важной особенностью КА является способность их формальной верификации. Более того, их разработка под конкретную задачу в отдельных случаях может быть автоматизирована, например с помощью формальных или даже эволюционных методов [1].

Применение конечно-автоматного управления в робототехнических системах имеет некоторую специфику. Если простые последовательности операций допускают использование КА как формы описания алгоритма функционирования, то более сложные задачи, связанные с реализацией поведения в сложных средах, требуют особого подхода. Так, при разработке комплексных программ поведения для роботов

удобными с точки зрения человеческого восприятия представляются так называемые деревья поведений [2], позволяющие представлять управляющие структуры в виде иерархии задач. При этом деревья поведений и КА эквиваленты по возможностям представления поведения.

Особое место автоматные модели занимают в групповом управлении, где КА используются, как правило, для определения верхнего, координирующего уровня управления [3]. Например, сначала группа роботов формирует построение, после чего приступает к исследованию территории, а в случае чрезмерного отклонения от заданного построения возвращается к перестроению. При этом базовые задачи стайного движения решаются путем формирования управляющих сигналов напрямую исходя из параметров окружающей среды. Это характерно для различных алгоритмов следования, движения строем [4], а также многочисленных моделей поведения на основе управления скоростью [5].

Решение задач, связанных с реализацией сложного поведения в динамически меняющейся среде, естественно, основывается на создании иерархии управления. Иерархии с верхним планирующим и нижним исполнительным уровнями применяются как для отдельных агентов [6], так и для групп агентов [7]. Тем не менее, существуют области группового управления, где не требуется решения интеллектуальных задач, требующих того же планирования или сложной координации действий. Обычно это такие задачи, в которых достаточно ограниченного локального взаимодействия и выбора того или иного поведения в зависимости от состояния ближайших соседей. Подобные системы называются роевыми, а в качестве показательного примера можно взять задачу согласованного движения группы подводных аппаратов. Помимо удобства исследования благодаря более свободному перемещению роботов в среде по сравнению с наземными роботами здесь имеется значительный потенциал для практического применения групповой робототехники [8], например, при решении задач мониторинга, исследования поверхности дна, поиска и спасения, сбора мусора и других. Так, в работе [9] с помощью группы подводных роботов демонстрируются результаты работы биологического алгоритма распределения задач.

Вместе с тем, подводная среда определяет принципиальные ограничения сенсорных и коммуникативных возможностей роботов. Это особенно актуально для малых аппаратов, несмотря на то, что в них могут использоваться камеры и габаритные маяки, как это было представлено в платформе *Blueswarm* [10]. Такая принципиальная ограниченность восприятия мира роботом позволяет говорить о том, что для управления его поведением достаточно моделей рефлекторного (дискретно-событийного) уровня.

Если рассматривать вместо алгоритмов планирования поведенческие шаблоны, переключаемые агентом в зависимости от ситуации, то образуется естественная иерархия управления. Совмещение высокоуровневого управления на основе КА, отвечающих за переключение низкоуровневых двигательных поведенческих процедур, является одним из способов организации сложного агента, способного выполнять различные модели поведения в рамках решения целевой задачи.

В предлагаемой работе рассматриваются особенности применения подхода на основе иерархии

КА для задачи управления группой подводных мобильных роботов. Сначала будет приведено описание основного объекта — агента как модели мобильного робототехнического устройства. Далее будут описаны законы его базового движения и рассмотрены вопросы представления сложного поведения на базе конечно-автоматных моделей. В заключительной части будут представлены вычислительные эксперименты, а также описание экспериментов с реальными объектами.

### Модель агента

Пусть у нас имеется некоторое устройство, оснащенное множеством сенсоров и способное к перемещению в трехмерном пространстве. Для определенности будем полагать, что это автономный необитаемый подводный аппарат (АНПА) или робот. Чтобы абстрагироваться от технических деталей, будем использовать вместо термина "АНПА" термин "агент".

Агент  $A_{fish}$  задается пятеркой

$$A_{fish} = \langle name, C, R, S, F \rangle, \quad (1)$$

где  $name$  — уникальное имя агента (идентификатор);  $C$  — множество характеристик агента;  $R$  — его пространственное положение;  $S$  — состояние сенсорной системы;  $F$  — функция управления.

Функция управления  $F$  определяет пространственное положение агента:

$$F(C, R_t, S) \rightarrow R_{t+1}. \quad (2)$$

Базовыми характеристиками агента является тройка

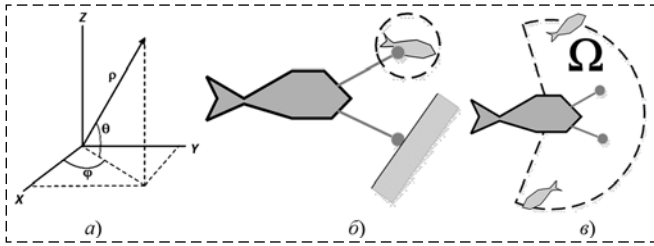
$$C = \langle tip, \omega, size \rangle,$$

где  $tip$  — элемент некоторого множества возможных типов агентов. Используется для реализации разных видов поведения по отношению к соответствующим категориям агентов (*свой, чужой, жертва, нарушитель* и т. п.);  $\omega$  — вес агента,  $\omega \in \mathbb{R}$ . Используется как основная характеристика при реализации иерархии управления;  $size$  — размер агента,  $size \in \mathbb{R}$ . Определяет геометрические характеристики агента.

