

УДК 004.85

Логико-вероятностный метод управления модульными роботами

Демин А.В. (Институт систем информатики СО РАН)

В данной работе представлен логико-вероятностный метод адаптивного управления модульными системами, основанный на использовании свойств функциональной схожести модулей и логико-вероятностного алгоритма направленного поиска правил. Предложенный метод основан на совместном обучении управляющих модулей, начиная с поиска общих для всех модулей управляющих правил и записывая их последующей спецификацией в соответствии с идеями вероятностного логического вывода. С помощью интерактивного 3D-симулятора были проведены успешные эксперименты с четырьмя виртуальными моделями роботов. Экспериментальные исследования показали, что предложенный подход достаточно эффективный и может быть использован для управления модульными системами с большим количеством степеней свободы.

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

1. Введение

Гиперизбыточные робототехнические системы, характеризующиеся большим числом степеней свободы, обладают рядом существенных преимуществ перед традиционными системами. Типичные представители гиперизбыточных систем – это змеевидные и многоногие роботы, многозвенные манипуляторы, модульные роботы и др. Гиперизбыточность наделяет подобные системы более универсальными свойствами, позволяя им решать больший круг задач, а также повышает их отказоустойчивость. Одними из наиболее интересных и перспективных представителей гиперизбыточных систем являются так называемые «модульные роботы». Это класс робототехнических систем, конструкции которых состоят из множества простых однотипных модулей [16,19]. Данное направление робототехники активно развивается в последнее время и сулит целый ряд новых возможностей, начиная с создания роботов-трансформеров, меняющих свою конструкцию для решения конкретных задач, и заканчивая удешевлением производства за счет использования однотипных модулей.

Однако развитие и использование гиперизбыточных робототехнических систем сталкивается с серьезными трудностями, связанными со значительной сложностью управления подобными системами. Наличие большого количества степеней свободы делает невозможным применение традиционных подходов к созданию систем управления путем прямого задания сенсорно-моторных функций человеком-разработчиком. Поэтому становится актуальной разработка способов автоматического порождения системы управления на основе различных моделей обучения.

В мировой практике в области адаптивного управления гиперизбыточными и модульными системами чаще всего предлагаются решения, основанные на использовании популяционных методов (эволюционные методы, методы роя частиц и т.д.) в интеграции с другими известными методами машинного обучения (Reinforcement Learning, нейронные сети и др.) [7,8,11-13,17,18]. Однако эволюционные методы имеют серьезные ограничения, связанные с необходимостью наличия популяции роботов, что не позволяет проводить обучение и адаптацию в режиме реальной работы [14]. Общим же недостатком подобных решений является невозможность обучения в режиме реальной работы и слабая масштабируемость относительно увеличения сложности системы (количества степеней свободы). В целом, следует отметить, что в настоящее время пока еще не предложено достаточно универсального решения задачи адаптивного управления гиперизбыточными системами.

В наших работах предлагается альтернативный подход к созданию обучающихся систем управления для модульных роботов, основанный на использовании логико-вероятностных методов извлечения знаний из данных и эксплуатации свойств функциональной симметрии элементов конструкции роботов [2-5,9]. В соответствии с данным подходом управляющие правила системы описываются при помощи языка логики первого порядка, что позволяет использовать логико-вероятностные методы извлечения знаний из данных для обнаружения эффективных правил управления в статистических данных о взаимодействии системы с окружающим миром. В качестве основного пути для преодоления проблемы большого числа степеней свободы предлагается идея использования функциональной симметрии элементов системы, что позволяет существенно сократить пространство поиска управляющих правил за счет использования одних и тех же правил для схожих по своим функциям модулей. Интеграция логико-вероятностного подхода и свойств функциональной симметрии позволили разработать специальный метод поиска управляющих правил, который в первую очередь пытается найти правила, общие для всех модулей, а уже затем специфицировать их для каждого конкретного модуля в отдельности. Эффективность подхода предлагается оценить на примере обучения типичных представителей простейших гиперизбыточных

модульных роботов: змеевидного робота, многоногого робота, хоботовидного манипулятора и многоногого робота с двумя типами конечностей.

2. Система управления

Для создания системы управления модульными роботами предлагается использовать модель с сетевой структурой, в которой базовым элементом управления является обучаемый логический нейрон. Задачей каждого логического нейрона является управление 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.

$P(i) \in P$ – предикаты из множества предикатов 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 содержит 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 [1,5]. При помощи данного алгоритма анализируются 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: $r = \sum_{i \in I} r_i / |I|$, 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 составе системы управления 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, система управления запускает на выполнения все действия, которые были активированы этими сигналами. После выполнения действий 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.

3. 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) [15], 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 четыре модели роботов: змеевидный робот, многоногий робот, хоботовидный манипулятор и многоногий робот с двумя типами конечностей.

Модель змеевидного робота была реализована в симуляторе в виде совокупности шести одинаковых прямоугольных блоков («позвонков»), соединенных вместе при помощи универсальных сочленений (рис. 1а). 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.

