

УДК 004.85

## Адаптивное управление роботами с модульной конструкцией

Демин А. В.

**Постановка задачи:** задача разработки систем управления для модульных роботов сталкивается с серьезными трудностями, являющимися следствием гиперизбыточности подобных систем. Существующие подходы на основе моделей обучения с подкреплением не эффективны для систем с большим числом степеней свободы. А эволюционные методы имеют серьезные ограничения, связанные с необходимостью наличия популяции роботов, что не позволяет проводить обучение и адаптацию в режиме реальной работы. **Целью работы** является разработка методов управления гиперизбыточными модульными системами, позволяющих осуществлять обучение и адаптацию подобных систем в режиме реального времени. Для решения данной задачи предлагается использовать логико-вероятностные методы извлечения знаний из данных, основанные на идеях семантического вероятностного вывода и адаптированные для задач управления. **Результаты:** в данной работе предложен новый подход к созданию обучающихся систем управления для модульных роботов, основанный на использовании свойств функциональной схожести модулей и логико-вероятностного алгоритма направленного поиска правил. Подход основан на совместном обучении управляющих модулей, начиная с поиска общих для всех модулей управляющих правил и закаливая их последующей спецификацией в соответствии с идеями вероятностного логического вывода. Основными преимуществами предложенного подхода является высокая скорость обучения и возможность обучения в режиме реальной работы, основываясь только на опыте взаимодействия системы с окружающей средой. Приведены примеры построения и обучения систем управления для трех виртуальных моделей роботов: мзеевидного, многоногого и хоботовидного манипулятора. Проведенные эксперименты подтвердили высокую скорость обучения и качество управления. **Практическая значимость:** с практической точки зрения, результаты экспериментов показывают, что предложенный подход к адаптивному управлению может быть использован в задачах разработки систем управления интеллектуальными агентами – программными или робототехническими системами, в том числе гиперизбыточными, от которых требуется возможность обучения и приспособления к изменяющимся обстоятельствам.

**Ключевые слова:** система управления, обнаружение закономерностей, извлечение знаний.

### Введение

В настоящее время активно развивается новое направление в робототехнике под названием «модульные роботы» [1, 2]. Основная идея данного подхода – конструирование роботов из множества простых однотипных модулей, которые сами по себе обладают небольшой подвижностью, но, соединяясь друг с другом, способны образовывать сложные механические системы с большим числом степеней свободы.

Подобные роботы обладают целым рядом интересных возможностей, превосходящих способности традиционных роботов. Во-первых, это возможность создания различных конструкций из одних и тех же модулей, что позволяет решать различные задачи, используя один и тот же набор модулей. Это намного выгоднее и удобнее, чем создавать множество специализированных роботов под каждую конкретную задачу. Более того, появляется возможность создания роботов-трансформеров, самостоятельно меняющих конструкцию в зависимости от задач и условий окружающей среды. Во-вторых, модульная структура и наличие большого количества степеней свободы (гиперизбыточность) позволяет создавать отказоустойчивые модели.

Выход из строя отдельных модулей такого робота не является критичным для работы всей системы и вызывает минимальное ухудшение рабочих характеристик. В-третьих, производство и использование подобных роботов экономически выгодно, поскольку однотипные модули проще и дешевле в изготовлении и ремонте.

Однако широкие возможности модульных роботов, связанные с гиперизбыточностью, имеют и обратную сторону – значительную сложность в управлении. В частности, актуальной задачей является создание системы управления движением для уже заданной конфигурации робота. В то время как для обычных роботов традиционным подходом к созданию систем управления является ручное программирование человеком-разработчиком, для модульных роботов подобный подход оказывается малоэффективным. Из-за наличия большого количества степеней свободы разработчику крайне трудно заранее предусмотреть и запрограммировать все возможные формы двигательной активности и ситуации, когда их необходимо применять, и в особенности – учесть возможность адаптации в случае поломки отдельных модулей или неожиданного изменения окружающей среды. Поэтому становится актуальной разработка способов автоматического порождения системы управления на основе различных моделей обучения.

Однако использование популярных методов, таких как обучение с подкреплением (Reinforcement Learning), напрямую для генерации систем управления гиперизбыточными роботами оказывается затруднительным по причине наличия большого количества степеней свободы у подобных роботов. Поэтому в настоящее время многие разработчики в основном отдают предпочтение эволюционным методам, в частности, генетическим алгоритмам и генетическому программированию, а также их комбинациям со стандартными методами обучения [3–9].

Но применение эволюционных методов также имеет недостатки, основными из которых являются следующие [10]. Во-первых, это существенное время, требуемое на проведение вычислений, поскольку на каждом эволюционном шаге каждое решение из популяции требует оценки качества управления, для чего необходимо каждое решение запустить на выполнение. Во-вторых, это практическая невозможность применения данного метода для адаптации в условиях реальной работы, поскольку для работы метода необходимо наличие популяции роботов.

В данной работе предлагается обучающаяся система управления, использующая логико-вероятностный метод извлечения знаний для генерации правил управления из опыта взаимодействия системы с окружающей средой [11–15]. Особенность предлагаемого подхода состоит в том, что система в первую очередь пытается обнаружить управляющие правила, общие для всех модулей, и только затем правила, специфичные для каждого конкретного модуля в отдельности. Эффективность подхода предлагается оценить на примере обучения типичных представителей простейших гиперизбыточных модульных роботов: змеевидного робота, многоногого робота и хоботовидного манипулятора.

## Симулятор

Для проведения экспериментов с предложенной моделью управления был разработан интерактивный 3D-симулятор с графическим интерфейсом. Основное назначение программы – проведение экспериментов по управлению роботами в среде, приближенной к реальному миру. Программа обладает возможностями визуализации виртуальной среды и записью экспериментов в видео-файл. В качестве физического движка в симуляторе используется библиотека Open Dynamic Library (ODE) [16], которая позволяет моделировать динамику твердых тел с различными видами сочленений. Преимуществом данной библиотеки является скорость, высокая стабильность интегрирования, а также встроенное обнаружение столкновений. При помощи данного симулятора было построено три модели роботов: змеевидный, многоногий и хоботовидный.

Модель змеевидного робота представлена в симуляторе в виде совокупности шести одинаковых прямоугольных блоков («позвонков»), соединенных вместе при помощи универсальных сочленений (рис. 1). Все сочленения идентичны и обладают двумя угловыми двигателями («мускулами»), обеспечивающими вращение суставов в вертикальной и горизонтальной плоскостях. Предложенная конструкция, несмотря на простоту, обеспечивает достаточную гибкость модели и позволяет принимать характерные для биологических змей положения тела.



