

**Акционерное общество «Институт точной механики и вычислительной  
техники им. С.А. Лебедева РАН»**

*На правах рукописи*

Шестаков Евгений Игоревич

**Алгоритмическое и программное обеспечение адаптивной системы  
управления модульными роботами**

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

**Диссертация**

на соискание учёной степени  
кандидата технических наук

Научный руководитель:

доктор физико-математических наук, профессор

Жданов Александр Аркадьевич

Москва – 2021

## СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	4
ВВЕДЕНИЕ .....	5
ГЛАВА 1. АКТУАЛЬНЫЕ ПРОБЛЕМЫ АЛГОРИТМИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЬНЫХ РОБОТОВ .....	11
1.1. Алгоритмы управления движением модульных роботов.....	12
1.2. Алгоритмы самообучения и искусственного интеллекта для управления модульными роботами .....	20
1.3. Программные средства моделирования модульных роботов .....	30
Выводы по главе .....	35
ГЛАВА 2. АДАПТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ МОДУЛЬНЫМИ РОБОТАМИ .....	37
2.1. Способ построения адаптивной системы управления модульными роботами.....	37
2.2. Математическая модель метода автономного адаптивного управления .....	44
2.3. Алгоритмы адаптивной системы управления движением модульных роботов в конфигурациях «манипулятор» и «шагающая платформа» .....	47
Выводы по главе .....	54
ГЛАВА 3. СРЕДСТВА МОДЕЛИРОВАНИЯ МОДУЛЬНЫХ РОБОТОВ И ИХ ПРОГРАММНОЕ ВЗАИМОДЕЙСТВИЕ С МАКЕТНЫМ ОБРАЗЦОМ МОДУЛЬНОГО РОБОТА.....	56
3.1. Способ построения программных средств моделирования модульных роботов.....	56

3.2. Компоненты программного обеспечения для моделирования адаптивной системы управления движением модульных роботов .....	61
3.3. Программное взаимодействие средств моделирования с макетным образцом модульного робота .....	77
Выводы по главе .....	86
<b>ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ АЛГОРИТМОВ АДАПТИВНОГО УПРАВЛЕНИЯ МОДУЛЬНЫМИ РОБОТАМИ.....</b>	<b>87</b>
4.1. Экспериментальные исследования адаптивной системы управления движением модульных роботов в конфигурациях «манипулятор» и «шагающая платформа».....	87
4.2. Преимущества предложенных алгоритмов адаптивного управления модульными роботами по сравнению с другими методами управления.....	95
Выводы по главе .....	97
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>99</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>104</b>

## **СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

ААУ – Автономное Адаптивное Управление

БЗ – База Знаний

ГА – Генетический алгоритм

ГП – Генетическое программирование

ИНС – Искусственная нейронная сеть

КА – Конечный автомат

МР – Модульный робот

ПО – программное обеспечение

СДКА – Самообучающийся динамический конечный автомат

УС – Управляющая система

ФРО – Формирование и распознавание образов

ЦГУА – Центральный генератор упорядоченной активности

## ВВЕДЕНИЕ

### Актуальность темы работы

В настоящее время наблюдается активная интеллектуализация программно-алгоритмических комплексов на основе различных методов, например таких, как нечеткая логика, нейронные сети, эволюционные алгоритмы, обучение с подкреплением и других. Одним из актуальных направлений развития систем искусственного интеллекта является разработка адаптивных систем управления. Данная диссертационная работа посвящена проблемам алгоритмического и программного обеспечения адаптивной системы управления модульными роботами. Модульный робот представляет собой единую многозвенную конструкцию, состоящую из однотипных механизмов (модулей). В настоящее время модульные роботы находят все более широкое применение в различных сферах человеческой деятельности, поскольку модульность конструкции и аппаратных средств обуславливает ряд преимуществ модульных роботов по сравнению с традиционными робототехническими системами.

Современные тенденции в робототехнике включают в себя активное применение методов искусственного интеллекта и машинного обучения. Вместе с тем, как показывает анализ публикаций, современные алгоритмы управления модульными роботами не предусматривают автоматической адаптивности робота к изменению свойств окружающей среды, самого робота или выполняемой им работы. В основном широко используется детерминированное управление, при котором желаемое качество управления не достигается по ряду причин, связанных с отсутствием адаптивных свойств у систем управления. Во-первых, модульные роботы обладают большим количеством степеней свободы, что, в свою очередь, делает весьма затруднительным, а в ряде случаев и невозможным, нахождение точных аналитических решений по управлению. Во-

вторых, свойства и окружающей среды, и работа, и выполняемых им заданий могут меняться незапланированным образом. К таким факторам можно отнести, например, изменение свойств подстилающей поверхности вследствие изменения погодных условий или выход из строя отдельных модулей. В связи с этим, актуальной задачей является разработка способа построения программного обеспечения, алгоритмов и методов адаптивного управления применительно к модульным роботам.