Вторая модель – многоногий робот представлена в виде конструкции из шести одинаковых модулей, связанных друг с другом жесткими сочленениями (рис. 1b). 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 поворачивать 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.

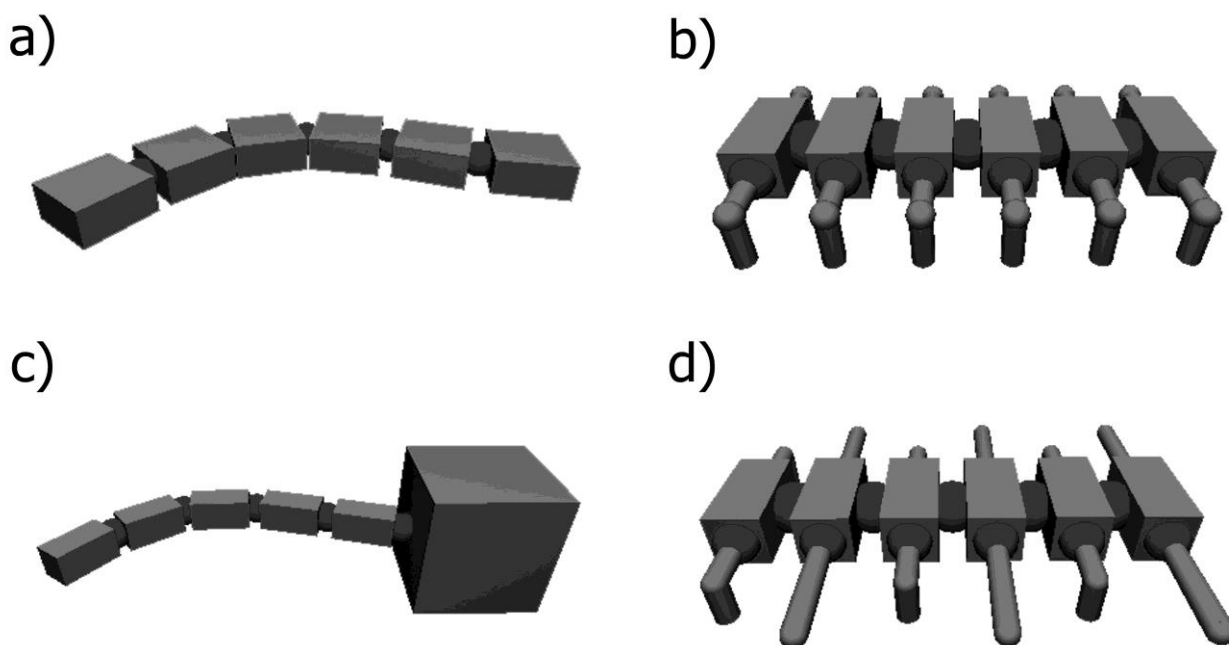


Рис.1. Модели роботов

Модель хоботовидного манипулятора представлена в виде многосекционного «хобота», соединенного универсальным сочленением с массивной неподвижной платформой, реализованной в виде куба (рис. 1c). Сам хобот реализован 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.

Последняя модель представляет собой многоного робота, состоящая из двух повторяющихся типов модулей (рис. 1d). Четные модули имеют пару Г-образных конечностей с правой и левой стороны, способные двигаться только в горизонтальной плоскости. Нечетные модули имеют пару прямых конечностей, способных двигаться только в вертикальной плоскости. Модули поочередно соединены друг с другом посредством жестких сочленений. Всего робот имеет шесть модулей: три модуля с Г-образными конечностями и три – с прямыми. Данная модель робота была разработана специально для того, чтобы проверить возможности предложенной модели успешно обнаруживать эффективные управляющие правила для различных типов модулей.

4. Система управления движением змеевидного робота

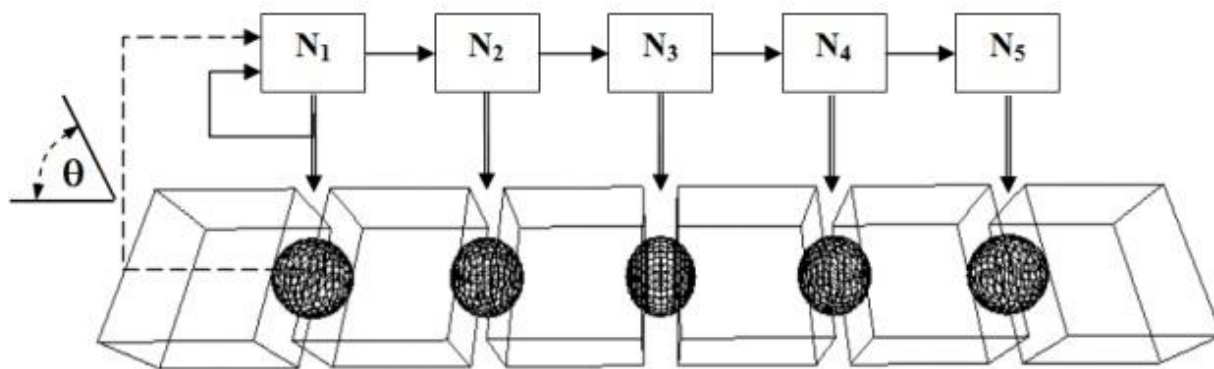


Рис.2. Схема нейронного контура управления движением змеевидного робота

Целью данного эксперимента являлось обучение способам передвижения вперед простейшей модели змеевидного робота (рис. 1a). В предыдущих работах [6,10] 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.