Рис. 1. Модель змеевидного робота

Вторая модель – многоногий робот представлена в виде конструкции из шести одинаковых модулей, связанных друг с другом жесткими сочленениями (рис. 2). Каждый модуль имеет пару Г-образных ног с правой и левой стороны соответственно. Таким образом, суммарно робот имеет двенадцать ног-конечностей. Каждая нога соединена с модулем при помощи универсального сочленения, имеющего два угловых двигателя, которые позволяют вращать ногу в суставе в горизонтальной и вертикальной плоскостях. В целом, конструкция робота напоминает своим видом биологических многоножек и позволяет реализовать характерные для данного вида способы передвижения.

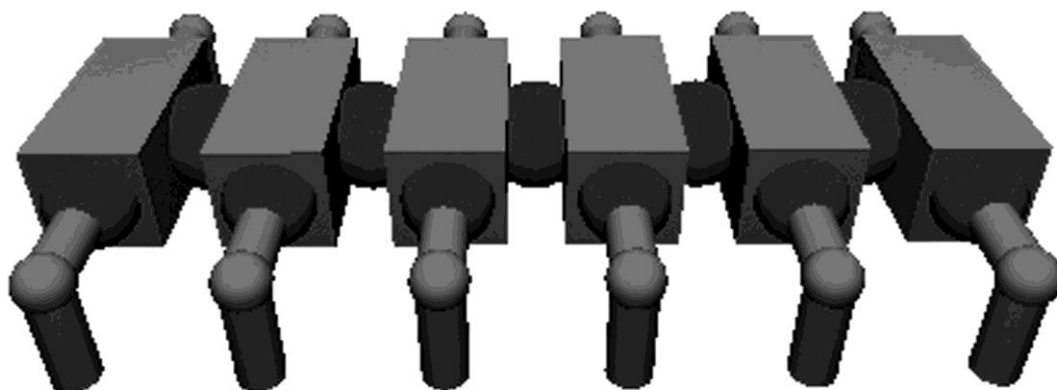


Рис. 2. Модель многоногого робота



Рис. 3. Модель хоботовидного манипулятора

Модель хоботовидного манипулятора представлена в виде многосекционного «хобота», соединенного универсальным сочленением с массивной неподвижной платформой, реализованной в виде куба (рис. 3). Сам хобот представлен как последовательность пяти одинаковых прямоугольных блоков, связанных универсальными сочленениями с угловыми двигателями. Размеры блоков и позиции сочленений были подобраны таким образом, чтобы обеспечить достаточную для проведения экспериментов гибкость системы и область достижимости.

### Система управления

Для создания системы управления модульными роботами предлагается использовать нейронные сети, состоящие из обучаемых логических нейронов, каждый из которых управляет отдельным модулем робота.

Логические нейроны функционируют в дискретном времени  $t = 0, 1, 2, \dots$ . Каждый нейрон содержит некоторый набор входов  $input_1, \dots, input_k$ , принимающих действительные значения, и один выход  $output$ , принимающий значения из заранее заданного набора  $\{y_1, \dots, y_m\}$ . В каждый момент времени  $t$  на входы нейрона подается входящая информация путем присвоения входам некоторых действительных значений  $input_1 = x_1, \dots, input_k = x_k$ ,  $x_1, \dots, x_k \in \mathbf{R}$ . Результатом работы нейрона является выходной сигнал  $output = y$ ,  $y \in \{y_1, \dots, y_m\}$ , принимающий одно из возможных значений  $\{y_1, \dots, y_m\}$ .

После того, как отработают все нейроны сети, от внешней среды поступает награда. Функция награды задается в зависимости от конечной цели и служит оценкой качества управления. Задачей системы управления является обнаружение таких закономерностей функционирования нейронов, которые бы обеспечивали получение максимальной награды.

Множество закономерностей, определяющих работу нейронов, предлагается искать в виде логических закономерностей с оценками, имеющих следующий вид:

$$\forall i(P(i), X_1(i), \dots, X_m(i), Y(i) \rightarrow r) \quad (1)$$

где  $i = 1, \dots, n$  – переменная по объектам - индексам нейронов.

$X_j(i) \in \mathbf{X}$  – предикаты из заданного множества входных предикатов  $\mathbf{X}$ , описывающих входы  $j$  нейронов  $N_i$  ( $i = 1, \dots, n$ ). К примеру, в простейшем случае данные предикаты могут быть заданы как  $X_j(i) = (input_k(i) = x_r)$ , где  $x_r$  – некоторые константы из области значений входящих сигналов, которые могут быть заданы, к примеру, путем квантования диапазона возможных значений соответствующих входов нейронов.

$Y_j(i) \in \mathbf{Y}$  – предикаты из заданного множества выходных предикатов  $\mathbf{Y}$ , описывающих выходы нейронов  $N_i$  ( $i = 1, \dots, n$ ) и имеющих вид  $Y_j(i) = (output(i) = y_r)$ , где  $y_r$  – некоторые константы из набора значений выходных сигналов.

$P(i) \in \mathbf{P}$  – предикаты из множества предикатов  $\mathbf{P}$ , имеющих вид  $(i = j)$ , где  $j = 1, \dots, n$ , смысл которых – сужать область применения правил вида (1) до конкретных нейронов.

$r$  – награда, максимизация которой является постоянной задачей нейрона.

Данные закономерности предсказывают, что если на вход нейрона  $N_i$ ,  $i = 1, \dots, n$  будут поданы сигналы, удовлетворяющие входным предикатам  $X_1(i), \dots, X_m(i)$  из посылки правила, и нейрон подаст на свой выход сигнал, указанный в выходном предикате  $Y(i)$ , то математическое ожидание награды будет равно некоторой величине  $r$ .

Отдельно отметим, что если какой-либо нейрон  $N_j$  имеет вход, специфичный только для этого нейрона, то предполагаем, что предикат  $X(i)$ ,



описывающий этот вход, будет принимать значение «0» для всех  $i \neq j$ , т.е. для всех других нейронов. Аналогично, если выход какого-либо нейрона  $N_j$  может принимать некоторое значение  $y$ , характерное только для этого нейрона, то соответствующий выходной предикат ( $output(i) = y$ ) также будет принимать значение «0» для всех  $i \neq j$ .

Поясним необходимость введения множества предикатов  $P$ . В том случае, если правила (1) не содержат предикатов из  $P$ , то они будут иметь вид  $\forall i(X_1(i), \dots, X_m(i), Y(i) \rightarrow r)$  и будут описывать закономерности, общие для всех нейронов  $N_i$ ,  $i = 1, \dots, n$ . Добавление в посылку правила предиката из  $P$  автоматически суживает область применения правила до конкретного нейрона. Таким образом, правила, содержащие предикаты из  $P$ , описывают закономерности, специфичные для конкретных нейронов. Также следует отметить, что сужение области применимости правил (1) может происходить не только за счет предикатов из  $P$ , но также за счет входных либо выходных предикатов из  $X$  и  $Y$ , описывающих специфичные входы либо выходы конкретных нейронов.