В зависимости от специфики решаемых задач набор характеристик  $C$  может расширяться.

Пространственное положение  $R$  задается тройкой

$$R = \langle X, v, A \rangle, \quad (3)$$



**Рис. 1. Сферическая система координат агента:**

$v = \rho$ ,  $\alpha_{yaw} = \varphi$ ,  $\alpha_{pitch} = \theta$  (а); датчики-дальномеры, (б); область видимости агента  $\Omega$  (в)

**Fig. 1. Agent's spherical coordinate frame:**

$v = \rho$ ,  $\alpha_{yaw} = \varphi$ ,  $\alpha_{pitch} = \theta$  (а); rangefinders (б); agent's field of view  $\Omega$  (в)

где  $X$  — координаты агента,  $X \in \mathbb{R}^3$ ;  $v$  — скорость движения;  $A$  — тройка углов Эйлера,  $A = (\alpha_{yaw}, \alpha_{pitch}, \alpha_{roll})$ ;  $\alpha_{yaw}$ ,  $\alpha_{pitch}$ ,  $\alpha_{roll}$  — углы рыскания, тангажа и крена соответственно. Фактически, пара  $(v, A)$  определяет сферические координаты агента, кроме крена, который во многих задачах, связанных с групповым движением, не важен. Использование углов Эйлера обусловлено лишь соображениями удобства и наглядности.

Знание координат агента  $X$  не является обязательным, явное задание  $X$  определяется исключительно общностью рассуждений. Далее будем полагать, что скорость  $v$  и углы  $\alpha_{yaw}$ ,  $\alpha_{pitch}$  соответствуют длине радиус-вектора  $\rho$  и углам  $\varphi$  и  $\theta$  (рис. 1, а).

### Сенсорная система

**"Контактные" сенсоры-дальномеры.** Для удобства управления и решения некоторых тактических задач будем полагать, что у агента имеются два сенсора-отрезка, определяющие объекты, которые пересекаются этими отрезками. Точки пересечения определяют расстояние до обнаруженных объектов (рис. 1, б).

**Область видимости.** Предположим, что сенсорика агента позволяет получить информацию о координатах, скоростях и углах ориентации некоторого множества агентов  $\Omega_s$ , попавших в его зону видимости — некоторую область пространства  $\mathbb{R}^3$  вблизи агента (рис. 1, в).  $\Omega_s$  может представлять собой множество индексов — идентификаторов агентов, а может содержать полный список наблюдаемых агентов целиком.

Обозначим  $\langle X_s, v_s, A_s \rangle$  — средние значения координат, скорости и ориентации наблюдаемых в этой области агентов. При этом пусть они вычисляются исходя из весов  $\omega_i$  агентов. Тогда компоненты координат центра тяжести наблюдаемых агентов вычисляются как

$$X_{sk} = \frac{\sum_{i \in \Omega_s} X_{ik}}{\sum_{i \in \Omega_s} \omega_i}, \quad k = 1, 2, 3. \quad (4)$$

Здесь  $X_{ik}$  — компоненты вектора координат  $X_i$  наблюдаемого агента.

Аналогично определяются средние скорости  $v_s$  и вектор ориентации  $A_s$ :

$$v_s = \frac{\sum_{i \in \Omega_s} v_i \omega_i}{\sum_{i \in \Omega_s} \omega_i}, \quad k = 1, 2, 3, \quad (5)$$

где  $v_i$  — скорости агентов;  $A_s$  определяется как итоговая ориентация суммы векторов скоростей агентов  $v_i$ .

Соответствующие вектору управления  $v_{ctl} = (v_x, v_y, v_z)$  координаты также можно записать в сферических координатах через углы  $\varphi = \alpha_{yaw}$  и  $\theta = \alpha_{pitch}$  вокруг осей  $Oz$  в виде

$$v_{ctl} = |(v_x, v_y, v_z)| = \sqrt{v_x^2 + v_y^2 + v_z^2}; \quad (6)$$

$$\varphi = \arctg \frac{v_y}{v_x}; \quad \theta = \arctg \frac{\sqrt{v_x^2 + v_y^2}}{v_z}.$$

Область видимости задается либо полной сферой наблюдения, либо только передней полусферой (что ближе к реальности). Обозначим радиус сферы наблюдения  $FOV$ . Это параметр используется при определении законов движения.

При описании алгоритмов мы будем использовать следующие соглашения. Обращение к сенсорике агента  $a$  будем обозначать  $a.Sensors[T]$ , где  $T$  — категория или тип детектируемого объекта. Это относится как к случаю обращения к объектам, попавшим в область видимости  $\Omega$  агента, так и к обращению к управляющим параметрам или измеряемым величинам. Так, определение целевого угла движения определяется содержимым некоторого виртуального сенсора (псевдосенсора)  $a.Sensors['goal\_angle'] = \langle \alpha_{yaw}, \alpha_{pitch}, \alpha_{roll} \rangle$ . Аналогичным образом будем обозначать обращение к объектам области видимости, препятствий и т. п.

### Законы движения

Рассмотрим далее ряд законов согласованного движения группы агентов. Эти законы основаны на классической модели движения К. Рейнольдса [11]. Общие правила движения агентов (бойдов — boids — в терминологию

гии Рейнольдса) просты и заключаются в том, что агент (а) не должен приближаться близко к другим агентам, (б) должен двигаться туда, куда все и (в) стремиться не отдаляться от других. Таким образом, реализуются следующие правила стайного движения:

1. *Отталкивание (Separation)*. Избегание скопления соседей, ближнее отталкивание.

