УДК 004.852

Иерархический регулятор на основе алгоритма обучения с подкреплением для реконфигурируемого многомодульного шагающего мобильного робота

Р. А. Мунасыпов[1], Т. Р. Шахмаметьев[2], С. С. Москвичев[3], И. Х. Хамадеев[4]

[1] rust40@mail.ru, [2] shahmametevtr@gmail.com, [3] mosk.sergey@gmail.com, [4] iamilgiz@mail.ru

ФГБОУ ВПО «Уфимский государственный авиационный технический университет» (УГАТУ)

Аннотация. Шагающие роботы способны преодолевать разнообразные препятствия и передвигаться по сложному рельефу. Однако реализация всех преимуществ шагающей конфигурации возможна лишь при использовании регулятора соответствующей сложности. В статье описано применение метода обучения с подкреплением для решения задачи преодоления препятствий реконфигурируемым шагающим мобильным роботом. Алгоритм основан на двухуровневой иерархической декомпозиции задачи, в которой регулятор верхнего уровня выбирает траекторию движения, а регулятор нижнего уровня осуществляет передвижение конечностей в заданные положения. Данный подход позволяет роботу успешно преодолевать препятствия и передвигаться в незнакомой среде.

Ключевые слова: потенциальное поле; обучение с подкреплением; вихревое поле; реконфигурируемый мобильный робот

Колесные и гусеничные приводы, применяемые в транспортных средствах и робототехнических системах, более энергоэффективны, чем шагающие и ползающие механизмы. В то же время шагающие механизмы обладают более высокой проходимостью и способны преодолевать препятствия, сравнимые с размерами самого робота. В данной статье мы рассматриваем возможность использования алгоритмов обучения с подкреплением в задаче синтеза регулятора для многомодульного шагающего робота. Данный подход может обеспечить роботу высокую степень проходимости и позволит преодолевать самые разные типы препятствий, включая заранее неизвестные.

Предлагаемый метод планирования движения многомодульного шагающего робота основан на обучении с подкреплением с использованием двухуровневой иерархической декомпозиции задачи. При наличии препятствия (такого как ступень, рис. 1) планировщик верхнего уровня генерирует оптимальную траекторию перемещения ступней робота. Задачей планировщика нижнего уровня является обеспечение движения ступней по заданным траекториям на основе планирования движения суставов каждой ноги с учетом сохранения равновесия робота и предотвращения столкновений ног с препятствием.

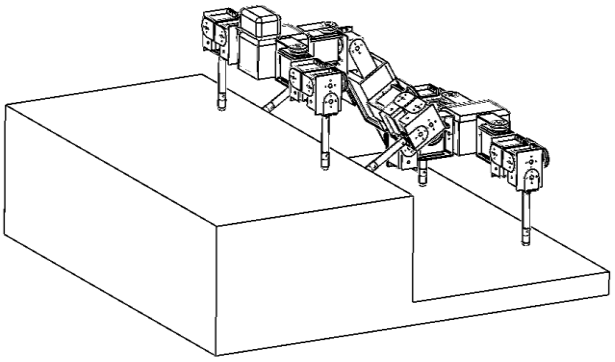

Рис. 1. Препятствие типа «ступень»

Планировщик нижнего уровня не требует многоступенчатого анализа и работает на очень коротких временных интервалах, поскольку должен обеспечивать координацию и равновесие робота. Для обучения планировщика нижнего уровня при выборе оптимальной траектории движения суставов каждой ноги робота используются алгоритмы поиска оптимальных стратегий с простыми параметризованными стратегиями.

Работа выполнена при финансовой поддержке РФФИ (гранты №№ 11-08-00871_а, 11-08-97063_р_поволжье_а).

Планирование траектории движения мобильного робота на верхнем уровне основано на анализе последовательности постановки ступней шагающего механизма и требует более длительных временных интервалов. В разработанном методе планирования выбор оптимальной траектории движения осуществляется с использованием лучевого алгоритма поиска на основе метода потенциалов. Для оценки приемлемости различных положений стоп робота используется функция стоимости, формируемая с использованием алгоритма обучения с подкреплением.