### **Степень разработанности проблемы**

Диссертационное исследование опирается на работы отечественной научной школы в области искусственного интеллекта и адаптивного управления, к которой относятся Д.А. Поспелов, М.Л. Цетлин, В.И. Варшавский, М.Г. Гаазе-Рапопорт, Я.З. Цыпкин, А.А. Жданов и ряд других, а также на основные достижения в области управления модульными роботами, представленными в работах следующих авторов: Марк Йим (M. Yim), Шигео Хироце (S. Hirose), К.Стой (K. Støy), В-М. Шен (W.-M. Shen), П. Уилл (P. Will), Войцех Вонашек (V.Vonásek), Е. Йошида (Eiichi Yoshida), Сатоши Мурата (Satoshi Murata), Хуан Гонсалес-Гомес и ряда других. В нашей стране достижения в этой области принадлежат С.В. Манько, М.В. Кадочникову, А.Л. Ронжину, Ю.В. Подураеву, А.А. Иванову, О.А. Шмакову, Д.А. Демидову и др.

**Объектом диссертационного исследования** является система управления многозвенными объектами. **Предметом исследования** являются алгоритмы и программное обеспечение адаптивного управления модульными роботами.

**Цель исследования** состоит в разработке способа построения программного обеспечения для интеллектуальных адаптивных систем управления модульными роботами.

### **Задачи исследования:**

- Разработка способа построения, модели и алгоритмов адаптивной системы управления модульными роботами.
- Разработка способа построения программных средств моделирования модульных роботов.
- Разработка способа программного взаимодействия средств программного моделирования и макетного образца модульного робота.
- Разработка программного обеспечения для моделирования модульных роботов и экспериментальная проверка алгоритмов адаптивного управления модульными роботами.

### **Методы исследования**

В работе использовались методы линейной алгебры, теории управления, метод Автономного Адаптивного Управления, методы компьютерного и аппаратного моделирования, а также элементы теории графов.

### **Теоретическая значимость**

Разработаны способ построения адаптивной системы управления модульными роботами, способ построения программных средств моделирования модульных роботов, предложены и экспериментально проверены алгоритмы адаптивного управления, предложен способ описания модульных конфигураций и алгоритмы построения программных моделей, которые могут быть использованы для развития теории управления многозвенными объектами.

**Практическая значимость** полученных результатов состоит в том, что предложенный способ построения системы управления и разработанные адаптивные алгоритмы можно применять при разработке опытных и серийных образцов многозвенных роботов различного типа и назначения. В работе приведены экспериментальные исследования разработанных алгоритмов,

демонстрирующие адаптивность и самообучаемость модульных роботов. Разработанное программное обеспечение для моделирования модульных роботов можно применять в учебных и научно-исследовательских целях.

**Научная новизна** диссертационной работы заключается в получении следующих оригинальных результатов:

- разработан способ построения адаптивной системы управления модульными роботами;
- разработан способ построения программных средств моделирования модульных роботов;
- предложен способ описания конфигураций модульных роботов и алгоритм автоматического построения программных моделей по файлу описания;
- разработано и экспериментально проверено алгоритмическое и программное обеспечение адаптивной системы управления модульными роботами в конфигурациях «манипулятор» и «шагающая платформа».

#### **Положения, выносимые на защиту**

1. Способ построения адаптивной системы управления модульными роботами.
2. Способ построения программного обеспечения для моделирования модульных роботов.
3. Алгоритмы адаптивного управления модульными роботами в конфигурациях «манипулятор» и «шагающая платформа».
4. Способ описания конфигураций модульных роботов и алгоритм автоматического построения 3-мерной модели роботов по файлу описания.

**Личный вклад.** Все представленные в работе результаты получены лично автором, а именно: предложены способ построения, модель и алгоритмы адаптивной системы управления модульными роботами на основе метода Автономного Адаптивного Управления, способ построения программного обеспечения для моделирования модульных роботов. На основе предложенного способа разработано программное обеспечение, включая программную реализацию адаптивного управления модульными роботами в конфигурациях «манипулятор» и «шагающая платформа», проведены экспериментальные исследования предложенных подходов.

**Публикации.** Основные результаты исследования отражены в 10 печатных работах [31,41,85,86,90,104,119-122], включая рецензируемые журналы, входящие в базу Scopus [31,41] и список ВАК [104], а также в одном свидетельстве о регистрации программы для ЭВМ [87].

В работе [104] автором предложена структура программных средств моделирования, проведен обзор существующих средств моделирования, разработаны способ описания модульных конфигураций, алгоритм построения программных моделей, макетный образец модульного робота и программные средства взаимодействия макетного образца с программным комплексом моделирования. В статьях [120,121] автором предложены способы и алгоритмы адаптивного управления модульным манипулятором и описаны результаты проведенных автором экспериментальных исследований. В работах [31,90,122] автором описаны алгоритмы самообучения шагающего модульного робота и результаты проведенных экспериментальных исследований. В публикации [41] автор провел обзор современных архитектур систем управления модульными роботами. В [85] автором рассмотрены возможности применения вероятностных автоматов для реализации механизмов самообучения. В [86] автором проведен обзор возможностей теории конечных автоматов в задачах управления роботами.

**Апробация работы.** Основные результаты диссертационного исследования докладывались в рамках следующих мероприятий:

- XXII Международная научно-техническая конференция "Нейроинформатика-2020" (г. Москва, Россия, 12-16 октября 2020 г.)