В результате был выбран нейронный контур, состоящий из 5 нейронов (рис. 2). 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 Δt . Остальные нейроны N_i , $i=2, \dots, 5$ получают на свой вход только сигнал от выхода предыдущего нейрона N_{i-1} с временной задержкой Δt .

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 Δt : чем выше скорость – тем больше награда.

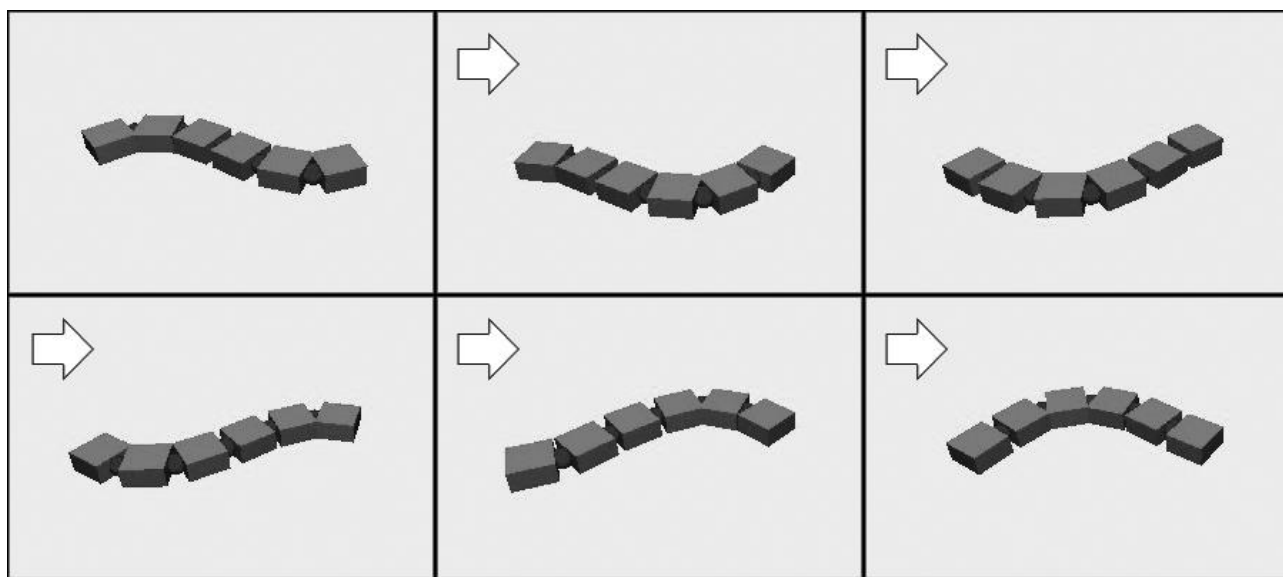


Рис.3. Последовательность движений змеевидного робота при перемещении вперед

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 рисунке 3

приведены найденные системой в ходе обучения оптимальные последовательности движений при перемещении вперед.