2. *Выравнивание курса (Alignment)*. Придерживаться среднего курса соседей.

3. *Когезия (Cohesion)*. Ориентация на среднее положение соседей, "притяжение" с больших расстояний.

Эти действующие на агента "силы" условно изображены на рис. 2.

На самом деле, правила движения агентов естественным образом описываются в терминах потенциальных полей, каждое из которых определяется действием соответствующего правила движения. Все программы движения, определяющие действия того или иного правила или закона, возвращают вектор функции управления  $F$  (строго говоря — кортеж из двух элементов):

$$Prog(agent) \rightarrow F = (v_{ctl}, [A_{yaw}, A_{pitch}, A_{roll}]), \quad (7)$$

представляющий линейную скорость и ориентацию в виде тройки углов Эйлера соответственно. Модуль вектора скорости будет также обозначаться  $|F|$ , а ориентация —  $arg(F)$ .

Далее рассмотрим их более подробно. При этом будем полагать, что на входе имеются состояние агента  $\langle X, v, A \rangle$  и наблюдаемые величины  $\langle X_s, v_s, A_s \rangle$ , а искомым выходом является управление  $\langle v_{ctrl}, A_{ctrl} \rangle$ .

**Составление (притяжение, Cohesion)**. В общем случае целевая ориентация определяется как

$$A_{ctl} = arg(F_A(X, X_s, A_s)), \quad (8)$$

$$v_{ctl} = |F_A(X, X_s, A_s)|;$$

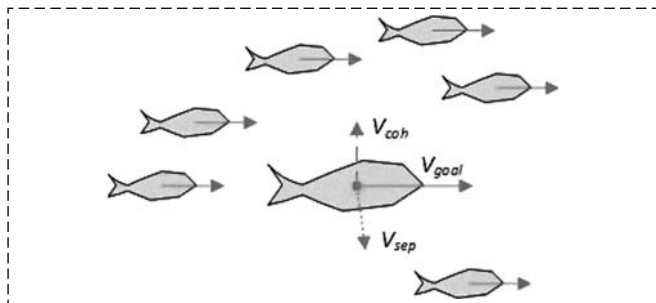


Рис. 2. "Силы", действующие на агента:

$V_{coh}$  — когезия,  $V_{sep}$  — отталкивание,  $V_{goal}$  — целевое направление движения

Fig. 2. "Forces" acting on the agent:

$V_{coh}$  — cohesion,  $V_{sep}$  — separation,  $V_{goal}$  — direction to the goal

пример реализации:

$$A_{ctl} = arg(F_A^{coh}(X, X_s, A_s)). \quad (9)$$

Итоговая скорость в общем случае может определяться собственной скоростью агента и расстоянием до центра масс группы  $r_s$ :

$$v_{ctl} = |F_v^{coh}(v, r_s)| = \min\left(v_{max}, \frac{k_v}{r_s}\right). \quad (10)$$

Здесь  $v_{max}$  — предельная скорость движения;  $k_v$  — некоторый коэффициент.

Направление движения может быть определено из списка всех наблюдаемых "своих".

В векторной форме скорость когезии будет определяться как

$$dv_{ctl} = \frac{X_s}{|X_s|}. \quad (11)$$

$A_{ctl}$  здесь и далее рассчитывается как поворот вектора, сонаправленного с результирующим вектором  $v_{ctl}$ . Для этого можно использовать формулу перехода в сферические координаты. Пример реализации:

$$v_{ctl}^{(i+1)} = v_{ctl}^{(i)} + dv_{ctl}. \quad (12)$$

Здесь  $v_{ctl}^{(i+1)}$  — новая скорость (на текущем шаге);  $v_{ctl}^{(i)}$  — текущая скорость (с прошлого шага). Рассчитанная скорость также ограничивается предельной скоростью движения  $v_{max}$ :

$$|v_{ctl}^{(i+1)}| = \min(v_{max}, |v_{ctl}^{(i+1)}|).$$

**Распределение (отталкивание, Separation)**. Итоговый угол ориентации  $A_{ctl}$  определяется как

$$A_{ctl} = arg(F_A^{sep}(X, X_s, A_s)). \quad (13)$$

Направление при управлении  $F_A^{sep}$  отличается от  $F_A^{coh}$  знаком:  $F_A^{sep} = -F_A^{coh}$ . Скорость определяется так же:

$$|F_A^{sep}(v, r_s)| = |F_A^{coh}(v, r_s)|.$$

По сути и по форме алгоритм распределения аналогичен рассмотренному выше, при этом центр масс также определяется списком видимых "своих"  $Sensors[\omega\_friend]$ . Кроме того, эффект отталкивания может быть реали-

зован и через следующее определение вектора скорости движения:

$$dvel_{ctl} = - \frac{\sum_{i \in \Omega_s} \frac{X_i}{|X_i|}}{\left| \sum_{i \in \Omega_s} \frac{X_i}{|X_i|} \right|}. \quad (14)$$

Аналогичным образом определяются прочие законы движения.