- Нейронаука для медицины и психологии: XVI Международный междисциплинарный конгресс (Судак, Крым, Россия; 6–16 октября 2020 г.)

- Третья Международная Конференция по интерактивной коллаборативной робототехнике [ICR-2019] (г. Лейпциг, Германия, 18-22 сентября 2018)

- 10-ая Всероссийская Мультиконференция по проблемам управления [МКПУ-2017] (с. Дивноморское, г. Геленджик, Россия, 25–30 сентября 2017)

- XXIV международная научно-техническая конференция "Современные технологии в задачах управления, автоматике и обработки информации" (г. Алушта, Россия, 14–20 сентября 2015)

**Объем и структура работы.** Диссертация состоит из введения, 4-х глав, заключения, списка литературы и списка сокращений. Полный объем диссертации составляет 118 страниц, включая 47 рисунков, 16 таблиц, 22 формулы. Список литературы содержит 122 наименования.

## **ГЛАВА 1. АКТУАЛЬНЫЕ ПРОБЛЕМЫ АЛГОРИТМИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЬНЫХ РОБОТОВ**

В данной главе приведен обзор основных алгоритмов управления модульными роботами, рассмотрены алгоритмы самообучения и искусственного интеллекта, а также программные средства для моделирования модульных роботов. Определена актуальная задача разработки нового способа построения адаптивной системы управления и программных средств моделирования модульных роботов.

Концепция построения модульных роботов (МР) предполагает наличие однотипных мехатронных модулей, объединяемых в единую многосвязную конструкцию. Типовые мехатронные модули, как правило, включают в свой состав интерфейсные площадки для обеспечения взаимного механического, электрического и информационного соединения, автономный источник питания, различные виды датчиков, управляющий контроллер, один или несколько двигателей, простейшие механические передачи. Подобная конструкция типовых модулей позволяет обеспечить их автоматическую стыковку и расстыковку для формирования необходимой кинематической структуры робота (конфигурации) в зависимости от целей и условий функционирования.

Функциональные возможности МР обеспечивают их востребованность в различных областях: в космической и промышленной отраслях, в системе образования, в бытовой сфере [5,19, 20, 23, 26, 27, 34, 38, 40, 48, 54, 55]. Всё это обуславливает высокую актуальность теоретических и прикладных исследований в этой области, включая проблемы построения программного обеспечения для систем управления данным классом объектов.

## 1.1. Алгоритмы управления движением модульных роботов

В настоящее время существует несколько способов программного управления модульными роботами. Рассмотрим подробнее каждый из них.

### Управляющие таблицы (*Gait control tables*)

Самым распространенным способом управления движением робота являются управляющие таблицы. Каждая строка таблицы представляет собой вектор обобщенных координат робота  $C(t)$ , который однозначно определяет положение всех модулей на соответствующем шаге алгоритма:

$$\overline{C}(t) = \{Q_1, \dots, Q_n\}, \quad (1)$$

где  $Q_i = \{q_{i1}, \dots, q_{im}\}$  – набор обобщенных координат  $i$ -го модуля,

$q_{ij}$  – обобщенная координата  $j$ -ой степени свободы  $i$ -го модуля.

Таким образом, управляющая таблица является последовательностью векторов  $\overline{C}(t_0), \overline{C}(t_1), \dots, \overline{C}(t_k)$ . В качестве иллюстрирующего примера приведены таблицы движения (табл. 1.2.1 и 1.2.2) МР с 1-степенными модулями в конфигурации «гусеница» (рис. 1.2.1а) и «колесо» (рис. 1.2.1б).

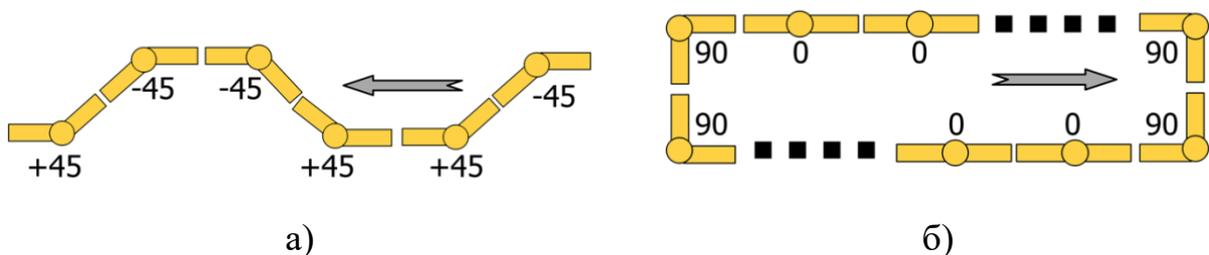


Рис. 1.2.1. Иллюстрация алгоритмов передвижения с использованием таблиц: а) конфигурация типа «гусеница», б) конфигурация типа «колесо» [43].

Таблица 1.2.1. Пример таблицы управления МР в конфигурации «гусеница» [43].

Шаг	Идентификатор модуля					
	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>	<i>M6</i>
0	+45°	-45°	-45°	+45°	+45°	-45°
1	-45°	-45°	+45°	+45°	-45°	-45°
2	-45°	+45°	+45°	-45°	-45°	+45°
3	+45°	+45°	-45°	-45°	+45°	+45°

Таблица 1.2.2. Пример таблицы управления МР в конфигурации «колесо» [43].