1. МОДЕЛЬ РЕКОНФИГУРИРУЕМОГО МОБИЛЬНОГО РОБОТА

Для решения поставленной задачи был построен реконфигурируемый многомодульный шагающий мобильный робот, показанный на рис. 2. Габариты робота составляют приблизительно 0,3 м в высоту, 0,5 м в ширину и 0,81 м в длину, вес составляет около 7,5 кг. В каждой ноге имеется три сервомотора: два в тазобедренном шарнире для вращения верхней части (бедра) каждой ноги вперед/назад и вверх/вниз; и один в коленном шарнире для вращения нижней части (голени) каждой ноги внутрь/наружу, а также два сервомотора в позвонковом шарнире между модулями [1, 2]. Сервомеханизмы имеют максимальный крутящий момент 25 кг-см. Задачей регулятора является синтез последовательности команд (курсовых углов) для данных сервомоторов, которые позволят роботу преодолеть препятствие.

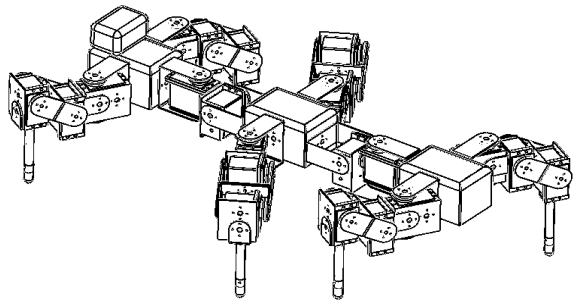

Рис. 2. Многомодульный мобильный робот «Легион»

Состояние робота полностью определяется его позицией (x, y, z); ориентацией (угол крена ψ_{k1}, угол тангажа ψ_{k2}, угол рыскания ψ_{k3}, где k – это номер модуля, т. е. углы ориентации, определяются для каждого модуля в зависимости от двух шарниров ε_{k1} и ε_{k2} для каждого позвонка относительно первоначального модуля, к которому привязаны мировые координаты); а также шестью шарнирными углами у ног для каждого модуля, т. е. в нашем случае восемнадцать шарнирных углов $\xi_1, \ldots, \xi_{18}$ при соответствующих частотах перемещения и угловых частотах вращения $\dot{x}, \dot{y}, \dot{z}, \dot{\psi}_{k1}, \dot{\psi}_{k2}, \dot{\psi}_{k3}, \dot{\xi}_i, \ldots, \dot{\xi}_{18}$. Значит, в нашем случае для шестиногой конфигурации это дает 60-мерное пространство состояний. Но данный регулятор нижнего уровня выбирает команды только как функцию переменных (x, y, z, ψ_{k1}, ψ_{k2}, ψ_{k3}, ξ_{i1}, ξ_{i2}, ξ_{i3}) которые перекрывают 22-мерное пространство [3–6]. То есть, для того чтобы решить задачу, мы будем изменять 22-мерное подпространство состояний за время t, обозначаемое как $\Omega_t = [\omega_1, \ldots, \omega_{22}]^T$. На рис. 3 мы видим, как кинематическая модель нашего робота и трансформация координат будут использованы для вычисления расположения его соединений. Таким образом, получаются стандартные кинематические схемы, которые будут использоваться для подсчета всех позиций данных ступней Ω_t [5]. Позиция ступни i в момент времени t определяется как:

$$u_t^i = T_{\text{мир-стопа}}(x, y, z, \psi_{k1}, \psi_{k2}, \psi_{k3}, \xi_{i1}, \xi_{i2}, \xi_{i3}), \quad (1)$$

где i_1, i_2, i_3 – индексы для трех соединений ступни i. Значение $T_{\text{мир-стопа}}$ может быть получено из простых кинематических вычислений.