**Движение в одну сторону (Alignment).** Задача — придерживаться среднего курса соседей:

$$v_{ctl} = \frac{v_s}{|v_s|}.$$

**Движение по непосредственно заданному закону.** Закон трактуется как задаваемые псевдосенсором "Цель" требуемые линейная и угловая скорости:

$$\langle v_{ctl}, A_{ctl} \rangle = \text{Sensors}[\text{'goal\_movement'}].$$

Этот закон определяет и частный случай — движение по пеленгу. В последнем случае скорость движения может не задаваться.

**Преследование.** Будем считать задачи преследования и сближения эквивалентными. Оптимальное сближение может быть реализовано методом параллельного преследования, описанного в работе [12]. В этом методе требуется знать местоположение агентов (хотя существуют модели преследования, в которых координаты агентов не используются, например модель, рассмотренная в работе [13]).

Пусть  $(x_v, y_v, z_v)$  — координаты преследуемого,  $v_v$  — его скорость, а  $\alpha_{yaw}^v, \alpha_{pitch}^v$  — углы поворота вокруг осей  $OZ$  и  $OY$  соответственно. Аналогично обозначим координаты  $(x_h, y_h, z_h)$  и скорость преследователя  $v_h$ . Результирующие компоненты угла ориентации  $\alpha_{yaw}^h, \alpha_{pitch}^h, \alpha_{roll}^h$  определяются следующим образом:

$$\begin{aligned} \alpha_{yaw} &= F_A^{purs}([x_v, y_v, v_v, \alpha_{yaw}^v], [x_h, y_h, v_h]); \\ \alpha_{pitch} &= F_A^{purs}([x_v, z_v, v_v, \alpha_{pitch}^v], [x_h, z_h, v_h]); \\ \alpha_{roll} &= 0; \\ A_{ctl} &= (\alpha_{yaw}, \alpha_{pitch}, \alpha_{roll}). \end{aligned} \quad (15)$$

Скорость преследователя постоянна и определяется как  $v_{ctl} = v_{max}$ .

На практике  $F_A^{purs}$  может задаваться не как функция от координат преследователя и пре-

следуемого, а определяться углом  $\gamma$  между ними в соответствующей плоскости. Пусть  $\gamma$  — угол между преследователем и преследуемым;  $\alpha_v, u_v$  — направление движения и скорость убегающего;  $V$  — скорость преследователя. Тогда направление движения  $\varphi$  преследователя определяется как

$$\varphi = \arctg \frac{v_{hy}}{v_{hx}} - \gamma. \quad (16)$$

**Комплексное движение (движение "бойдов").**

Здесь движение агента определяется действием рассмотренных выше частных законов — отталкиванием, выравниванием курса и когезией.

Скорость представляется как

$$v_{ctl} = \omega_a dvel_a + \omega_s dvel_s + \omega_c dvel_c, \quad (17)$$

где  $dvel_a, dvel_s, dvel_c$  — управление, определяемое поведением *Alignment, Separation* и *Cohesion* соответственно;  $v_{ctl}$  также ограничивается максимальной скоростью  $v_{max}$ .

Добавление к когезии, распределению и движению в одну сторону законов движения по заданному пеленгу *FBearing* и непосредственно заданной скорости *FSetSpeed* приводит к комплексному алгоритму, где компоненты их скоростей помножаются на коэффициенты, определяющие значимость этих факторов, а затем векторы скоростей складываются друг с другом.

**Движение вдоль цепочки ориентиров.** Ориентирами являются некоторые наблюдаемые агенты — буи. На входе — наблюдаемые в зоне видимости буи  $\Omega_B = \langle X_B, v_B, A_B \rangle$ .

Пусть ориентир  $i$  из области  $\Omega_B$  находится на расстоянии  $r_i = |X - X_i|$  от агента. Обозначим  $A_i$  — углы, задающие направление от агента на ориентир. Если местоположения заданы декартовыми координатами, то  $r_i, A_i$  определяются как сферические координаты точки

$$(X_B^{(1)} - X^{(1)}, X_B^{(2)} - X^{(2)}, X_B^{(3)} - X^{(3)}). \quad (18)$$

Здесь  $X_B^{(i)}, X^{(i)}$  — координаты буя и агента соответственно.

Обозначим  $d_i^a$  — рассогласование направлений векторов ориентации агента и вектора направления на ориентир:

$$d_i^a = A - A_i.$$

Тогда кортеж  $(v_s, A_s)$  определяется как сумма векторов:

$$(v_s, A_s) = \sum_{i \in \Omega_B} (F_i, A_i).$$

Здесь  $F_i$  — значение "силы притяжения" ориентира  $i$  из области видимости  $\Omega_B$ :

$$F_i = \begin{cases} 0, & \text{если } r_i < r_{\min}; \\ \frac{K}{r_i^n}, & \text{иначе,} \end{cases} \quad (19)$$

$$K = \cos \max_j (d_i^a)_j.$$

Смысл введения коэффициента  $K$  заключается в том, чтобы для агента были более предпочтительны те ориентиры, которые расположены по ходу движения агента. Чтобы не увеличивать влияние более близких ориентиров, целесообразно в знаменателе иметь расстояние  $r_i$  в степени  $n > 1$ . Кроме того, совсем близкие ориентиры не должны участвовать в определении направления движения. Для этого вводится некоторое минимальное расстояние  $r_{\min}$ , кратное, например, размеру агента.

В простейшем случае скорость движения агента может быть постоянной. Здесь важным является определение направления движения.

В итоге  $v_{ctl} = \text{const}$ ,  $A_{ctl} = A_s$ .