Для нахождения закономерностей вида (1) предлагается использовать алгоритм, основанный на идеях семантического вероятностного вывода, описанного в работах [13, 17]. При помощи данного алгоритма анализируется множества данных, хранящих статистику работы нейронной сети (вход-выход нейронов и полученная награда) и извлекаются все статистически значимые закономерности вида (1).

Рассмотрим алгоритм поиска закономерностей подробнее.

Для простоты в дальнейшем будем записывать правила (1) в упрощенном виде:

$$P, X_1, \dots, X_m, Y \rightarrow r. \quad (2)$$

Введем ряд формальных определений.

Перепишем правило (2) в виде  $A, Y \rightarrow r$ , где  $A$  обозначает множество предикатов из множеств  $P$  или  $X$ , входящих в посылку правила, т.е.  $A = \{P, X_1, \dots, X_m\}$ .

**Определение 1.** Подправилом правила  $R_1 = A_1, Y \rightarrow r$ ,  $A_1 \neq \emptyset$  будем называть любое правило  $R_2 = A_2, Y \rightarrow r$ , для которого выполнено условие  $A_2 \subset A_1$ .

**Определение 2.** Закономерностью будем называть правило вида (2), удовлетворяющее следующим условиям:

1. Математическое ожидание награды  $r$  для правила определено.
2. Математическое ожидание награды  $r$  правила строго больше математических ожиданий награды для каждого из его подправил.

**Определение 3.** Правило  $R_2 = A_2, Y \rightarrow r$  будем называть уточнением правила  $R_1 = A_1, Y \rightarrow r$ , если для него выполняется одно из условий

- 1)  $A_2 = A_1 \cup X$ , где  $X \in X$  и  $X \notin A_1$ , либо

2)  $A_2 = A_1 \cup P$ , где  $P \in P$  и  $A_1$  содержит только предикаты из  $X$ .

Т. е.  $R_2$  будет являться уточнением  $R_1$ , если оно получено либо добавлением в посылку  $R_1$  нового входящего предиката из  $X$ , либо добавлением любого предиката из  $P$ , если в  $R_1$  нет предикатов данного вида. Таким образом, суть операции уточнения состоит в конкретизации области применения правила либо путем добавления новых входных признаков, либо путем сужения применимости правила до конкретного нейрона.

Суть алгоритма обнаружения закономерностей заключается в последовательном уточнении правил, начиная с правил единичной длины, путем добавления в посылку правил новых предикатов с последующей проверкой уточненных правил на принадлежность к вероятностным закономерностям. По существу, реализуется направленный перебор правил, позволяющий существенно сократить пространство поиска. Сокращение перебора достигается за счет использования эвристики, которая заключается в том, что, начиная с момента, когда длина посылки правил достигает некоторой заданной величины, называемой глубиной базового перебора, начинается последовательное уточнение только тех правил, которые являются вероятностными закономерностями.

Перейдем к описанию алгоритма, реализующего поиск множества закономерностей, определяющих работу нейронов. Обозначим через  $Spec(RUL)$  – множество всех возможных уточнений всех правил из  $RUL$ , где  $RUL$  – произвольное множество правил вида (2). Входным параметром алгоритма также является глубина базового перебора  $d$ , где  $d \geq 1$  – натуральное число.

1. На первом шаге генерируем множество  $RUL_1$  всех правил единичной длины, имеющих следующий вид  $Y \rightarrow r$ ,  $Y \in \{Y_1, \dots, Y_k\}$ . Все правила  $RUL_1$  проходят проверку на выполнение условий принадлежности к закономерностям. Правила, прошедшие проверку, будут являться закономерностями. Обозначим через  $REG_1$  множество всех закономерностей, обнаруженных на первом шаге.

2. На шаге  $k \leq d$  генерируется множество  $RUL_k = Spec(RUL_{k-1})$  всех уточнений правил, сгенерированных на предыдущем шаге. Все правила из  $RUL_k$  проходят проверку на выполнение условий принадлежности к закономерностям. Обозначим через  $REG_k$  полученное множество закономерностей.

3. На шаге  $l > d$  генерируется множество  $RUL_l = Spec(REG_{l-1})$  уточнений всех закономерностей, обнаруженных на предыдущем шаге. Все правила из  $RUL_l$  проходят проверку на выполнение условий принадлежности к закономерностям. Обозначим  $REG_l$  – множество всех закономерностей, обнаруженных на данном шаге.

4. Алгоритм останавливается на шаге  $m > d$ , когда не обнаружено новых закономерностей  $REG_m = \emptyset$ .

5. Результирующее множество закономерностей является объединение всех множеств обнаруженных закономерностей  $REG_i$ .

Шаги алгоритма  $k \leq d$  соответствуют базовому перебору, а шаги  $k > d$  – дополнительному перебору.

Оценка математического ожидания награды для правил осуществляется по множеству данных, хранящих статистику работы системы (вход-выход нейронов и полученная награда), следующим образом:

$$r = \frac{1}{|I|} \sum_{i \in I} r_i,$$

где  $I$  – множество событий, когда правило может быть применено,  $r_i$  – награда нейрона для  $i$ -го события,  $i \in I$ .

Преимущество использования семантического вероятностного вывода и правил вида (1) состоит в организации поиска правил таким образом, что сначала будут обнаруживаться правила, общие для всех нейронов, а только затем – более сложные, включающие специфичные для конкретных нейронов правила. В результате, в задачах управления модульными роботами, если хотя бы часть модулей имеет схожие функции, которые можно описать общими правилами, предложенный подход позволяет существенно сократить время поиска решения.

Функционирование нейронной сети в целом происходит следующим образом. На каждом такте работы сети на входы нейронов поступают входящие сигналы. После чего последовательно для каждого нейрона запускается процедура принятия решения, в процессе которой из множества правил, описывающих работу нейронов, выбираются те, которые применимы к текущему нейрону на текущих входных сигналах. Затем среди отобранных правил выбирается одно правило, прогнозирующее максимальное значение математического ожидания награды  $r$ . Далее на выход нейрона подается выходной сигнал  $output = y$ , указанный в правиле. В начальной стадии функционирования сети, когда множество правил, описывающих работу нейронов, еще пусто, либо когда нет правил, применимых к текущему набору входящих сигналов, выход нейрона определяется случайным образом. После того, как все нейроны сгенерируют свои выходные сигналы, от внешней среды поступает награда и осуществляется обучение, в процессе которого ищутся новые и корректируются текущие правила работы в соответствии с предложенным алгоритмом поиска закономерностей.

### Система управления движением змеевидного робота

Целью данного эксперимента являлось обучение способам передвижения вперед простейшей модели змеевидного робота (рис. 1). В предыдущих работах [18, 19] нами была предложена модель нейронного контура управления локомоцией нематоды *C. Elegans*, которая показала высокую эффективность в экспериментах по обучению волнообразному способу передвижения. Схема данного контура предполагала, что головная часть нематоды выступает в качестве источника колебаний, основываясь только на обратной связи от



рецептора растяжения. Далее сигнал распространяется по телу нематоды с некоторой временной задержкой, обеспечивая тем самым характерное волнообразное движение. Поскольку конструкция змеевидного робота имеет много общих черт с моделью нематоды, то в данной работе было решено использовать похожую схему нейронного контура для управления движением робота.