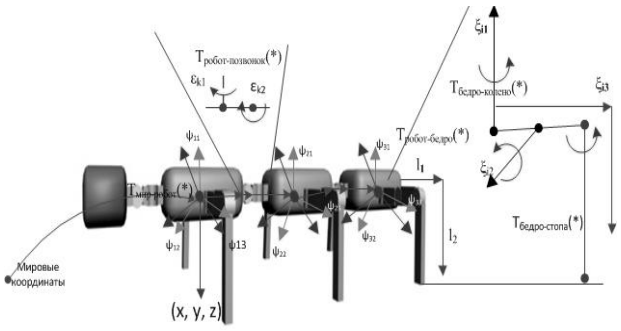

Рис. 3. Состояния и координаты многомодульного мобильного робота

2. РЕГУЛЯТОР НИЖНЕГО УРОВНЯ

Основная задача регулятора нижнего уровня – генерировать последовательность команд, которые передвигают ступню в заданную целевую позицию, при этом для обеспечения динамической и статической устойчивости робота остаются неподвижными как минимум три ступни. Регулятор отображает текущую и целевую позиции перемещаемой i-й ступни в команды управления 22 сервоприводами. Регулятор нижнего уровня генерирует параметры движе-

ния ступней таким образом, чтобы ступни не столкнулись с возможными препятствиями во время движения к цели. Поэтому был выбран метод дискретизации, основанный на потенциальных полях.

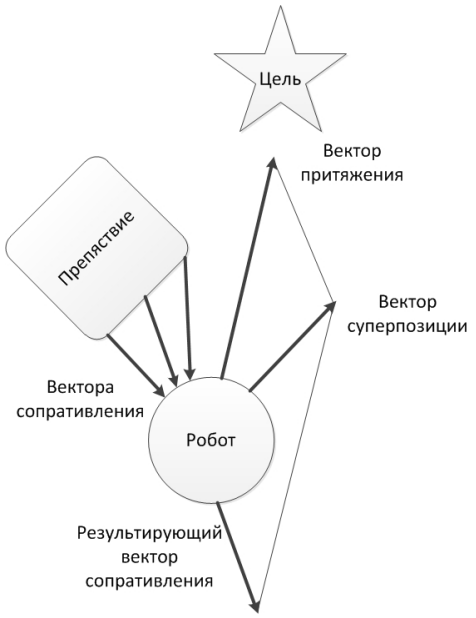

Рис. 4. Схема векторов потенциального поля

Потенциальные поля часто используются в задачах планирования движений робота для поиска траектории движения к цели с учетом обхода препятствий. Для этого необходимо определить вектор «потенциал притяжения» на цель (этот потенциал представлен функцией над пространством состояний, которая убывает по мере приближения к цели); и вектор «потенциал отталкивания» на препятствия (представленный функциями, которые возрастают, когда мы приближаемся к препятствию). Общая функция потенциала – это сумма всех потенциалов притяжения и отталкивания (рис. 4). На каждом шаге робот движется, используя направление наискорейшего спуска через общую функцию потенциала. Таким образом, регулятор нижнего уровня формирует оптимальную траекторию ступни, используя для этого потенциальное поле, состоящее из трех функций:

1) потенциал цели: поле потенциала притяжения, которое заставляет робот двигать ногу в направлении позиции конкретной цели, выбранной планировщиком верхнего уровня.

2) потенциал поверхности: потенциал отталкивания, удерживающий движущуюся ступню на некотором расстоянии от препятствий.

3) потенциал положения: функция потенциала, способствующая поддержанию устойчивого положения робота в пространстве.