Перейдем далее к описанию более сложных форм поведения агента.

### Автоматное управление

Итак, выше были рассмотрены законы, определяющие движение агентов. В определенном смысле они могут считаться базовыми, элементарными действиями. Вместе с тем, такое понятие, как поведение, если под ним понимать устоявшиеся формы и способы взаимодействия агента со средой, требует для своего описания иного подхода.

**Поведение и действия.** Понятия "поведение", "действие", "активность" являются во многом условными, но между ними всегда можно установить однозначную границу. В целях упорядочения механизмов управления агентами примем следующие соглашения. Будем рассматривать организацию поведения с точки зрения следующей иерархии, используя вместо термина "агент" термин "робот".

1. *Эффекторный уровень.* Это — нижний, "аппаратный" уровень управления. На нем ре-

ализуются отработка эффекторов робота, отвечающих за скорость движения, углы поворота и т. п. В общем случае это элементарные воздействия на окружающую среду.

2. *Действие.* Под действием мы будем понимать некую явно выделяемую совокупность эффекторной активности. На этом уровне происходит согласованная работа эффекторов. Примерами простых действий могут быть различного рода движения.

3. *Сложное действие.* Это некоторая характерная, повторяющаяся последовательность действий. В биологии это называется фиксированным комплексом действий — ФКД.

4. *Базовое поведение.* Некоторая характерная, повторяющаяся последовательность действий (в том числе сложных), которая связана с достижением некоторой цели управления. Примерами базового поведения может быть защитная реакция, поисковое поведение, целенаправленное движение и т. п.

5. *Поведение.* В целом, это устоявшаяся форма или множество форм взаимодействия агента со средой.

**Реализация.** С технической точки зрения реализация действий, и простых, и сложных, вплоть до ФКД, удобна с помощью КА с выходом. В рамках рассмотренных выше моделей выходом автомата Мили, описывающего действия, является запуск моторной программы — независимого движения; движения, определяемого когезией; следования за лидером и т. п. Входом автомата являются некие внешние условия. Формально КА работает со входным алфавитом из двух символов — *true* и *false*, при этом ус-

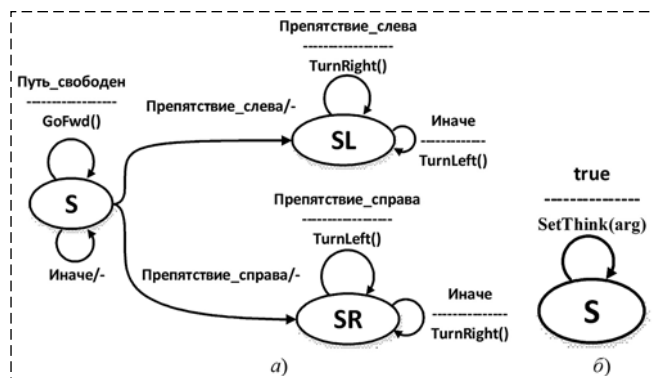


Рис. 3. Автомат fsm\_MoveObstSide — движение вдоль препятствия (а); автомат fsm\_DirMovementProc, реализующий закон движения, задаваемый аргументом arg (б)

Fig. 3. Finite state machine fsm\_MoveObstSide — movement along an obstacle (a); finite state machine fsm\_DirMovementProc, that implements a movement law defined by the parameter arg (b)

ловия определяются вычисляемыми в каждый момент времени предикатными функциями. Поскольку здесь мы имеем дело с недетерминированным автоматом, то для автомата вводится приоритет выполнения условий перехода. Примеры автоматов, реализующих действия, приведены на рис. 3.

На рис. 3, а приведен автомат, определяющий такое сложное действие, как движение вдоль препятствия, где S — состояние, в котором агент движется прямо, а SL и SR — состояния, в котором он поворачивает направо и налево, соответственно. На рис. 3, б — автомат, реализующий прямое назначение поведенческой процедуры. Аргумент этой процедуры определяет некоторую программу движения — когезию, следование за лидером и т. п., т. е. такие программы, которые не описываются в автоматном виде. Здесь S — состояние агента, в котором отработывается эта поведенческая процедура.

Для реализации поведения требуется уже другой, более сложный механизм. Таким механизмом может быть метаавтомат.

**Метаавтомат.** Суть его работы заключается в том, чтобы в зависимости от результатов анализа состояния внешней и внутренней среды активизировать тот или иной автомат, реализующий некоторое действие. На рис. 4 приведен метаавтомат, реализующий процедуру движения по пеленгу, задаваемому аргументом *arg*.

Метаавтомат представляет собой автомат Мили, у которого пометки дуг (процедуры) определяют выполняемые агентом действия. Выполнение этих процедур заключается в том, что загружаются соответствующие автоматы, т. е. автоматы, реализующие действия вида "Обход препятствия" (*fsm\_MoveObstSide*), "Стайное движение" (*fsm\_DirMovementProc* с аргументом 'FFlockMovement'). Условия переходов

(загрузки новых КА) определяются, например, сенсорикой агента (в данном контексте это означает определение того, что путь свободен (предикат "Путь\_Свободен"). В примере S — стартовое состояние метаавтомата, S1 — состояние в котором определяется, свободен ли путь к цели, а SFree и SObst — состояния, характерные для отсутствия на пути препятствия и его наличия, соответственно. Более сложное поведение может задаваться работой метаавтоматов второго и т. д. рангов. Формально многоуровневая метаавтоматная схема определяется следующим образом.