5. Система управления движением многоногого робота

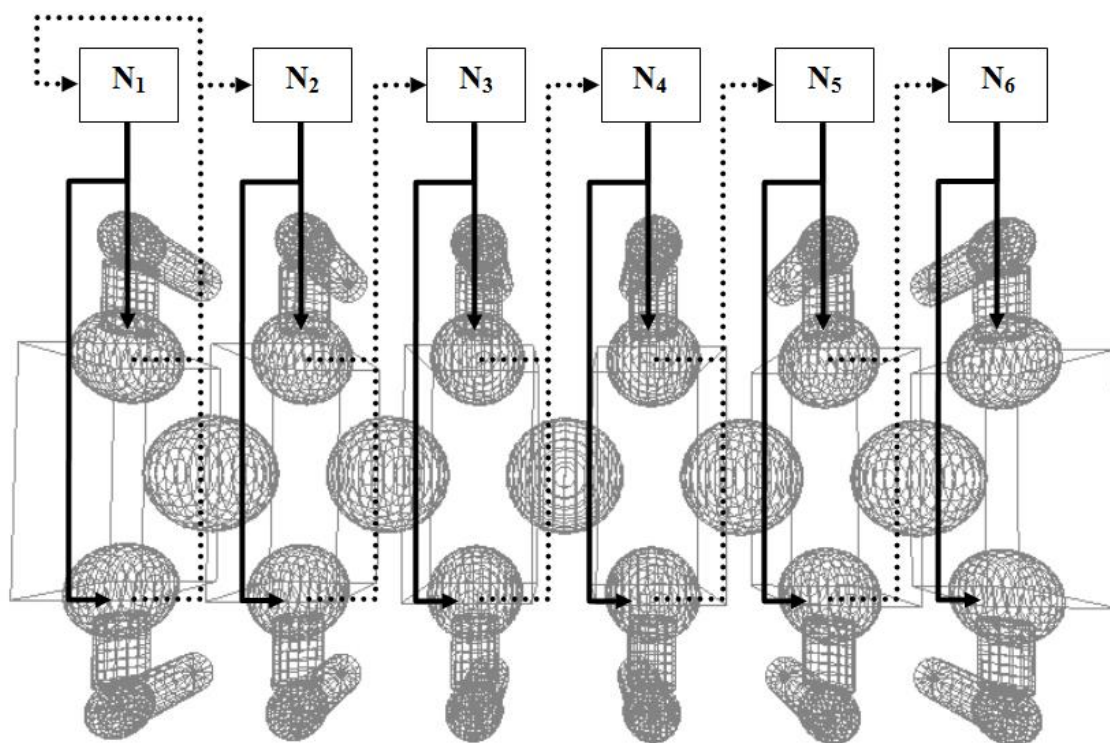


Рис.4. Схема нейронного контура управления движением многоногого робота

В данном эксперименте была поставлена задача обучить способу движения вперед модель многоногого робота (рис. 1b). Для системы управления роботом был выбран нейронный контур, состоящий из шести нейронов – по одному нейрону на каждый модуль робота (рис. 4). 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 Δt : чем выше скорость – тем больше награда.

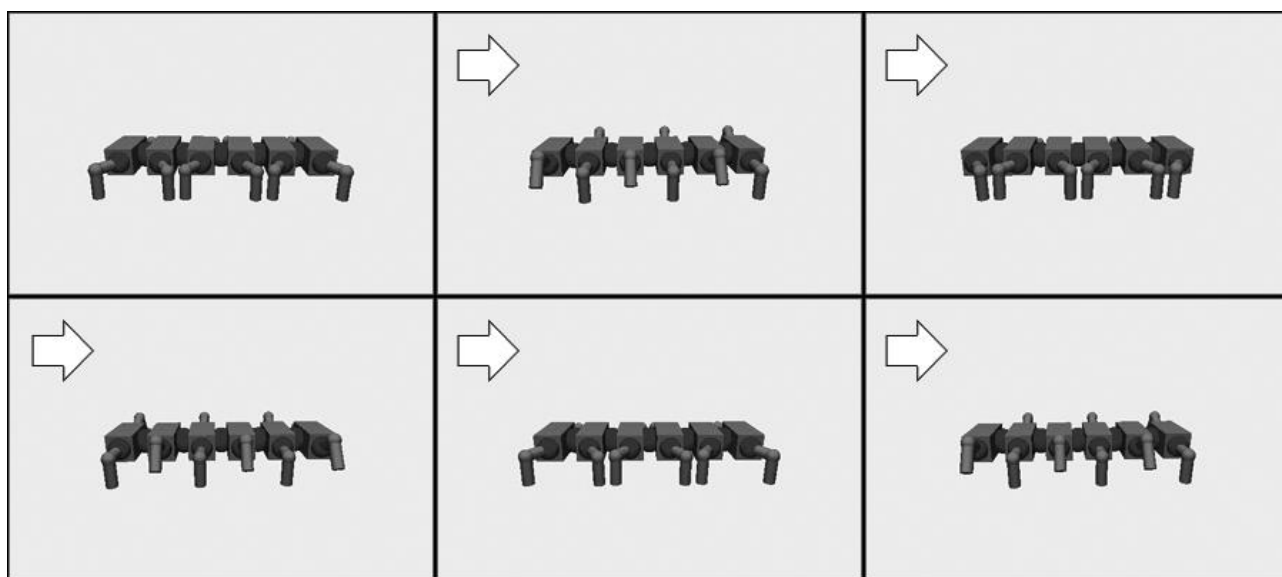


Рис.5. Последовательность движений многоногого робота при перемещении вперед

При помощи 3D-симулятор 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 5 приведен пример оптимальной последовательности движений, найденной в процессе обучения.

6. Система управления хоботовидным манипулятором

В данном эксперименте перед манипулятором (рис. 1с) была поставлена задача захватить цель, которая появляется в случайной позиции в пределах области, достижимой для 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.