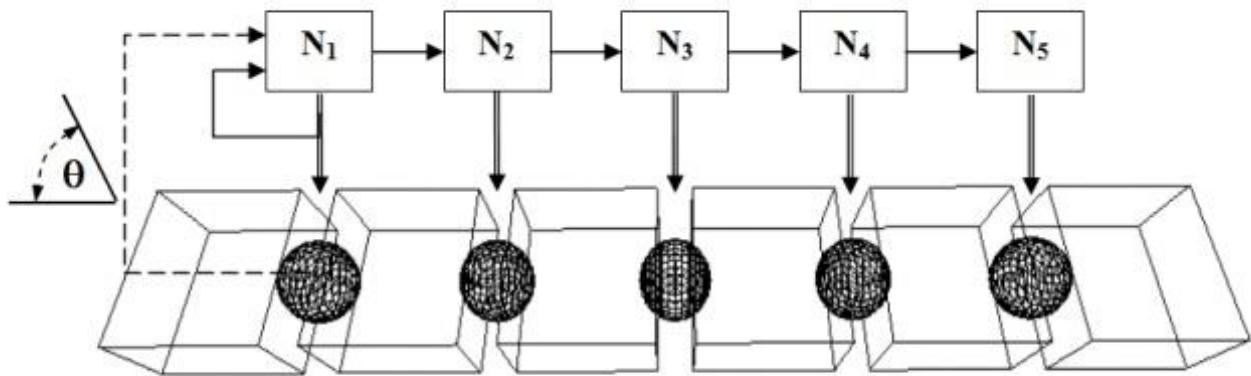


Рис. 4. Схема нейронного контура управления движением змеевидного робота

В результате был выбран нейронный контур, состоящий из 5 нейронов (рис. 4). Каждый нейрон  $N_i$ ,  $i=1, \dots, 5$  контролирует один сустав тела робота, подавая активирующие сигналы на угловые двигатели, расположенные в этом суставе. Головной нейрон  $N_1$  получает на вход информацию об углах сгиба между головным и последующим сегментом. Помимо этого на вход нейрона по обратным связям поступает сигнал от его собственного выхода с временной задержкой  $\Delta t$ . Остальные нейроны  $N_i$ ,  $i=2, \dots, 5$  получают на свой вход только сигнал от выхода предыдущего нейрона  $N_{i-1}$  с временной задержкой  $\Delta t$ .

Множество входных и выходных предикатов для нейронов задается путем квантования диапазона возможных значений соответствующих входов и выходов нейрона. Награда для всего нейронного контура управления движением определяется в зависимости от величины скорости, которую разовьет робот на отрезке времени  $\Delta t$ : чем выше скорость – тем больше награда.

Используя 3D-симулятор, был проведен ряд успешных экспериментов по обучению предложенной модели способам передвижения. Как показали результаты экспериментов, системе управления удастся стабильно обучаться эффективному способу передвижения вперед, основанному на волнообразном движении туловища в горизонтальной плоскости. Данный способ передвижения является самым распространенным среди биологических змей, а также характерен и для некоторых других животных, к примеру, нематод. На рис. 5 приведены найденные системой в ходе обучения оптимальные последовательности движений при перемещении вперед.

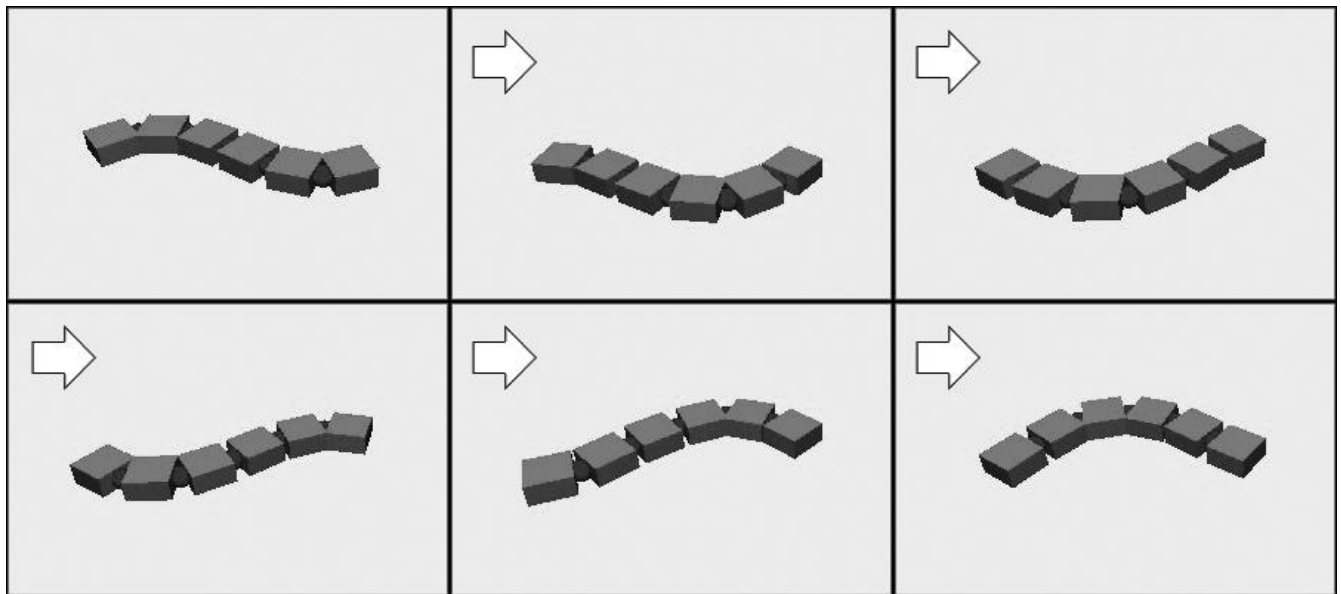


Рис. 5. Последовательность движений змеевидного робота при перемещении вперед

### Система управления движением многоногого робота

В данном эксперименте была поставлена задача обучить способу движения вперед модель многоногого робота (рис. 2). Для системы управления роботом был выбран нейронный контур, состоящий из шести нейронов – по одному нейрону на каждый модуль робота (рис. 6). Каждый нейрон  $N_i$ ,  $i = 1, \dots, 6$  контролирует движения левой и правой ноги своего модуля, подавая активирующие сигналы на соответствующие угловые двигатели, вращающие конечности в суставе. Чтобы немного упростить задачу, движения правой и левой ног робота были синхронизированы таким образом, что движение одной ноги всегда происходит в противофазе с другой. Т.е., к примеру, движение левой ноги вперед всегда сопровождается движением правой ноги назад. Таким образом, нейрону, по сути, достаточно контролировать движения только одной ноги, поскольку вторая нога будет повторять эти же движения, только в противофазе.