Если действиями КА являются активизации низкоуровневых двигательных функций, то мы считаем его автоматом ранга 0 —  $M^0$ . Метаавтомат ранга 1  $M^1$  — это КА, действия которого заключаются в управлении автоматами  $M^0$  (запуск, останов, возобновление работы). Таким образом, метаавтомат ранга  $n$   $M^n$  управляет автоматами ранга  $M^{n-1}$ . Введение понятия метаавтомата — это структуризация иерархии управления в автоматном представлении. Основа автоматов всех рангов — это КА с выходом. Таким образом, мы получаем следующую иерархию компонентов архитектуры (в порядке усложнения):

- 1) *эффлекторный уровень.* Реализуется моделями движения;
- 2) *действия.* Реализуется множеством КА. При этом для простоты полагалось, что в каждый момент времени активен только один — текущий — автомат, т. е. агент не может совершать несколько действий одновременно;
- 3) *базовое поведение.* Поведенческие реакции как последовательности действий определяются метаавтоматами ранга 1;
- 4) *поведение.* Определяется метаавтоматами различных рангов.

Итак, мы имеем двухуровневую систему:

1. *Базовый уровень.* Реактивные модели прямого управления движением. Здесь осуществляется формирование вектора управления движением на основе анализа сенсорных данных. На этом уровне реализуются рассмотренные выше двигательные программы (сближение, распределение, преследование и т. д.).

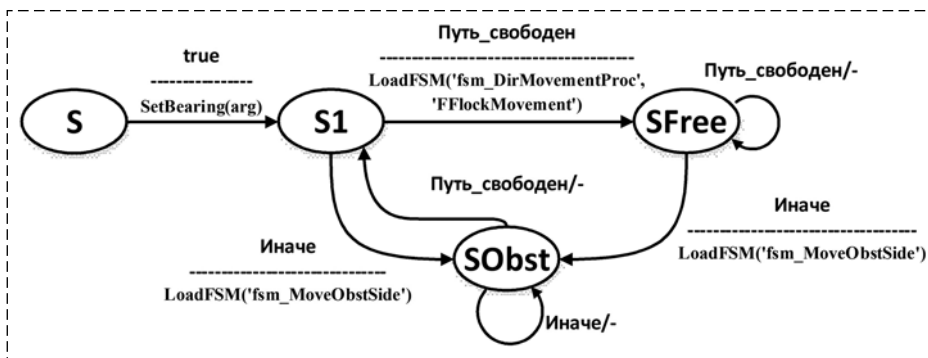


Рис. 4. Мета-автомат, реализующий движение по пеленгу  
Fig. 4. Meta-finite state machine that implements movement according to bearing

2. *Уровень поведения.* Здесь требуется понятие состояния (память) и применяются дискретно-событийные (автоматные) модели. На этом уровне определяются последовательности двигательных программ в виде системы метаавтоматов, формируя сложные поведенческие процедуры типа охоты, движения по пеленгу с обходом препятствий, патрулирования и т. п.

### Вычислительные эксперименты

Исходя из метаавтоматного принципа реализации поведения общий цикл управления выглядит следующим образом: для каждого агента сначала обрабатывается процедура чтения сенсоров, затем выполняется шаг автоматной логики, после чего определяются скорость и углы движения агента, и осуществляется его передвижение.

В качестве первоначальной, упрощенной кинематической модели движения с точечными агентами использовалась разработанная на *Python* симуляция с визуализацией в *RViz* и *Matplotlib* (рис. 5, а, см. вторую сторону обложки). Для апробации использовалась задача групповой охоты и патрулирования в коридоре с препятствиями и опасностями. Здесь опасности — это объекты, реакцией на которые является рассредоточение. Для моделирования группы роботов в более реалистичных условиях использовался симулятор *Gazebo*, облегчающего перенос разработанного кода между моделью и аппаратной реализацией роботов (рис. 5, б, см. вторую сторону обложки).

Использование симулятора *Gazebo* позволяет добавлять реалистичные датчики и использовать объемные модели агентов. При симуляции применялись лишь кинематические модели роботов, так как для исследования группового поведения агентов динамикой можно пренебречь, поскольку она не оказывает значительного влияния на взаимодействие агентов. При апробации использовалась группа из одного лидера,двигающегося самостоятельно по круговой траектории большого радиуса, и агентов с алгоритмом бойдов *FFlockMovement*, которые двигаются за лидером (рис. 5, б, см. вторую сторону обложки). Следует отметить, что агенты считают лидера равноправным членом группы, никак не выделяя его среди других.

Наряду с вычислительными были проведены и натурные эксперименты на группе малых подводных аппаратов. Основная задача — отработка в условиях лабораторного бассейна элементов законов движения и вопросов сенсорики. Внешний вид аппаратов приведен на рис. 6 (см. вторую сторону обложки). Они представляют собой герметичный отсек, расположенный на ходовой платформе. Платформа оснащена парой маршевых двигателей, которые задают скорость движения и угол рыскания, а также парой водометных движителей в носовой части, задающих угол тангажа.

Кроме того, каждый аппарат оснащен все-направленным излучателем модулированного сигнала в ИК диапазоне и соответствующими приемниками (рис. 7, а).