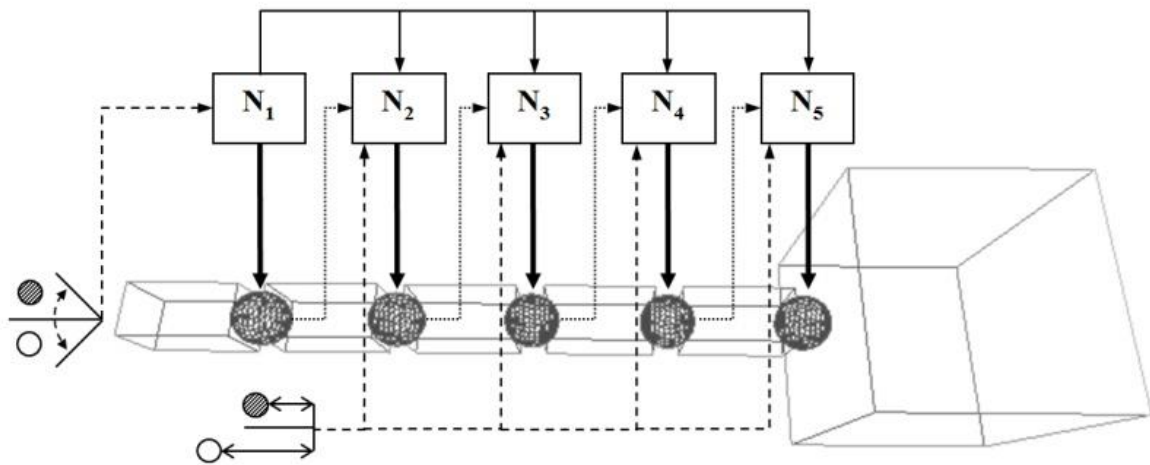


Рис.6. Схема нейронного контура управления хоботовидным манипулятором

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. 6). 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.

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.

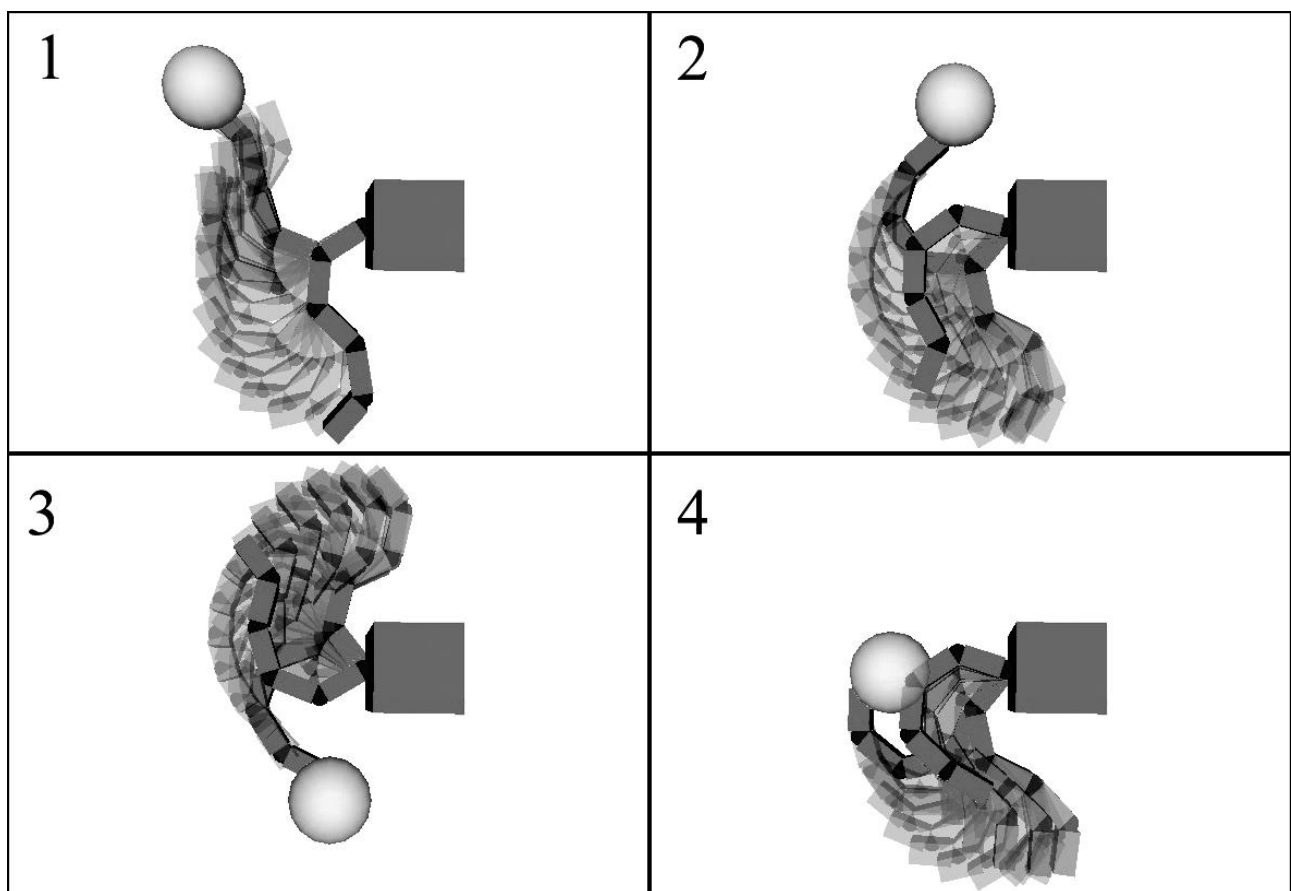


Рис.7. Примеры траекторий движения манипулятора при захвате цели

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, в результате обучения системе управления удастся обнаружить эффективные правила, обеспечивающие стопроцентный захват целей в указанной зоне. На рисунке 7 приведены четыре примера траекторий движения уже обученного хобота при захвате цели. Примеры со второго по четвертый показывают, что система управления научилась делать специальные подготовительные движения, чтобы

выйти на удобное положение для совершения захвата. 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, а затем успешно захватывает цель. Приведенные примеры являются демонстрацией того, что обнаруженные системой правила управления порождают достаточно сложное поведение, направленное на достижение цели.