Нейрон первого модуля  $N_1$  получает на вход информацию о положении ног первого модуля. Остальные нейроны  $N_i$ ,  $i = 2, \dots, 6$  получают на свои вход данные о положении ног предыдущего модуля. Информация о положении ноги задается парой углов сгиба конечности в суставе в вертикальной и горизонтальной плоскостях. Как и в предыдущей задаче множество входных и выходных предикатов для нейронов задается путем квантования диапазона возможных значений их входов и выходов. Аналогично, награда для всей системы управления движением определяется в зависимости от величины скорости, которую разовьет робот на отрезке времени  $\Delta t$ : чем выше скорость – тем больше награда.

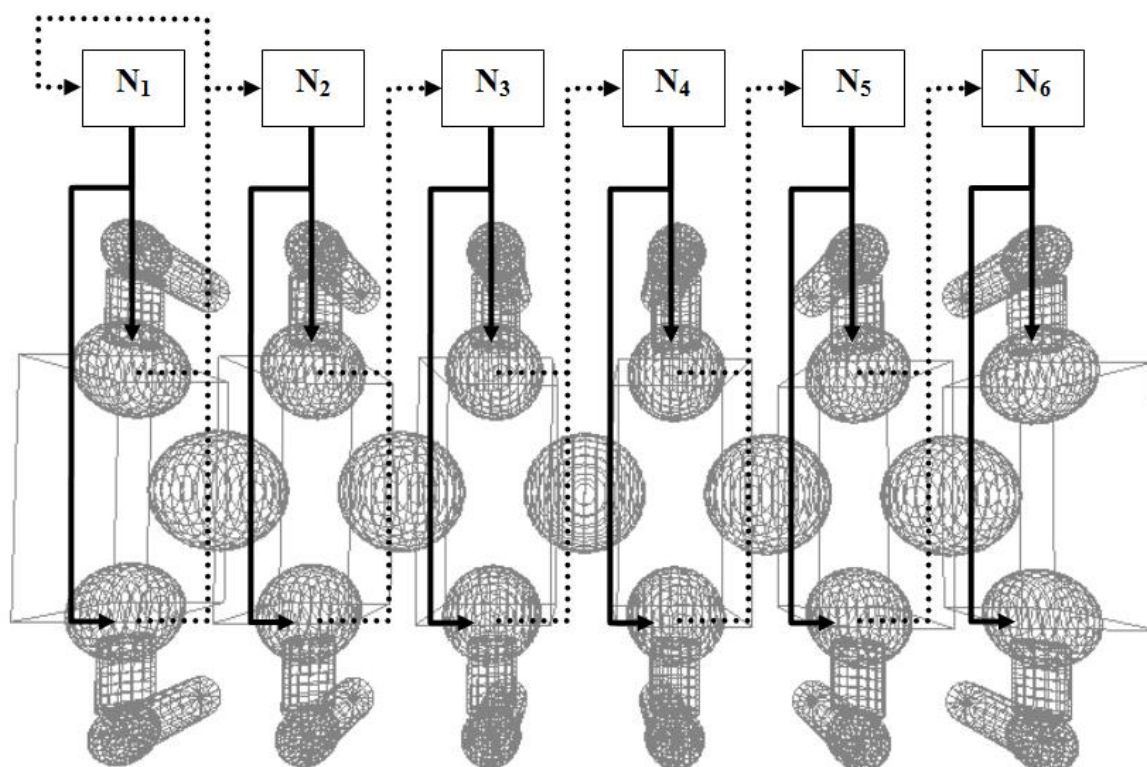


Рис. 6. Схема нейронного контура управления движением многоногого робота

При помощи 3D-симулятора была проведена серия экспериментов по обучению движению модели многоногого робота. Результаты экспериментов показали, что система управления успешно обнаруживает согласованные движения конечностей, обеспечивающие эффективное перемещение вперед. На рисунке 7 приведен пример оптимальной последовательности движений, найденной в процессе обучения.

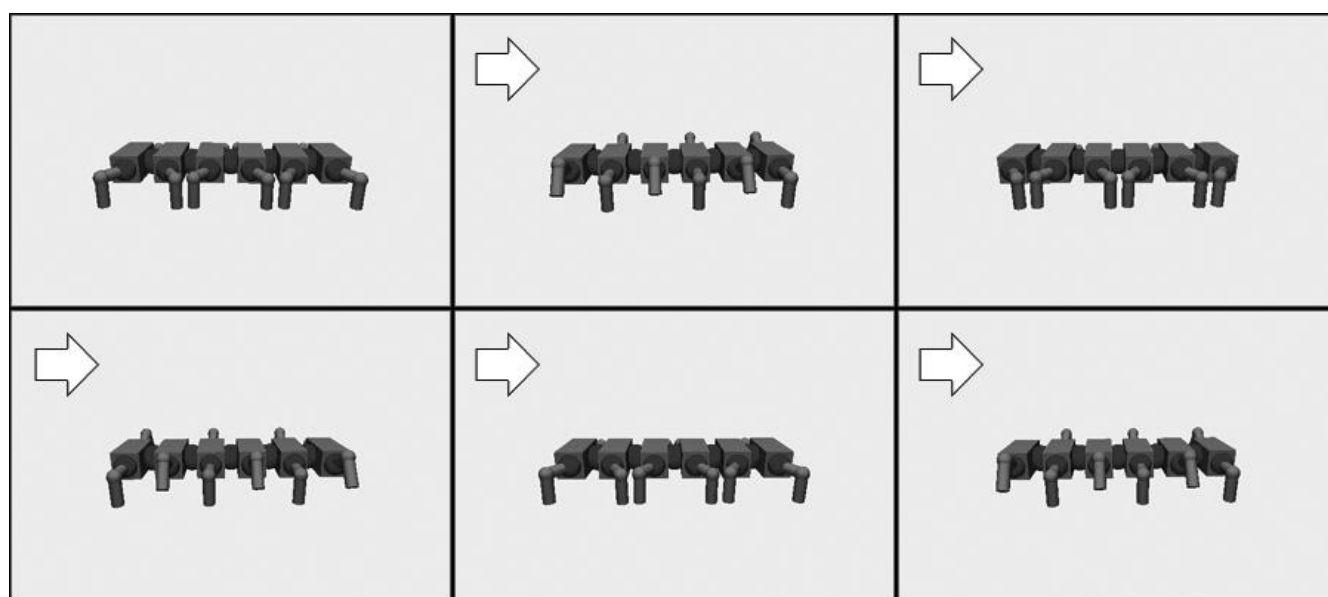


Рис. 7. Последовательность движений многоногого робота при перемещении вперед

### Система управления хоботовидным манипулятором

В данном эксперименте перед манипулятором (рис. 3) была поставлена задача захватить цель, которая появляется в случайной позиции в пределах области достижимости хобота. В качестве цели выступает сфера радиусом равным длине одного сегмента хобота. Цель считается захваченной, если конец хобота (последний сегмент) окажется внутри сферы. После захвата сфера-цель исчезает и появляется в новой случайной позиции. Таким образом, эксперимент может продолжаться непрерывно неограниченное время. Задачей системы управления является обнаружение таких правил управления движением хобота, которые бы обеспечивали захват цели в любой доступной для манипулятора позиции.