Шаг	Идентификатор модуля									
	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>	<i>M6</i>	<i>M7</i>	<i>M8</i>	<i>M9</i>	<i>M10</i>
0	0°	0°	+90°	+90°	0°	0°	0°	+90°	+90°	0°
1	0°	0°	0°	+90°	+90°	0°	0°	0°	+90°	+90°
2	+90°	0°	0°	0°	+90°	+90°	0°	0°	0°	+90°
3	+90°	+90°	0°	0°	0°	+90°	+90°	0°	0°	0°
4	0°	+90°	+90°	0°	0°	0°	+90°	+90°	0°	0°

Поскольку хранение таблиц для управления различными конфигурациями занимает существенное количество памяти, был разработан алгоритм автоматического построения управляющих таблиц [15, 16].

В качестве примера работы данного алгоритма рассмотрим формирование управляющей таблицы для линейной модульной структуры, которая должна двигаться в форме синусоиды, заданной функцией  $f(x, t)$ . Положим, что координаты  $(x_i, y_i)$  являются координатами центра  $i$ -го модуля. В начальный момент времени  $t_0$  все углы в сочленениях робота установлены в нулевое положение (рис. 1.2.2а). Затем рассчитывается вектор углового положения  $\overrightarrow{C}(t_0)$  (строка управляющей таблицы) путем изменения каждого угла в сочленении в соответствии с волновой функцией  $f(x, t_0)$  так, что  $y_i = f(x_i, t_0)$ , при этом сохраняя расстояние  $L$  между соседними модулями (рис. 1.2.2б). Затем волна смещается, (рис. 1.2.2в) и выполняется расчёт нового вектора  $\overrightarrow{C}(t_1)$  (рис. 1.2.2г). Сдвиг волны и расчёт нового

вектора углового положения происходит до тех пор, пока фаза заданной волновой функции не достигнет своего начального состояния.

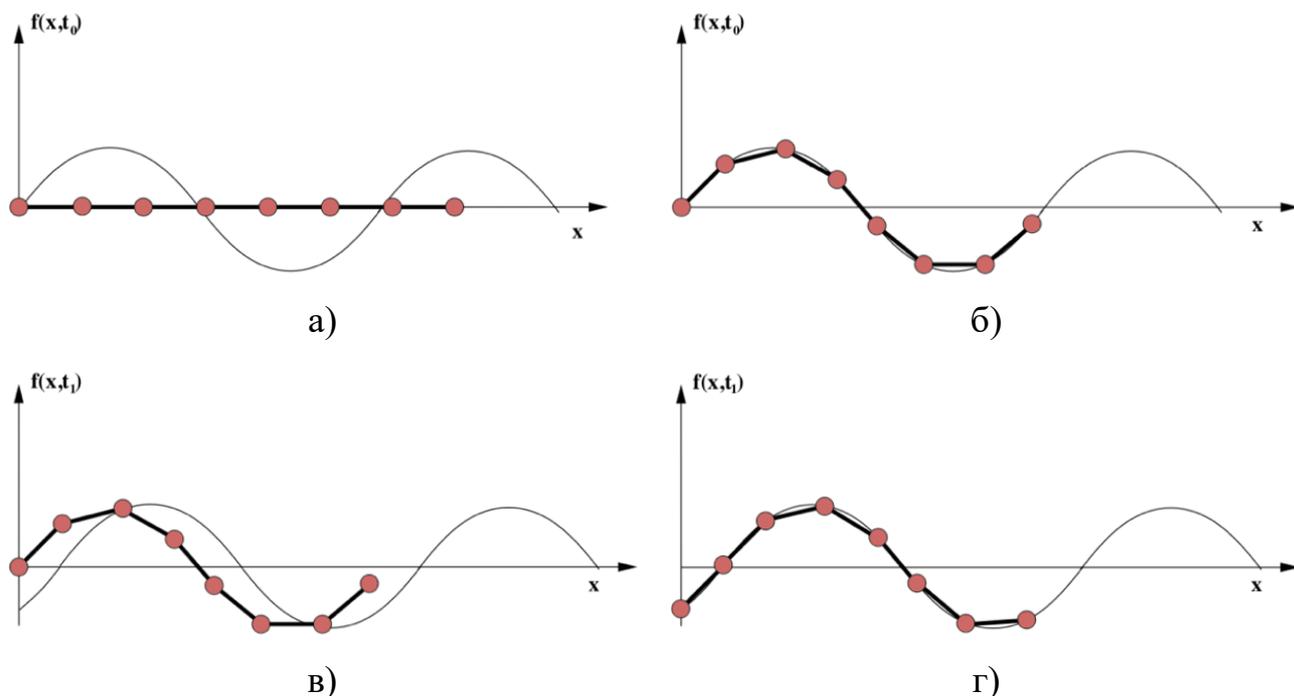


Рис. 1.2.2. Визуализация алгоритма автоматического создания управляющих таблиц а) начальное состояние, б) модульная структура описывает заданную волну, в) смещение волны, г) повторное изменение обобщенных координат робота по отношению к волне [15, 16].