7. Система управления многоногим роботом с двумя типами конечностей

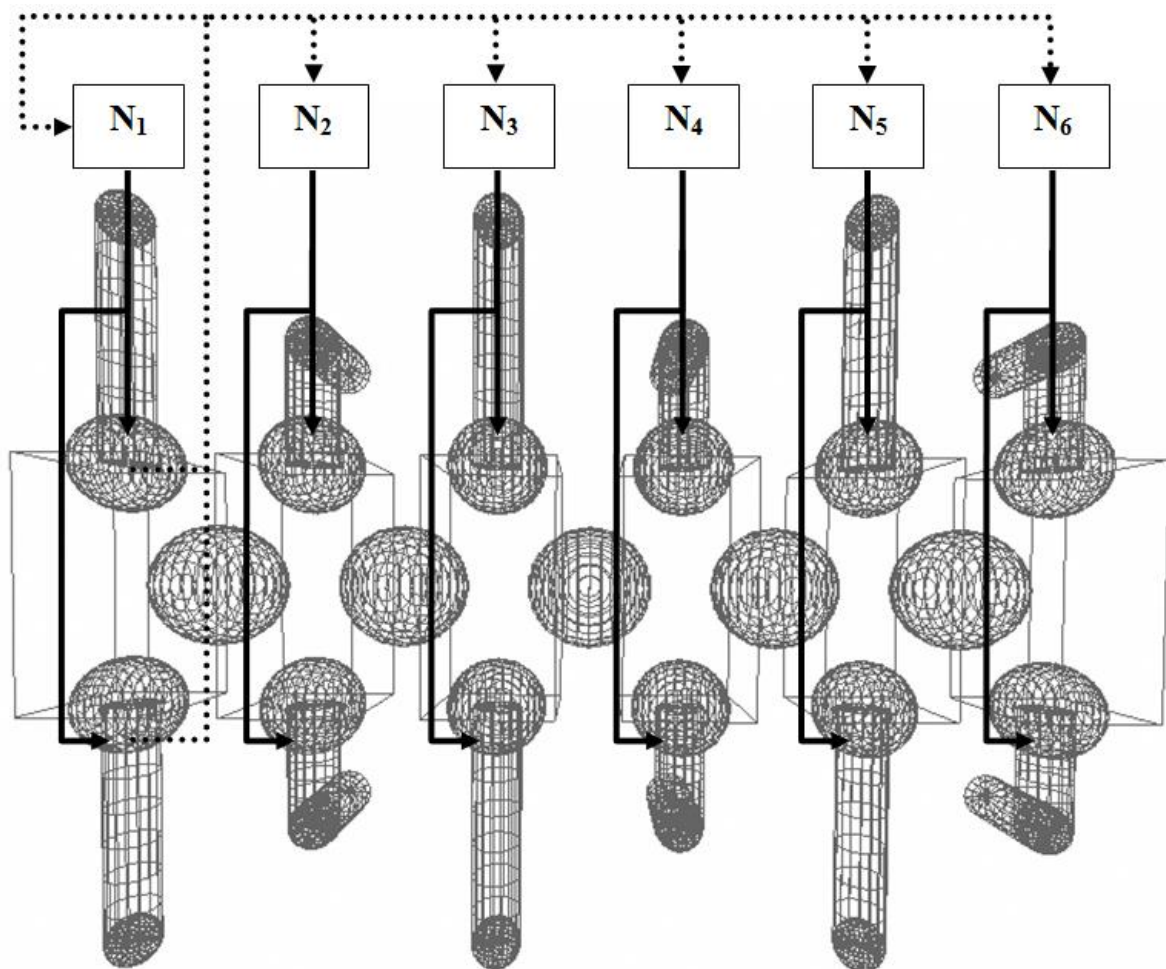


Рис.8. Схема нейронного контура управления многоногим роботом с двумя типами конечностей

Рассмотренные выше эксперименты показали высокую скорость обучения системы управления и хорошую масштабируемость относительно увеличения числа модулей, однако они были ограничены тем, что конструкции роботов состояли из одинаковых модулей.