Этот сигнал задает такие характеристики, как тип наблюдаемого объекта ("опасность", "свой", "чужой"), "сила" (опасности своего и чужого), а также позволяет оценить расстояние до источника. Для этого генерируемый сигнал представляет собой последовательности пачек импульсов, модулированных несущей частотой 36 кГц. Принимаемая огибающая пачек  $w$  делится далее на диапазоны, определяющие тип источника сигнала, а интервал между импульсами  $r$  позволяет оценить расстояние до источника (см. рис. 7, б).

Изменение параметров  $w$  и  $r$  обусловлено характером распространения сигнала в плотной среде, а точнее, потерями. Подобного рода организация ИК связи позволяет получать до-

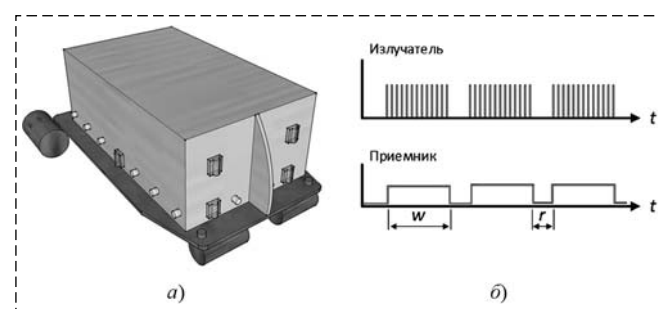


Рис. 7. Расположение сенсоров (ИК приемники) и излучателей (а); излучаемые и принимаемые сигналы: определение типа и параметров излучателя по величине  $w$  и расстояния до источника по значению  $r$  (б)

Fig. 7. Sensor placement (IR-receivers) and emitters (а); emitted and received signals: type and emitter parameter identification by the value of  $w$  and distance to the source by the value of  $r$  (б)

статочно устойчивую сенсорную картину на небольшом (до нескольких линейных размеров робота) расстоянии.

Эксперименты заключались в проверке работоспособности сенсорной системы и отработке законов стайного движения. Кроме того, важно было определить актуальные представления данных об объектах в области видимости агентов. Очевидно, что реальная сенсорная система не способна определить область  $\Omega$ , как список наблюдаемых агентов с определенными для них координатами, скоростями, габаритными размерами и т. п. Вместо этого регистрации подлежит информация о наличии некоторого агента в зоне видимости, дискретизированной до уровня, определяемого числом и расположением приемников ИК сигналов, а также косвенным определением расстояния и скорости (по последовательности измерения параметра  $r$  сигнала) наблюдаемого агента. Такое радикальное загромождение, тем не менее, оказалось приемлемым для реализации простых видов поведения типа стайного движения за лидером. Более сложные формы движения (движение с учетом глобальной цели, движение по ориентирам, преследование) требует, разумеется, развития сенсорной системы робота.

### Заключение

В работе предложена некоторая технология применения дискретно-событийных моделей для описания поведения агентов при решении задач группового управления. Важно, что уровень базовых движений, являясь принципиально "непрерывным" (в этой работе в его основе — различного рода модификации метода потенциальных полей), может быть естественным образом интегрирован в дискретно-событийную схему, а вопрос систематизации поведения различного уровня может быть решен иерархией метаавтоматных моделей. В этой статье не рассматривались вопросы вычислительной или емкостной эффективности. Сложность алгоритмов уровня базовых движений остается прежней. Речь идет о некоторой единой технологии применения автоматных моделей для решения подобного рода задач.

Более серьезной проблемой являются ограничения сенсорной системы реального робота по сравнению с тем, что "видит" модельный

агент. Перспективным представляется создание конструкций, оснащенных камерами и габаритными маяками, однако в данной работе мы рассматриваем принципы управления, поэтому улучшение сенсорной системы остается за ее рамками и является предметом дальнейших исследований.

### Список литературы

1. Бужинский И. П., Казаков С. В., Ульянов В. И., Царев Ф. Н., Шальто А. А. Модификация метода генерации управляющих конечных автоматов с непрерывными воздействиями по обучающим примерам // Известия Российской академии наук. Теория и системы управления. 2015. № 6. С. 17—30.
2. Iovino M., Scukins E., Styrud J., Ögren P., Smith C. A survey of Behavior Trees in robotics and AI // Robotics and Autonomous System. 2022. Vol. 154. P. 104096.
3. Saad S. Ben, Zerr B., Probst I., Dambreville F. Hybrid coordination strategy of a group of cooperating autonomous underwater vehicles // IFAC-PapersOnLine. 2015. Vol. 28, N. 5. P. 47—52.
4. Зенкевич С. Л., Назарова А. В., Хуа Ч. Моделирование и анализ движения группы мобильных роботов в среде ROS // Мехатроника, автоматизация, управление. 2017. Т. 18, № 5. С. 317—320.
5. Reynolds C. W. Steering behaviors for autonomous characters // Game Developers Conference. 1999. P. 763—782.
6. Макаров Д. А., Панов А. И., Яковлев К. С. Архитектура многоуровневой интеллектуальной системы управления беспилотными летательными аппаратами // Искусственный интеллект и принятие решений. 2015. № 3. С. 18—33.
7. Панов А. И., Яковлев К. С. Взаимодействие стратегического и тактического планирования поведения коалиции агентов в динамической среде // Искусственный интеллект и принятие решений. 2016. № 4. С. 68—78.
8. Hamann H. Swarm Robotics: A Formal Approach. Springer Cham., 2018. 221 p.
9. Zahadat P., Schmickl T. Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition // Adaptive Behavior. 2016. Vol. 24, N. 2. P. 87—101.
10. Berlinger F., Gauci M., Nagpal R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm // Science Robotics. 2021. Vol. 6, N. 50, P. 1—14.
11. Reynolds C. W. Flocks, herds, and schools: A distributed behavioral model // Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. 1987. Vol. 21, N. 4. P. 25—34.
12. Петросян Л. А., Рихсиев Б. Б. Преследование на плоскости. М.: Наука. 1991. 96 с.
13. Yu J., LaValle S. M., Liberzon D. Rendezvous Without Coordinates // IEEE Transactions on Automatic Control. 2012. Vol. 57, N. 2. P. 421—434.