### ***Центральный генератор упорядоченной активности (Central Pattern Generator)***

Другим способом управления МР является центральный генератор упорядоченной активности (ЦГУА) — нейронная сеть, подающая сигналы, наглядно похожие на ритмично подаваемые осмысленные сигналы головного мозга опорно-двигательному аппарату, причём имеется возможность подавать их даже при отсутствии обратной связи от конечностей и других частей тела с целевыми мышцами [32, 24].

Модели ЦГУА описываются нелинейными дифференциальными уравнениями [21, 25], которые должны быть решены с помощью численных

методов, например, метода Рунге-Кутты. Для достижения желаемого передвижения решения должны находиться с интервалом не более 15 мс.

### *Гармонический осциллятор*

Гармонические осцилляторы представляют упрощенную модель ЦГУА, описываемую периодической синусоидальной функцией. В простейшем случае для робота, состоящего из типовых модулей с одной степенью свободы, она имеет следующий вид:

$$q_i = A_i \sin\left(\frac{2\pi}{T_i}t + \Phi_i\right) + O_i, \quad i \in \{1, 2, \dots, n\}, \quad (2)$$

где  $q_i$  – угол в  $i$ -ом модуле,

$A_i$  – амплитуда,

$T_i$  – период,

$\Phi_i$  – фаза,

$O_i$  – смещение,

$n$  – количество модулей.

Таким образом, для каждого сочленения каждого модуля требуется наличие только четырех параметров: периода  $T$ , амплитуды  $A$ , фазы  $\Phi$  и смещения  $O$ . Частота гармонического осциллятора, как правило, одинакова для всех модулей системы и зачастую пропорциональна скорости движения робота. Стоит отметить, что данная модель легко преобразуется в управляющие таблицы путем установки параметра амплитуды в ноль.

В работе [17] данный способ управления использовался для исследования различных вариантов передвижения 8-ми модульного робота Y1. В целях упрощения исследования движения роботов, авторами применялся ряд допущений:

- Все модули двигаются с одинаковым периодом  $T_i = T$ .

- Модули условно разделены на две группы: вертикальные (V) и горизонтальные (H) модули (рис 1.2.3).
- Все вертикальные (горизонтальные) модули имеют одинаковые амплитуду  $A_V$  ( $A_H$ ), смещение  $O_V$  ( $O_H$ ), разницу фаз между двумя соседними модулями  $\Delta\Phi_V$  ( $\Delta\Phi_H$ ).

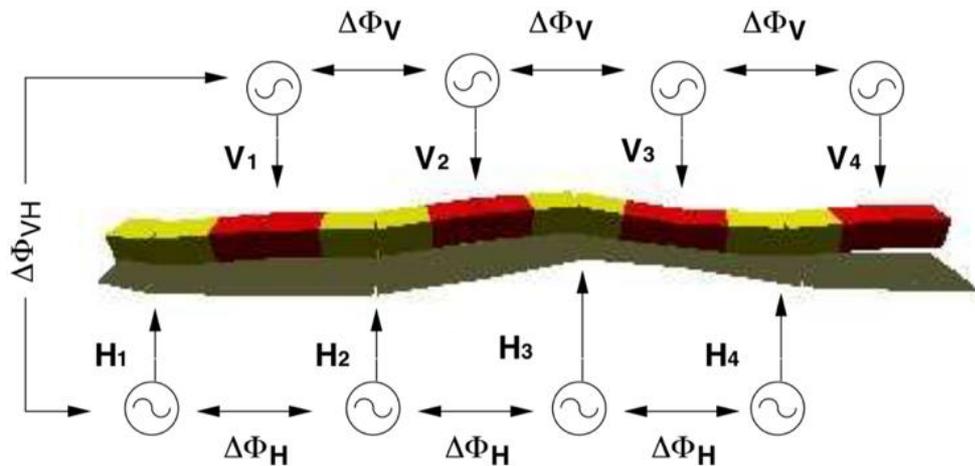


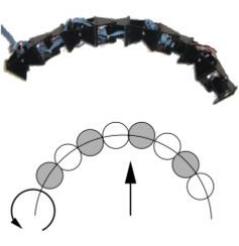
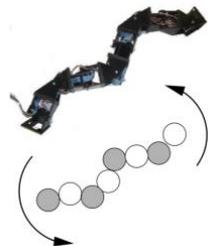
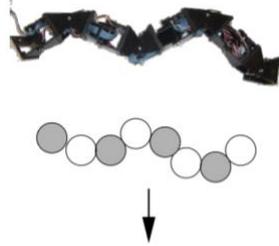
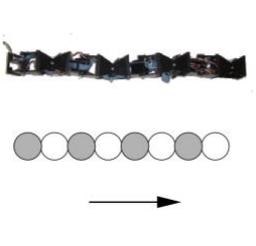
Рис. 1.2.3. Графическое представление управления модулями Y1 при помощи гармонического синусоидального осциллятора [17].