Основной задачей последнего эксперимента являлась демонстрация применимости предложенного подхода для управления модульными роботами, состоящими из разных типов модулей. С этой целью был поставлен эксперимент по обучению способам передвижения вперед простого многоногого робота, имеющего конечности двух разных типов (рис. 1d). При этом Г-образные конечности могут двигаться только в горизонтальной плоскости, а прямые конечности – только в вертикальной. Очевидно, что данный робот может эффективно двинуться вперед только за счет продвижения Г-образных конечностей назад. Однако поскольку Г-образная конечность может двигаться только в горизонтальной плоскости, то для того, чтобы забросить ее вперед для следующего шага и при этом не сдвинуть робота в обратном направлении, необходимо задействовать прямые конечности, чтобы приподнять робота над землей. В результате, эффективное движение робота возможно только при согласованной работе модулей разных типов. Таким образом, выбранная конструкция робота, несмотря на простоту, является хорошей тестовой моделью для проверки возможностей системы обнаруживать согласованные управляющие правила для различных типов модулей.

Схема нейронного контура, выбранного для управления роботом, состояла из шести нейронов – по одному нейрону на каждый модуль робота (рис. 8). Каждый нейрон N_i , $i = 1, \dots, 6$ контролировал движения левой и правой конечности своего модуля, подавая активирующие сигналы на соответствующие угловые двигатели, вращающие конечности в суставе. Для упрощения задачи движения левой и правой конечностей каждого модуля были синхронизированы таким образом, что одна конечность зеркально повторяла движения другой. Таким образом, каждому нейрону достаточно было выдавать только один активирующий сигнал, чтобы привести в движение сразу обе конечности.

Первый нейрон N_1 получает на свой вход информацию о положении конечностей первого модуля. Эта же информация поступает на входы всех остальных нейронов N_i , $i = 2, \dots, 6$. Таким образом, в данной схеме состояние первого модуля, по сути, можно рассматривать как своеобразный счетчик тактов для всех остальных модулей.

Награда для системы управления рассчитывалась по факту завершения цикла выполнения шага и возврата конечностей первого модуля в исходную точку. Под шагом подразумевается вся последовательность действий, которая была выполнена в промежуток времени между текущим и предыдущим фактами нахождения конечностей в исходном состоянии. В качестве исходной точки было выбрано максимальное вертикальное положение конечностей первого модуля.

Вычисление награды осуществлялось следующим образом. Пусть в текущий момент времени t_1 положение конечностей первого модуля соответствуют исходной точке начала шага. Пусть t_0 – предыдущий момент времени, когда эти конечности находились в исходной точке. Тогда все действия в промежутке времени от $(t_0 + 1)$ до t_1 будут входить в цикл выполнения шага, а награда для этих моментов времени t , где $(t_0 + 1) \leq t \leq t_1$ и $(t_0 + 1) < t_1$, будет равна $r = S / (t_1 - (t_0 + 1))$. Где S – расстояние, которое преодолел робот по направлению вперед за этот же промежуток времени (от $(t_0 + 1)$ до t_1). В случае «пустого» шага, т.е. когда $t_1 = (t_0 + 1)$ и конечности первого модуля просто остаются в исходной точке два такта подряд, награда для момента времени t_1 устанавливается равной 0. Данная функция награды стимулирует систему управления находить такие последовательности действий, которые бы позволяли преодолевать как можно большее расстояние при совершении как можно меньшего числа действий.