Для решения задачи движения ступни i из существующей позиции u_i в целевую u_{ig} можно определить следующий результирующий потенциал:

$$U_t(\Omega_t, u_t^i, u_g^i) = U_t^g(u_t^i, u_g^i) + U_t^s(u_t^i, u_g^i) + \\ + U_t^p(\Omega_t), \quad (2)$$

где $U_t^s(\cdot)$, $U_t^g(\cdot)$ и $U_t^p(\cdot)$ представляют собой потенциалы цели, поверхности и положения соответственно. Для совершения движения в каждый момент времени робот вычисляет отрицательный градиент потенциальной функции, вычисляемый для текущего положения:

$$\tilde{g}_{t,j} = -\nabla_{\omega j} U_t(\Omega_t, u_t^i, u_g^i),\ j = 1,...,33, \quad (3)$$

в соответствии с ним робот изменяет шарнирные углы соответствующей ноги.

Следует отметить, что простое следование за градиентом потенциальной функции будет работать только при отсутствии локальных минимумов. Возможна ситуация, когда непосредственно вблизи препятствия потенциал притяжения цели и потенциал отталкивания поверхности компенсируют друг друга, что приводит к образованию локального минимума потенциальной функции. Для решения данной проблемы может быть введен дополнительный член – вихревое поле, формируемое вокруг препятствия. Использование вихревого поля позволит движущейся ноге не только избегать столкновений с препятствиями, но и огибать их. В ситуациях, когда потенциал притяжения цели и потенциал отталкивания поверхности практически скомпенсированы, вихревое поле направляет движущуюся ступню вверх над препятствием.

Дополнительным требованием к алгоритму является необходимость наличия трех опорных ступней во время движения. Допустим, что на каждом шаге робот пытается изменить свое положение в каком-либо направлении $\hat{g}_t$ (полученном из суммы градиента U_t и вектора вихревого поля), при этом изменяется положение и опорных (неподвижных) ступней u_t^{s1}, u_t^{s2}, и u_t^{s3}, где $s1$, $s2$ и $s3$ – индексы трех опорных ступней в момент времени t. Чтобы избежать этого, формируется подпространство движений,

в котором положение опорных стоп остается неизменным. Другими словами, определяем $\Phi^t \in R^{30\times 15}$ как матрицу, столбцы которой являются градиентами компонентов u_t^{s1}, u_t^{s2} и u_t^{s3} относительно 30 переменных состояния. Изменение положения опорных ступней из-за малых изменений переменных состояния $\delta\Omega_t$ составляет приблизительно $\Phi_t^T \cdot \delta\Omega_t$ Таким образом, чтобы сохранить опорные стопы в покое, робот должен двигаться в направлениях, находящихся в нуль-пространстве Φ_t^T. Пусть $\hat{g}_t^*$ – это проекция $\hat{g}_t$ в нуль-пространство Φ_t^T. Нахождение $\hat{g}_t^*$ может рассматриваться как задача минимизации:

$$\min \left\| \hat{g}_t^* - \hat{g}_t \right\|_2,$$
$$\Phi_t^T \cdot \hat{g}_t^* = 0.$$

Решение в аналитическом виде относительно $\hat{g}_t^*$:

$$\hat{g}_t^* = (I - \Phi_t \cdot (\Phi_t^T \cdot \Phi_t)^{-1} \cdot \Phi_t^T) \cdot \hat{g}_t. \quad (4)$$

Таким образом, мы меняем шарнирные углы в направлении $\hat{g}_t^*$. Минимизация этих трех потенциальных функций приведет к тому, что робот будет двигать ступню к цели, сохраняя равновесие и избегая столкновений с препятствиями или землей. Проекция в нуль-пространство Φ_t не позволит поддерживающей ступне двигаться.