В зависимости от того к какой группе принадлежит модуль, он управляется по одному из двух законов (3) – (4). В результате различные способы передвижения модульного робота задаются с использованием всего 3-х параметров – амплитуды, фазы и смещения, как показано в табл. 1.2.3.

$$V_i = A_V \sin\left(\frac{2\pi}{T}t + (i-1)\Delta\Phi_V\right) + O_V, \quad (3)$$

$$H_j = A_H \sin\left(\frac{2\pi}{T}t + (j-1)\Delta\Phi_H + \Delta\Phi_{VH}\right) + O_H \quad (4)$$

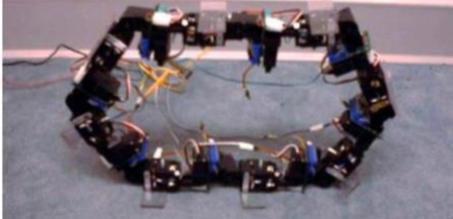
Таблица 1.2.3. Варианты движения 8-ми модульного робота Y1 и параметры гармонического осциллятора [17].

Перекачивание (Rolling)	Поворот (Rotating)	Боковое движение (Lateral Shifting)	Поворот (Turning)	Прямолинейное синусоидальное движение (1D Sinusoidal)
				
$A_H \neq 0$	$A_H \neq 0$	$A_V \neq 0$	$A_H = 0$	$A_H = 0$
$O_H = 0$	$O_H = 0$	$O_V = 0$	$O_H \neq 0$	$O_H = 0$
$\Delta\Phi_V = 0$	$\Delta\Phi_V = 120$	$\Delta\Phi_V = 100$	$\Delta\Phi_V = 120$	$\Delta\Phi_V = 120$
$\Delta\Phi_H = 0$	$\Delta\Phi_H = 50$	$\Delta\Phi_H = 100$		
$\Delta\Phi_{VH} = 90$	$\Delta\Phi_{VH} = 0$	$\Delta\Phi_{VH} = 0$		

Разработчики робота CONRO в работе [47] предложили свой вариант реализации алгоритма управления линейными конфигурациями (табл. 1.2.4, листинг 1.1), в основе которого лежит гармонический осциллятор. Для его работы необходимо задать три компонента:

1. Циклическую последовательность действий  $A(t)$ , т.е. непосредственно гармонический осциллятор.
2.  $T$  - период цикла.
3.  $d$  – задержка перед переходом на следующий цикл.

Таблица 1.2.4. Варианты движения МР CONRO и параметры алгоритма на основе гармонического осциллятора [47].

Прямолинейное движение в змеевидной конфигурации	Прямолинейное движение в конфигурации колеса
	
$T = 180$	
$d = \frac{T}{5}$	$d = \frac{T}{4}$
$A(t) = 50^\circ \sin\left(\frac{2\pi}{T}t\right)$	$A(t) = \begin{cases} 60^\circ \left(1 - \sin\left(\frac{2\pi}{T}t\right)\right), & \text{если } t \leq \frac{T}{2} \\ 60^\circ, & \text{если } t > \frac{T}{2} \end{cases}$

**Листинг 1.1.** Универсальный алгоритм управления роботом CONRO [47]

$t \leftarrow 0$

**ПОКА** (ИСТИНА)

**ЕСЛИ** (t=d) **ТО**

        <Отправить сигнал дочернему модулю>

**ЕСЛИ** <Сигнал получен от родительского модуля> **ТО**

$t \leftarrow 0$

        <Выполнить действие  $A(t)$ >

$t \leftarrow (t+1) \bmod T$

### **Фазовый автомат**

Фазовые автоматы [56] – это разновидность конечных автоматов, дополненных фазовой задержкой – вещественным числом в диапазоне от 0 до 1. С математической точки зрения, фазовый автомат описывается как

$$PA = (S_t, S_r, E, A, f, s_0, \delta), \quad (5)$$

где  $S_t$  – множество внутренних состояний,

$S_r$  – множество показаний датчиков,

$E$  – множество событий, зависящих от показаний датчиков,

$A$  – множество действий. Действие может быть логическим, например изменение параметров управления, или физическим.

$f$  – функция переходов  $S_t \times S_r \times E \rightarrow S_t \times A$ ,

$s_0$  – начальное состояние,

$\delta$  – фазовая задержка по отношению к начальному состоянию.

Разработчики MR PolyBot провели ряд модельных и натурных исследований по способам управления роботами в различных конфигурациях при помощи фазового автомата [56].

Как мы видим, рассмотренные выше алгоритмы управления MR имеют ряд существенных ограничений. Из-за большого числа степеней свободы разработка адекватного алгоритма управления становится очень трудоемкой задачей. Кроме того, разработчику весьма затруднительно заранее запрограммировать все формы движения в той или иной конфигурации и учесть все ситуации, в которых может оказаться робот. Также существует возможность выхода из строя отдельных модулей или изменения свойств окружающей среды. Учитывать все эти факторы в жестко детерминированных алгоритмах управления представляется невозможным. Таким образом, становится актуальным применение методов искусственного интеллекта для обучения и управления MR.

## 1.2. Алгоритмы самообучения и искусственного интеллекта для управления модульными роботами

С возрастанием количества модулей, используемых в роботе, возрастает и число возможных конфигураций. В частности, в конфигурации «шагающая платформа» используется большое количество модулей, и соответственно может быть различное количество конечностей и количество модулей в каждой из них. При этом невозможно заранее заложить алгоритмы управления для всех потенциально возможных конфигураций. Также необходимо адекватно реагировать на изменения свойств среды или робота, например в случае выхода из строя одного из модулей.