# About Some Aspects of Finite State Machine Models Application to Group Control

V. E. Karpov, karpov\_ve@mail.ru, V. V. Vorobiev, gatus86@mail.ru, M. A. Rovbo, rovboma@gmail.com,  
National Research Center "Kurchatov Institute", Moscow, 123182, Russian Federation

Corresponding author: Vorobiev V. V., Deputy Head of Robotics Laboratory,  
National Research Center "Kurchatov Institute", Moscow, 123182, Russian Federation,  
e-mail: e-mail: gatus86@mail.ru

Accepted on November 22, 2023

## Abstract

The article discusses using discrete-event models to control the coordinated behavior of groups of agents and proposes a mechanism for organizing behavior control based on a hierarchy of finite state machines (FSM). The basic level of agent movement control uses a continuous model of joint movement — the method of potentials, in which the direction of movement is determined by the total sum of the forces of "repulsion", "attraction" and "course alignment". The paper describes a set of independent rules for the group movement of agents — cohesion, distribution, leader following, movement along a chain of landmarks, pursuit, etc. Moreover, all of them are interpreted in terms of the results of the operation of FSM models. Mathematical modeling of controlling a group of agents was carried out using the independent rules of group movement, which confirmed their efficacy. Computational experiments were also carried out using the Gazebo 3D simulator. In addition, the paper describes an experimental group of three underwater vehicles that were used to test the independent rules of group movement: cohesion, distribution, leader following. Communication between the devices was carried out using a system of infrared transceivers via a pseudo-analog channel. The experiments were carried out in a laboratory pool and were successful.

The paper shows that the technology for creating group control systems based on the hierarchy of FSM (meta-FSM) allows solving control problems not only at the level of abstract models and simulation modeling, but also at the level of creating real robotic complexes.

**Keywords:** finite state machine, group control, movement models, behavior, method of potentials, state machine control, control hierarchy

For citation:

Karpov V. E., Vorobiev V. V., Rovbo M. A. About Some Aspects of Finite State Machine Models Application to Group Control, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2023, vol. 24, no. 4, pp. 171–180.

DOI: 10.17587/mau.24.171-180

## References

1. Buzhinskij I. P., Kazakov S. V., Ul'jancev V. I., Carev F. N., Shalyto A. A. Modification of the Generation Method for Finite State Machines with Continuous Actions Using Training Examples, *Izvestija Rossijskoj Akademii Nauk. Teorija I Sistemy Upravlenija*, 2015, no. 6, pp. 17–30 (in Russian).
2. Iovino M., Scukins E., Styrud J., Ögren P., Smith C. A survey of Behavior Trees in robotics and AI, *Robotics and Autonomous Systems*, 2022, vol. 154, p. 104096.
3. Saad S. Ben, Zerr B., Probst I., Dambreville F. Hybrid coordination strategy of a group of cooperating autonomous underwater vehicles, *IFAC-PapersOnLine*, 2015, vol. 28, no. 5, pp. 47–52.
4. Zenkevich S. L., Nazarova A. V., Zhu H. Simulation and Analysis of the Movement of a Group of Mobile Robots in ROS, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no. 5, pp. 317–320.
5. Reynolds C. W. Steering behaviors for autonomous characters, *Game Developers Conference*, 1999, pp. 763–782.
6. Makarov D. A., Panov A. I., Jakovlev K. S. Architecture Of The Multilayered Intelligent Control System For Unmanned Aerial Vehicles, *Iskusstvennyj Intellekt I Prinjatje Reshenij*, 2015, no. 3, pp. 18–33 (in Russian).
7. Panov A. I., Jakovlev K. S. On Interaction Of Strategic And Tactical Planning For The Coalition Of Agents In Dynamic Environment, *Iskusstvennyj Intellekt I Prinjatje Reshenij*, 2016, no. 4, pp. 68–78 (in Russian).
8. Hamann H. Swarm Robotics: A Formal Approach, Springer Cham, 2018, 221 p.
9. Zahadat P., Schmickl T. Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition, *Adaptive Behavior*, 2016, vol. 24, no. 2, pp. 87–101.
10. Berlinger F., Gauci M., Nagpal R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm, *Science Robotics*, 2021, vol. 6, no. 50, pp. 1–14.
11. Reynolds C. W. Flocks, herds, and schools: A distributed behavioral model, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 1987, vol. 21, no. 4, pp. 25–34.
12. Petrosjan L. A., Rihsiev B. B. Pursuit on a flat surface, Moscow, Nauka, 1991, 96 p. (in Russian).
13. Yu J., LaValle S. M., Liberzon D. Rendezvous Without Coordinates, *IEEE Transactions on Automatic Control*, 2012, vol. 57, no. 2, pp. 421–434.