3. РЕГУЛЯТОР ВЕРХНЕГО УРОВНЯ

Планирование последовательности шагов представляет собой сложную задачу поиска, поскольку плохой выбор позиции ступней, сделанный ранее в последовательности, может привести робота в «плохую» позицию (например, неудобное положение), из которой далее будет сложно продолжать движение. Полный перебор всех возможных последовательностей позиций ступней невозможен из-за высокого фактора ветвления и большой глубины поиска. Вместо этого можно использовать некоторую оценку того, насколько «хорошо» текущее положение робота. Например, при использовании алгоритма поиска *A** эвристическая функция «действие–стоимость» позволяет оценить оптимальную будущую стоимость из любой заданной позиции до цели. Однако бывает проблематично выбрать подходящую эвристическую функцию стоимости в сложных задачах, так как это требует оценки всей последовательности неизвестных будущих затрат. В обучении с подкреплением «добротность» состояния s определяется оптимальной функцией стоимости $V^*(s)$. Проще говоря, это наибольшая ожидаемая сумма будущих вознаграждений, начиная с состояния s. Используя обучение с подкреплением, мы автоматически определим приближенную функцию стоимости.

Обучение с подкреплением

В обучении с подкреплением модель объекта управления имеет набор возможных состояний S и набор возможных действий A [8]. Также имеется функция вознаграждений $R : S \times A \to R$ такая, что $R(s,a)$ – вознаграждение за действие a в состоянии s. Динамика системы описывается функцией переходов $F : S \times A \to S$, при этом $s' = F(s,a)$ – состояние, достигаемое путем выполнения действия a в состоянии s. Стратегия – это любая функция π: $S \to A$, отображающая состояния в действия. Можно говорить, что реализация стратегии π заключается в выполнении действия $a = \pi(s)$ при нахождении в состоянии s. Функция стоимости $V^\pi(s_0)$ для стратегии π определяется как сумма дисконтированных вознаграждений, которую можно получить, начиная из состояния s_0 и руководствуясь стратегией π:

$$V^\pi(s_0) = R(s_0, a_0) + \gamma R(s_1, a_1) + \\ + \gamma^2 R(s_2, a_2) + ..., \quad (5)$$

где $a_i = \pi(s_i)$, $s_{i+1} = F(s_i, a_i)$, а $\gamma \in [0,1)$ – дисконт-фактор. Наличие дисконт-фактора приводит к тому, что вознаграждения в отдаленном будущем имеют меньший вес, чем вознаграждения в ближайшем будущем, т. е. он определяет «жадность» алгоритма. Оптимальная функция стоимости определяется как

$$V^*(a) = \max_\pi V^\pi(a).$$

Это наилучшая из возможных сумм дисконтированных наград, которая может быть достигнута путем реализации какой-либо политики. Оптимальная функция стоимости удовлетворяет равенству Беллмана:

$$V^*(s) = \max_{\pi} R(s,a) + \gamma V^*(s'), \qquad (6)$$

где *s'* – состояние, достигаемое при выполнении действия *a* в состоянии *s*. Иными словами, максимальная возможная ценность в текущем состоянии *s* определяется как максимальная сумма непосредственного вознаграждения за один шаг $R(s,a)$ и будущих наград, начиная со следующего состояния *s'*. Когда $V^*(\cdot)$ известно, нахождение политики, дающей максимальную сумму дисконтированных наград, становится простой задачей. Оптимальная политика $\pi^*(s) : S \to A$ определяется как

$$\pi^*(s) = \arg\max_{a} R(s,a) + \gamma V^*(s'), \qquad (7)$$

В нашем планировщике верхнего уровня выполнение действия *a* соответствует выбору нового положения для одной из ног. Использованная нами функция вознаграждений дает положительные вознаграждения (поощрения) за движение центра тяжести робота к цели; небольшие отрицательные вознаграждения (наказания) за время, затрачиваемое на выполнение движения; и значительные наказания в случае, когда регулятор нижнего уровня не смог выполнить требуемое действие.

Приближенная функция стоимости

Существует множество алгоритмов обучения с подкреплением для точного определения оптимальной функции стоимости. Однако большинство из них применимо лишь к задачам с небольшим, конечным пространством состояний. Для задач с большими непрерывными пространствами состояний найти непосредственно $V^*(\cdot)$, как правило, невозможно. В таких случаях используется приближенная функция стоимости. Как правило, для этого применяется линейная комбинация множества признаков состояния *s*. В частности, используется следующая приближенная функция стоимости:

$$\hat{V} = \theta^T \phi(s),$$

где $\phi(s)$ – вектор признаков состояния *s*, θ – вектор искомых параметров.