В связи с этим представляется актуальным вопрос о применении к модульным роботам методов искусственного интеллекта, а именно механизмов самообучения и адаптивных методов управления. К таким методам относятся искусственные нейронные сети, эволюционные алгоритмы, самообучающиеся конечные автоматы, метод Автономного Адаптивного Управления (ААУ) и др. Рассмотрим их подробнее.

*Эволюционные алгоритмы* являются методами адаптивного поиска для решения задач оптимизации и используют аналоги механизма естественного отбора и генетического наследования в живой природе, позволяя найти оптимальное решение за приемлемое время. К алгоритмам данного класса относятся генетические алгоритмы (ГА), генетическое программирование (ГП) и нейроэволюционные методы.

Классический ГА работает по схеме, представленной на рис. 1.3.1. Задача формализуется таким образом, чтобы ее решение было закодировано в виде битового вектора. Набор потенциальных решений (особей), представленных в битовом виде, образуют популяцию. Каждая особь из текущей популяции оценивается функцией пригодности (*fitness function*), которая и задает критерий

оптимизации. Наиболее «приспособленные» особи принимают участие в генерации новой «популяции» - нового набора возможных решений, через механизмы скрещивания и мутации.

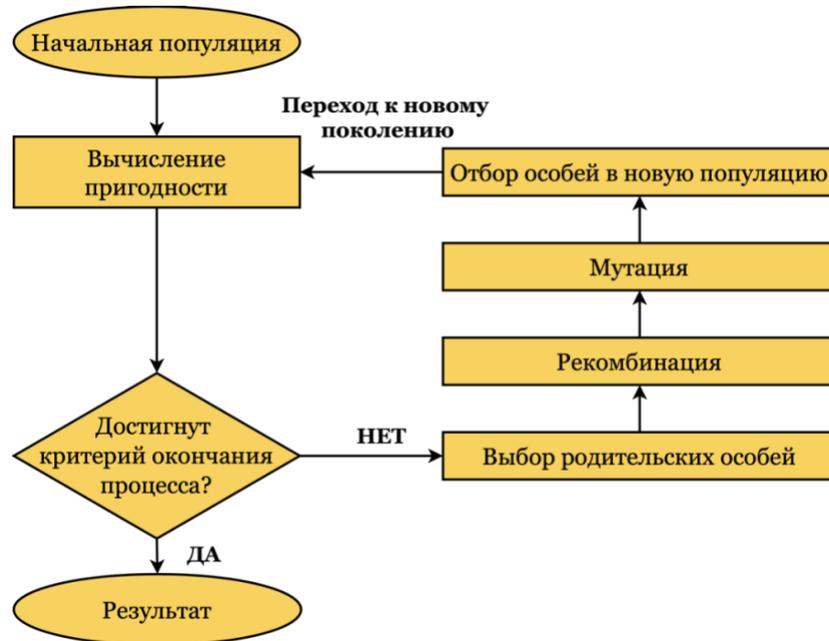


Рис. 1.3.1. Блок-схема работы стандартного генетического алгоритма.

Операция скрещивания подразумевает под собой взаимный обмен генами двух особей, в результате чего появляются два новых потенциальных решения. В общем случае оператор скрещивания является многоточечным, и подлежащие обмену фрагменты генов, определяются случайным выбором точек разрыва в векторе генов родительских особей (рис.1.3.2). На практике чаще всего используют одноточечный оператор скрещивания, имеющий только одну точку разрыва. Скрещивание образует аналог постепенной оптимизации по градиенту [105].

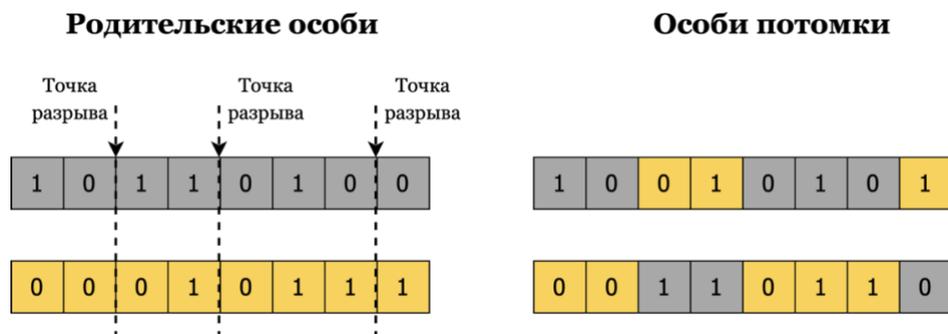


Рис. 1.3.2. Иллюстрация операции многоточечного скрещивания особей.

Операция мутации подразумевает инвертирование случайного гена в особи (рис. 1.3.3.) и необходима для вывода решений из областей локальных экстремумов. На практике вероятность мутации выбирают в диапазоне от 0.5-1%, чтобы не свести работу ГА к методу случайного поиска [100].