Для решения данной задачи была выбрана схема управления из пяти нейронов  $N_i$ ,  $i = 1, \dots, 5$ , каждый из которых контролирует один сегмент хобота (рис. 8). Нейроны подают активирующие сигналы на угловые двигатели, вызывая тем самым изгиб хобота в соответствующих сочленениях. Поскольку в данной задаче цель может располагаться только на одном уровне, то все движения ограничены только горизонтальной плоскостью, что упрощает задачу.

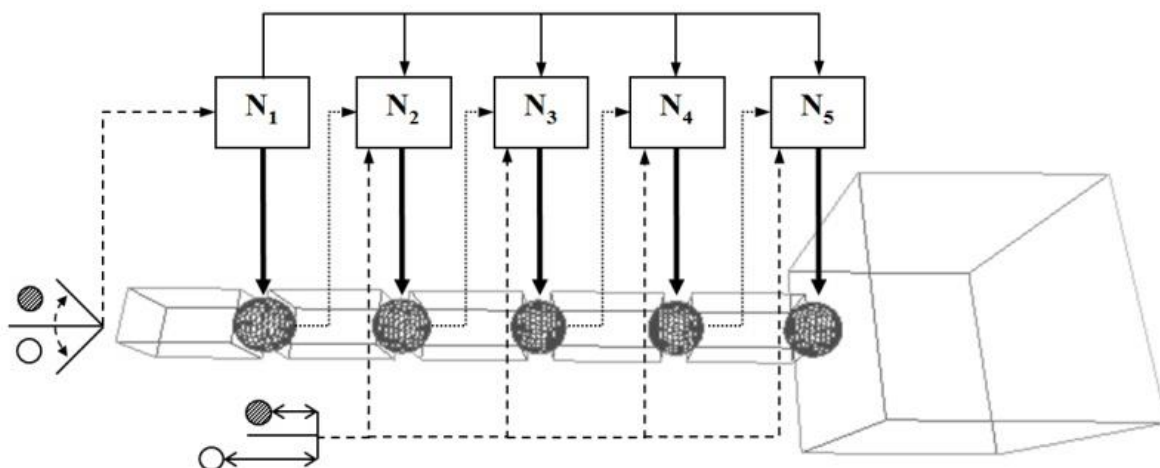


Рис. 8. Схема нейронного контура управления хоботовидным манипулятором

Первый нейрон  $N_1$  получает на свой вход бинарную информацию о том, по какую сторону относительно хобота располагается цель: справа или слева. Остальные нейроны  $N_i$ ,  $i = 2, \dots, 5$  получают на свой вход следующие сигналы:

- 1) информацию об угле сгиба между контролируемым и предыдущим сегментом;
- 2) сигнал от выхода предыдущего нейрона  $N_{i-1}$  в предшествующий момент времени;
- 3) бинарную информацию о положении цели по отношению к концу хобота и точки его крепления – результат сравнения расстояний от крепления до цели и от крепления до конца манипулятора.

Таким образом, в данном эксперименте манипулятор фактически «слеп», т. е. не видит точное положение цели, а только «ощущает» ее: справа-слева и ближе-дальше.

Награда для системы управления рассчитывается по факту захвата цели манипулятором следующим образом. Пусть цель появилась в новой позиции в момент времени  $t_0$ , а манипулятор захватил ее в момент времени  $t_1$ . Тогда все действия с момента времени  $t_0$  по момент времени  $t_1$  получают награду в размере  $r = 1 / (1 + (t_1 - t_0))$ , где  $t$  – момент времени, для которого рассчитывается награда. Таким образом, наибольшую награду получают самые последние действия, приведшие к достижению цели, и чем глубже в прошлом от момента захвата произошло действие, тем меньшую награду оно получает.

При помощи 3D-симулятора были проведены эксперименты по обучению системы управления манипулятором захвату целей в любой позиции в пределах зоны достижимости. Как показали результаты экспериментов, системе управления удастся обнаружить эффективные правила, обеспечивающие стопроцентный захват целей в указанной зоне. На рисунке 9 приведены примеры траекторий движения уже обученного хобота при захвате цели.