В нашем случае $\phi(s)$ состоит из следующих признаков состояния: расстояние от центра тяжести робота до цели; среднее расстояние от опорных точек ног до цели; положение тела, описываемое как (1 – cos (ψ_1), 1 – cos (ψ_2), 1 − cos (ψ_3)); максимальный угол поворота коленного сустава; коэффициент неровности рельефа под каждой ногой; наклон рельефа; перепад высот между самой верхней и самой нижней из текущих точек опоры; площадь треугольника, образованного тремя опорными ногами; радиус окружности, описанной вокруг опорного треугольника; расстояния между каждой из пар ног.

Алгоритм обучения

Для нахождения приближенной функции стоимости может быть использован алгоритм обучения с частичным подкреплением, основанный на методе опорных векторов [9]. В основе метода лежит следующий эмпирический факт: для того чтобы $\pi(s)$ (стратегия выбора действий на основе приближенной функции стоимости *V*) совпадала с оптимальной стратегией $\pi^*(s)$, необходимо и достаточно, чтобы выполнялось следующее неравенство:

$$\begin{aligned} &R(s,\pi^*(s)) + \gamma \hat{V}(F(s,\pi^*(s))) > \\ &> R(s,a) + \gamma \hat{V}(F(s,a)). \end{aligned} \qquad (8)$$

для всех действий $a \neq \pi^*(s)$. Таким образом, если у нас имеется обучающая выборка, состоящая из кортежей (s_i, a_i^*, a_i), $i = 1,\ldots, m$, где $s_i \in S$, $a_i^* = \pi^*(s_i)$, $a_i \neq \pi^*(s_i)$, то можно попытаться отыскать приближенную функцию стоимости $\hat{V}$, которая удовлетворяет приведенному неравенству для всех кортежей в обучающей выборке. $\hat{V}(s) = \theta^{\mathrm{T}}\phi(s)$ линейна относительно параметров θ, что означает наличие линейных ограничений, и для поиска параметров может быть использован любой из стандартных алгоритмов линейной классификации [9, 10].

В нашем случае имеет место еще одно уточнение, значительно облегчающее решение задачи. При определенных допущениях можно сказать, что оптимальная функция стоимости $V^*(s)$ является единственным решением равенств Беллмана (6). Так, одним из возможных методов поиска приближенной функции стоимости является поиск таких параметров θ, которые наиболее близки к значениям, удовлетворяющим равенствам Беллмана. Более строго, требуется минимизировать квадрат ошибки Беллмана:

$$\min_{\theta} \sum \left| V(s_1) - R(s_1, a_i^*) - \gamma \hat{V}(F(s_i, a_i^*)) \right|^2, \qquad (9)$$

где s_i и a_i^* взяты из обучающей выборки, $\hat{V}(s) = \theta^{\mathrm{T}}\phi(s)$ [11].

С учетом (8) и (9) получаем следующую задачу оптимизации:

$$\max_{\theta,\delta}\ \delta - \beta\sum\left|V(s_1) - R(s_1,a_i^*) - \gamma\hat{V}(F(s_i,a_i^*))\right|^2 - \\ -\alpha\left\|\theta\right\|_2^2,$$

$$R(s_i,a_i^*) + \gamma\hat{V}(F(s_i,a_i^*)) - R(s_i,a_i) - \\ -\gamma\hat{V}(F(s_i,a_i)) \geq \delta,\ i = 1,...,m, \qquad (10)$$

где α и β – константы, определяющие относительный вес двух членов задачи оптимизации. Данная задача является квадратичной программой и может быть решена непосредственно [12]. Заметим, что ограничения в данной задаче соответствуют требованию удовлетворения неравенства (8) для всех кортежей в обучающей выборке (полагаем $\delta > 0$), более того, необходимо, чтобы левая часть неравенства была больше правой на величину, не меньшую, чем δ («зазор»).