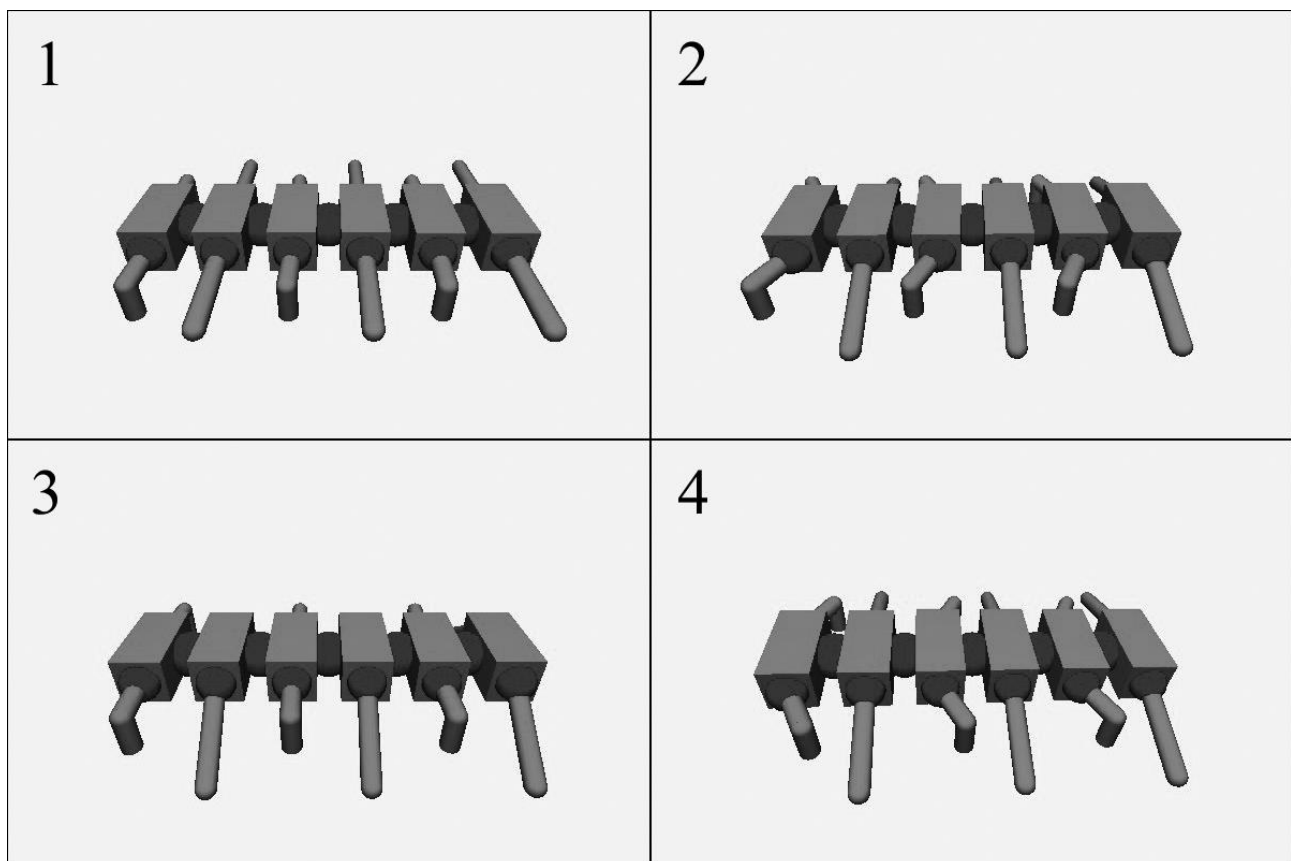


Рис.9. Последовательность движений многоногого робота с двумя типами конечностей при перемещении вперед

Используя симулятор 3D-симулятор, был проведен ряд успешных экспериментов по обучению рассмотренной системы управления способам передвижения. В серии экспериментов системе управления удавалось стабильно обнаруживать правила управления, обеспечивающие согласованные движения конечностей модулей разных типов, приводящие к эффективному перемещению робота вперед. На рисунке 9 приведен пример оптимальной последовательности движений, найденной в процессе обучения.

8. Заключение

В данной работе рассмотрен логико-вероятностный подход к задаче адаптивного управления модульными системами с большим числом степеней свободы. 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 конструкции робота. С уменьшением доли схожих модулей преимущества от использования общих правил теряются.

Список литературы

1. Витяев Е.Е. Извлечение знаний из данных. Компьютерное познание. Модели когнитивных процессов. – Новосибирск: НГУ, 2006. – 293 с.
2. Демин А.В. Модель адаптивной системы управления и ее применение для управления движением виртуального робота // Молодой ученый. – 2012. – № 11 (46) – С. 114-119.
3. Демин А. В. Обучающаяся система управления движением для 3D модели многоногого робота // Молодой ученый. — 2015. — №19 (99). — С. 74-78.
4. Демин А.В. Обучение способам передвижения виртуальной модели змеевидного робота // Молодой ученый. – 2014. – № 19 (78) – С. 147-150.
5. Демин А.В., Витяев Е.Е. Логическая модель адаптивной системы управления // Нейроинформатика. – 2008. – Т. 3. – № 1. – С. 79-107.
6. Демин А.В. Обучающаяся модель управления хемотаксисом нематоды *C.Elegans* // Нейроинформатика. – 2013. – Т. 7. – № 1. – С. 29-41.

7. Bongard J.C. Evolutionary Robotics // Communications of the ACM. – 2013. – Vol. 56. – No. 8. – pp. 74-83.
8. Daoxiong Gong, Jie Yan, Guoyu Zuo. A Review of Gait Optimization Based on Evolutionary Computation // Applied Computational Intelligence and Soft Computing. – 2010. – vol. 2010. – Article ID 413179. – 12 p.
9. 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.
10. 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. – V. 7. – pp. 9-14.
11. Ito K., Matsuno F. Control of hyper-redundant robot using QDSEGA // Proceedings of the 41st SICE Annual Conference (2002). – 2002. – V. 3. – pp. 1499-1504.
12. 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.
13. 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.
14. Mataric M., Cliff D. Challenges in evolving controllers for physical robots // Robotics and Autonomous Systems. – October 1996. – 19(1). – pp. 67-83.
15. Smith R. Open Dynamics Engine. – URL: <http://ode.org/>.
16. Stoy K., Brandt D., Christensen D.J. Self-Reconfigurable robots: an introduction // Intelligent robotics and autonomous agents series. – MIT Press, 2010. – 216 p.
17. Tanev I., Ray T., Buller A. Automated Evolutionary Design, Robustness and Adaptation of Sidewinding Locomotion of Simulated Snake-like Robot // IEEE Transactions on Robotics. – V.21. – N. 4. – August 2005. – pp.632-645.
18. Valsalam V. K. Miikkulainen R. Modular neuroevolution for multilegged locomotion // In Proceedings of GECCO. – 2008. – pp. 265-272.
19. 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.