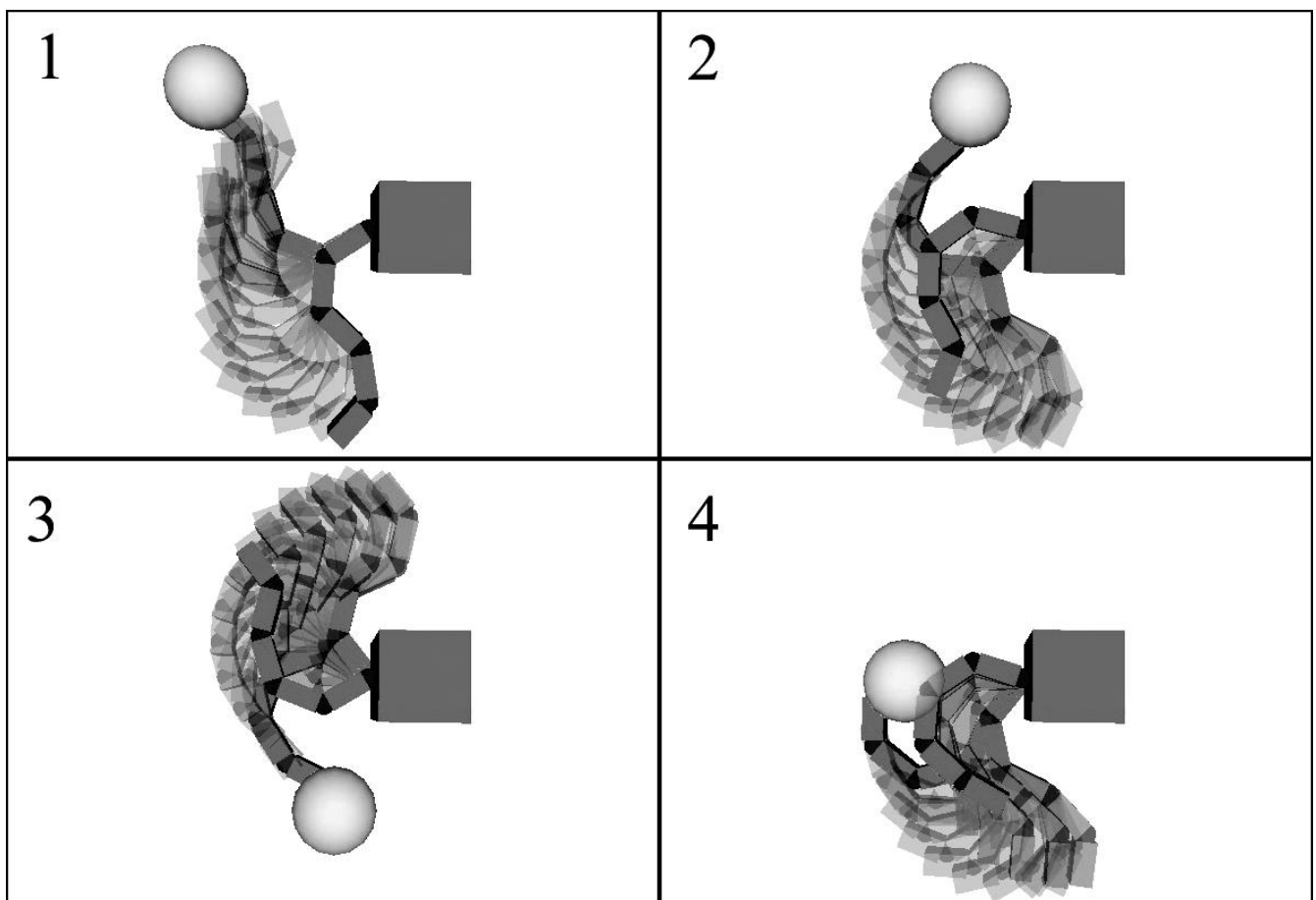


Рис. 9. Примеры траекторий движения манипулятора при захвате цели

Примеры со второго по четвертый показывают, что система управления научилась делать специальные подготовительные движения, чтобы выйти на захват. В указанных примерах стартовый изгиб хобота направлен в сторону,



противоположную цели, и для того, чтобы достичь цель, манипулятор делает волнообразное движение в другом направлении, в результате которого меняет изгиб тела на противоположное и затем успешно захватывает цель. Приведенные примеры являются демонстрацией того, что обнаруженные системой правила управления порождают достаточно сложное поведение, направленное на достижение цели.

### Заключение

В данной работе предложен подход к адаптивному управлению модульными системами с большим числом степеней свободы, основанный на совместном обучении управляющих модулей, начиная с нахождения общих для всех модулей управляющих правил и записывая их последующей спецификацией в соответствии с идеями семантического вероятностного вывода. Основными преимуществами предложенного подхода являются способность обучения в режиме реальной работы, основываясь только на опыте взаимодействия с окружающей средой, и высокая скорость обучения, которая достигается за счет эффективного использования свойств функциональной схожести модулей и алгоритма направленного поиска правил. Кроме того, предложенный подход достаточно хорошо масштабируется относительно увеличения числа модулей. В частности, добавление новых сегментов к конструкциям роботов в проведенных экспериментах слабо влияет на эффективность обучения, поскольку не изменяет количество общих для модулей правил. Однако следует отметить, что эффективность предложенного подхода во многом зависит от количества схожих по функциям модулей в схеме робота. С уменьшением доли схожих модулей преимущества от использования общих правил теряются. С практической точки зрения, проведенные экспериментальные исследования на примере моделей змеевидного и многоногого роботов, а также хоботовидного манипулятора показали, что предложенный подход является достаточно эффективным и может быть использован для управления сложными модульными системами, имеющими множество степеней свободы.

Работа выполнена при финансовой поддержке гранта РФФИ № 14-07-00386.

### Литература

1. Yim M. H., Duff D. G., Roufas K. D. Modular reconfigurable robots, an approach to urban search and rescue // 1st International Workshop on Human Welfare Robotics Systems (HWRS-2000). – 2000. – pp. 19-20.
2. Stoy K., Brandt D., Christensen D. J. Self-Reconfigurable robots: an introduction // Intelligent robotics and autonomous agents series. – MIT Press, 2010. – 216 p.
3. Bongard J. C. Evolutionary Robotics // Communications of the ACM. 2013. Vol. 56. № 8. pp. 74-83.

4. Kamimura A., Kurokawa H., Yoshida E., Tomita K., Murata S., Kokaji S. Automatic locomotion pattern generation for modular robots // Proceedings of 2003 IEEE International Conference on Robotics and Automation. 2003. pp. 714-720.
5. Ito K., Matsuno F. Control of hyper-redundant robot using QDSEGA // Proceedings of the 41st SICE Annual Conference (2002). 2002. Vol. 3. pp. 1499-1504.
6. Marbach D., Ijspeert A. J. Co-evolution of configuration and control for homogenous modular robots // Proceedings of the eighth conference on Intelligent Autonomous Systems (IAS 8). – IOS Press, 2004. – pp. 712-719.
7. Gong D., Yan J., Zuo G. A Review of Gait Optimization Based on Evolutionary Computation // Applied Computational Intelligence and Soft Computing. 2010. Vol. 2010. 12 p. – Article ID 413179.
8. Valsalam V. K. Miikkulainen R. Modular neuroevolution for multilegged locomotion // In Proceedings of GECCO. 2008. pp. 265–272.
9. Tanev I., Ray T., Buller A. Automated Evolutionary Design, Robustness and Adaptation of Sidewinding Locomotion of Simulated Snake-like Robot // IEEE Transactions on Robotics. 2005. Vol. 21. № 4. pp. 632-645.
10. Mataric M., Cliff D. Challenges in evolving controllers for physical robots // Robotics and Autonomous Systems. 1996. № 19(1). pp. 67–83.
11. Demin A. V. Logical Model of the Adaptive Control System Based on Functional Systems Theory // Young Scientist USA. Applied science. – Auburn, Washington, 2014. – pp. 113-118.
12. Демин А. В. Модель адаптивной системы управления и ее применение для управления движением виртуального робота // Молодой ученый. 2012. № 11 (46). С. 114-119.
13. Демин А. В., Витяев Е. Е. Логическая модель адаптивной системы управления // Нейроинформатика. 2008. Т. 3. № 1. С. 79-107.
14. Демин А. В. Обучающаяся система управления движением для 3D модели многоногого робота // Молодой ученый. 2015. №19 (99). С. 74-78.
15. Демин А. В. Обучение способам передвижения виртуальной модели змеевидного робота // Молодой ученый. 2014. № 19 (78). С. 147-150.
16. Smith R. Open Dynamics Engine [Электронный ресурс]. URL: <http://ode.org> (дата обращения 20.11.2015).
17. Витяев Е. Е. Извлечение знаний из данных. Компьютерное познание. Модели когнитивных процессов. – Новосибирск: НГУ, 2006. – 293 с.
18. Demin A. V., Vityaev E. E. Learning in a virtual model of the C. Elegans nematode for locomotion and chemotaxis // Biologically Inspired Cognitive Architectures (2014). – Elsevier, 2014. – Vol. 7. – pp. 9-14.
19. Демин А. В. Обучающаяся модель управления хемотаксисом нематоды C.Elegans // Нейроинформатика. 2013. Т. 7. № 1. С. 29-41.