При $\beta = 0$ задача сводится к оптимизации (8), при этом алгоритм становится похож на метод опорных векторов [9] с параметрами θ при обучении классификатора различать оптимальные действия a_i* и субоптимальные a_i. При малом или нулевом α задача сводится к минимизации квадрата ошибки Беллмана (9). Но, решая задачу оптимизации обеих целевых функций, мы получаем более эффективный алгоритм.

ВЫВОДЫ

Нами был рассмотрен иерархический двухуровневый регулятор на основе алгоритма обучения с подкреплением для решения задач по преодолению препятствий шагающим роботом. Анализ показал, что данная иерархическая архитектура имеет большие возможности для управления процессами преодоления препятствий и движением по пересеченной местности шагающих (двуногих, четвероногих или шестиногих) роботов.

СПИСОК ЛИТЕРАТУРЫ

1. **Шахмаметьев Т. Р.** Принципы построения системы управления реконфигурируемого мобильного робота // Актуальные проблемы науки и техники: матер. 5-й Всеросс. зимн. шк. аспирантов и молодых ученых (Уфа, 17–20 февраля 2010 г.). Уфа: УГАТУ, 2011. С. 348–351.

2. **Мунасыпов Р. А., Москвичев С. С.** Методика синтеза стратегии движения автономного мобильного робота на основе эволюционных процессов // Вестник УГАТУ. 2012. Т. 16, № 3 (48). С. 56–62.

3. **Sutton R., Precup D., Singh S.** Intra-option learning about temporally abstract actions // Int. Conf. Machine Learning, vol. 98, pp. 556–564, 1998.

4. **Hauskrecht M., Meuleau N., Boutilier C., Kaelbling L. P., Dean T.** Hierarchical solution of markov decision processes using macroactions // in Proc. 14th Conf. on Uncertainty in Artificial Intelligence (UAI-98), 1998.

5. **Parr R., Russell S.** Reinforcement learning with hierarchies of machines // Advances in neural information processing systems, pp. 1043–1049, 1998.

6. **Dietterich T. G.** Hierarchical reinforcement learning with the maxqvalue function decomposition // Journal of Artificial Intelligence Research, vol. 13, pp. 227–303, 1999.

7. **Craig J.** Introduction to Robotics: Mechanics and Control. 2nd ed. Addison-Wesley Longman Publishing, 1989.

8. **Sutton R. S., Barto A. G.** Reinforcement Learning. MIT Press, 1998.

9. **Vapnik V. N.** Statistical Learning Theory. John Wiley & Sons, 1998.

10. **McCullaghand P., Nelder J. A.** Generalized Linear Models (2nd edition). Chapman and Hall, 1989.

11. **Baird L. C.** Residual algorithms: Reinforcement learning with function approximation // in Proc. 12th Int. Conf. on Machine Learning, pp. 30–37, 1995.

12. **Boyd S., Vandenberghe L.** Convex Optimization. Cambridge University Press, 2004.

ОБ АВТОРАХ

МУНАСЫПОВ Рустэм Анварович, проф. каф. техн. кибернетики. Дипл. инж. электрон. техн. (УАИ, 1982). Д-р техн. наук по сист. анализу, управ. и обраб. инф. (УГАТУ, 2003). Иссл. в обл. интел. и адапт. систем управ. слож. динам. объектами, интел. систем, микроробототехники.

ШАХМАМЕТЬЕВ Тимур Рашитович, мл. науч. сотр. той же каф. Дипл. инж. (УГАТУ, 2009). Готовит дис. о сист. упр. многомодульными шагающими мобильными роботами.

МОСКВИЧЕВ Сергей Сергеевич, мл. науч. сотр. той же каф. Магистр техн. и технол. (УГАТУ, 2008). Готовит дис. о сист. упр. автономными мобильными роботами.

ХАМАДЕЕВ Ильгиз Ханифович, магистрант той же каф.

METADATA

Title: The hierarchical controller based on reinforcement learning algorithm for a multimodule reconfigurable mobile walking robot.

Authors: R. A. Munasypov, T. R. Shakhmametyev, S. S. Moskvichev, and I. K. Khamadeev.

Affiliation: Ufa State Aviation Technical University (UGATU), Russia.

Email: mosk.sergey@gmail.com.

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 17, no. 5 (58), pp. 31-37, 2013. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: Walking robots are able to traverse a variety of obstacles and move around complex landforms. However, the implementation of all the features of the walking configuration is only possible by using a sufficiently complex controller. The article describes a reinforcement learning method applied to solving the problem of traversing ob-

stacles by a reconfigurable walking robot. The algorithm is based on a two-level hierarchical decomposition of the problem, where the high-level controller makes decisions on the trajectory, and the low-level controller moves the limbs into the desired positions. This approach allows the robot to successfully overcome obstacles and move around in unknown environments.

Key words: potential field; reinforcement learning; vortex field; reconfigurable mobile robot.

References (English Transliteration):

1. T. R. Shakhmametyev, "Principles of control system design for a reconfigurable mobile robot" (in Russian), in *Proc. 5th All-Russian Winter School*, Ufa, Russia, 2011, pp. 348-351.
2. R. A. Munasypov and S. S. Moskvichev, "Synthesis of locomotion strategy for an autonomous mobile robot using evolutionary methods," (in Russian), *Vestnik UGATU*, vol. 16, no. 3 (48), pp. 56-62, 2012.
3. R. Sutton, D. Precup, and S. Singh, "Intra-option learning about temporally abstract actions," in *Proc. Int. Conf. Machine Learning*, vol. 98, pp. 556-564, 1998.
4. M. Hauskrecht, N. Meuleau, C. Boutilier, L. P. Kaelbling, and T. Dean, "Hierarchical solution of markov decision processes using macroactions," in *Proc. 14th Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, 1998.
5. R. Parr and S. Russell, "Reinforcement learning with hierarchies of machines," *Advances in Neural Information Processing Systems*, pp. 1043-1049, 1998.
6. T. G. Dietterich, "Hierarchical reinforcement learning with the maxqvalue function decomposition," *J. Artificial Intelligence Research*, vol. 13, pp. 227-303, 1999.
7. J. Craig, *Introduction to Robotics: Mechanics and Control, 2nd ed*. Addison-Wesley Longman Publishing, 1989.
8. R. S. Sutton and A. G. Barto, *Reinforcement Learning*. MIT Press, 1998.
9. V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.
10. P. McCullaghand and J. A. Nelder, *Generalized Linear Models (2nd edition)*. Chapman and Hall, 1989.
11. L. C. Baird, "Residual algorithms: Reinforcement learning with function approximation", in *Proc. 12th Int. Conf. on Machine Learning*, pp. 30-37, 1995.
12. S. Boyd, L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

About authors:

MUNASYPOV, Rustem Anvarovich, Prof., Dept. of Technical Cybernetics. Dipl. electronics engineer (UGATU, 1982). Dr. of Tech. Sci. in systems analysis, management and information processing (UGATU, 2003).

SHAKHMAMETYEV, Timur Rashitovich, Junior Researcher, Postgrad. (PhD) Student, Dept. of Technical Cybernetics. Dipl. Engineer (UGATU, 2009).

MOSKVICHEV, Sergey Sergeyevich, Junior Researcher, Postgrad. (PhD) Student, Dept. of Technical Cybernetics. Master of Technics & Technology (UGATU, 2008).

KHAMADEEV, Ilgiz Khanifovich, Undergrad. Student, Dept. of Technical Cybernetics.