### References

1. Yim M. H., Duff D. G., Roufas K. D. Modular reconfigurable robots, an approach to urban search and rescue. *1st International Workshop on Human Welfare Robotics Systems (HWRS2000)*, 2000, pp. 19-20.

2. Stoy K., Brandt D., Christensen D. J. Self-Reconfigurable robots: an introduction. *Intellegent robotics and autonomous agents series*, MIT Press, 2010, 216 p.
3. Bongard J. C. Evolutionary Robotics. *Communications of the ACM*, 2013, vol. 56, no. 8, pp. 74-83.
4. Kamimura A., Kurokawa H., Yoshida E., Tomita K., Murata S., Kokaji S. Automatic locomotion pattern generation for modular robots. *Proceedings of 2003 IEEE International Conference on Robotics and Automation*, 2003, pp. 714-720.
5. Ito K., Matsuno F. Control of hyper-redundant robot using QDSEGA. *Proceedings of the 41st SICE Annual Conference (2002)*, 2002, vol. 3, pp. 1499-1504.
6. Marbach D., Ijspeert A. J. Co-evolution of configuration and control for homogenous modular robots. *Proceedings of the eighth conference on Intelligent Autonomous Systems (IAS8)*, IOS Press, 2004, pp. 712-719.
7. Gong D., Yan J., Zuo G. A Review of Gait Optimization Based on Evolutionary Computation. *Applied Computational Intelligence and Soft Computing*, 2010, vol. 2010, Article ID 413179, 12 p.
8. Valsalam V. K. Miikkulainen R. Modular neuroevolution for multilegged locomotion. *Proceedings of GECCO*, 2008, pp. 265-272.
9. Tanev I., Ray T., Buller A. Automated Evolutionary Design, Robustness and Adaptation of Sidewinding Locomotion of Simulated Snake-like Robot. *IEEE Transactions on Robotics*, vol. 21, no. 4, August 2005, pp.632-645.
10. Mataric M., Cliff D. Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 1996, no. 19(1), pp. 67-83.
11. Demin A. V. Logical Model of the Adaptive Control System Based on Functional Systems Theory. *Young Scientist USA. Applied science*, Auburn, Washington, 2014, pp. 113-118.
12. Demin A. V. Model' adaptivnoi sistemy upravleniia i ee primeneniie dlia upravleniia dvizheniem virtual'nogo robota [Adaptive control system model and its application for virtual robot locomotion control]. *Young Scientist*, 2012, vol. 46, no. 11, pp. 114-119 (in Russian).
13. Demin A. V., Vityaev E. E. Logicheskaiia model' adaptivnoi sistemy upravleniia [Logical model of the adaptive control system]. *Neiroinformatika [Neuroinformatics]*, 2008, vol. 3, no. 1, pp. 79-107 (in Russian).
14. Demin A. V. Obuchaiushchaia sistema upravleniia dvizheniem dlia 3D modeli mnogonogogo robota [Learning locomotion control system for 3D multiped robot model]. *Young Scientist*, 2015, vol. 99. no. 19, pp. 74-78 (in Russian).
15. Demin A. V. Obuchenie sposobam peredvizheniia virtual'noi modeli zmeevidnogo robota [Locomotion patterns learning in virtual model of snake-like robot]. *Young Scientist*, 2014, vol. 78, no. 19, pp. 147-150 (in Russian).
16. Smith R. Open Dynamics Engine. Available at: <http://ode.org> (accessed 20 November 2015).
17. Vityaev E. E. *Iz vlechenie znanii iz dannykh. Komp'iuternoe poznanie. Modeli kognitivnykh protsessov* [Knowledge Discovery in Databases. Computational

Cognition. Models of Cognitive Processes]. Novosibirsk, Novosibirsk National Research State University, 2006. 293 p. (in Russian).

18. Demin A. V., Vityaev E. E. Learning in a virtual model of the C.Elegans nematode for locomotion and chemotaxis. *Biologically Inspired Cognitive Architectures* (2014), Elsevier, 2014, vol. 7, pp. 9-14.

19. Demin A. V. Learning control model of chemotaxis for C.Elegans nematode. *Neiroinformatika [Neuroinformatics]*, 2013, vol. 7, no. 1, pp. 29-41 (in Russian).

Статья поступила 26 ноября 2015 г.

### Информация об авторе

Демин Александр Викторович – кандидат физико-математический наук. Старший научный сотрудник. Институт систем информатики имени А.П. Ершова СО РАН. Область научных интересов: анализ данных, обнаружение закономерностей в базах данных, машинное обучение, адаптивное управление. Тел.: +7 (383) 330 66 60. E-mail: alexandredemin@yandex.ru

Адрес: 630090, Россия, г. Новосибирск, пр. Лаврентьева, 6.

---

## Localization Adaptive Control of Robots with Modular Structure

A. V. Demin

**Problem formulation:** the task of modular robot control system development implies serious difficulties being a consequence of such systems' hyper-redundancy. Existing approaches based on reinforcement learning models are non efficient for systems with many degrees of freedom. Evolutionary methods experience serious limitations associated based on necessity of having a population of robots, which does not allow learning and adapting these systems in real work conditions. **The purpose of this work** is developing hyper-redundant modular system control methods allowing for systems' real-time learning and adaptation. For solving this problem, we propose using logic-probabilistic methods for knowledge discovery from data based on semantic probabilistic inference approach ideas and adapted to control tasks. **Results:** In this paper we propose a new approach to creating learning modular robot control systems based on using modules' functional similarity properties and logic-probabilistic directional rule search algorithm. The approach is based on cooperative control modules' training, from discovering common control rules for all the modules to their subsequent specification in accordance with semantic probabilistic inference ideas. The main advantages of proposed approach are high learning speed and the ability to learn during real work based only on environment interaction experience. The examples of control system design and training has been presented for three virtual robots model: snake-like robot, multyped robot and trunk manipulator. The experiments have confirmed high learning speed and control quality. **Practical relevance:** From the practical point of view, the results of experiments have shown that the proposed approach for adaptive control can be used for the intelligent agent design tasks with agents being programs or robotic systems including hyper-redundant systems, which require the possibility of learning and adapting to changing environment.

Key words: control system, data mining, knowledge discovery.

### Information about Author

*Alexandr Viktorovich Demin* – Ph.D. of Physico-mathematical Sciences. Senior Research Officer. A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences. Field of research: Data Mining, Knowledge Discovery in Databases, Machine Learning, Adaptive control. Tel.: +7 (383) 330 66 60. E-mail: [alexandredemin@yandex.ru](mailto:alexandredemin@yandex.ru)

Address: Russia, 630090, Novosibirsk, Lavrentjev pr., 6.