Рис. 1.3.3. Иллюстрация операции мутации особи.

Критерием остановки работы ГА может быть нахождение оптимального решения или исчерпание количества итераций алгоритма или времени его работы.

Сложность практического применения генетических алгоритмов связана с выбором способа кодирования решений и с формированием функции пригодности. В целом ГА, как аналог природного метода оптимизации биологических видов – это скорее метод оптимизации конструкции, системы управления или алгоритма управления, чем собственно метод адаптивного управления [69].

Интеллектуальные системы на основе *искусственных нейронных сетей* (ИНС) позволяют успешно решать задачи распознавания образов, прогнозирования, оптимизации и управления сложными динамическими системами, и постепенно вытесняют традиционные методы, не обладающие достаточной гибкостью [95]. Применение ИНС обусловлено как высоким быстродействием, устойчивостью к шумам, так и адаптацией к изменениям окружающей среды. Но в случае с ИНС под адаптацией понимают возможность априорного обучения ИНС действовать по прецедентам, имеющимся в обучающей выборке. Адаптация (обучение) ИНС производится, как правило, а priori - до начала использования системы управления в рабочем режиме. В процессе работы ИНС уже не может дообучаться или переобучаться без угрозы «катастрофического забывания» результатов предварительного обучения. Поэтому адаптацией это

можно называть только условно, в ограниченном смысле. В этом один из важнейших недостатков ИНС.

Элементарной единицей ИНС является нейрон, представляющий собой упрощенную модель естественного нейрона. Нейроны связаны между собой синапсами, характеризующиеся числовым параметром в диапазоне  $[0,1]$  или  $[-1,1]$ , называемым весом. С математической точки зрения, искусственный нейрон является взвешенным сумматором всех входящих сигналов  $x$  с учетом соответствующих весов  $w$ , применяющим к полученной сумме некоторую простую функцию  $F(S)$  (рис. 1.3.4.).

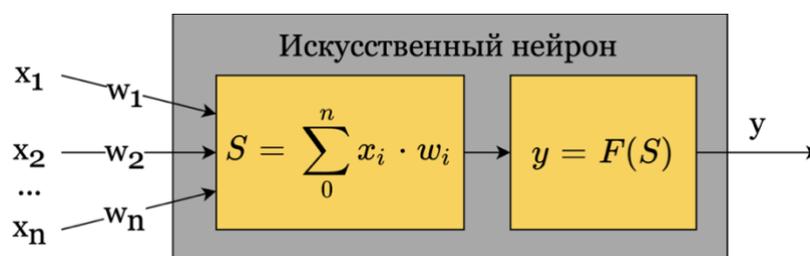


Рис. 1.3.4. Структурная схема искусственного нейрона.

Входные и выходной сигналы нейрона связаны между собой следующей зависимостью:

$$y = F\left(\sum_0^N x_i \cdot w_i\right), \quad (6)$$

где  $x_i$  –  $i$ -ый входной сигнал нейрона;

$w_i$  –  $i$ -ый вес входного сигнала нейрона;

$F(S)$  – функция активации (функция возбуждения), вычисляющая выходной сигнал нейрона.

Основными типами функций активации, использующимися на настоящий момент на практике, являются:

- Пороговая функция, принимающая значения 0 или 1, в зависимости от некоторого уровня  $T$ :

$$y(s) = \begin{cases} 1 & \text{если } s \geq T \\ 0 & \text{если } s < T \end{cases} \quad (7)$$

- Логистическая функция, принимающая значения в диапазоне  $[0;1]$ :

$$y(s) = \frac{1}{1 + e^{-\alpha s}}, \quad (8)$$

где  $\alpha$  – параметр наклона

- Функция гиперболического тангенса, принимающая значения в интервале  $[-1;1]$ :

$$y(s) = th(\alpha s) = \frac{e^{\alpha s} - e^{-\alpha s}}{e^{\alpha s} + e^{-\alpha s}}, \quad (9)$$

где  $\alpha$  – параметр наклона.

Нейроны делятся на три основных типа: входные, скрытые и выходные. В случае, когда ИНС состоит из большого количества нейронов, вводится понятие слоя. На рис. 1.3.5. приведен пример многослойной ИНС, имеющей входной (показан оранжевым цветом), выходной (показан синим цветом) и два скрытых слоя (показаны серым цветом).

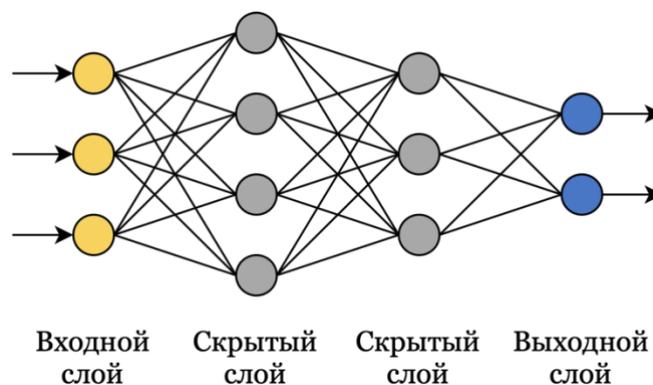


Рис. 1.3.5. Пример многослойной искусственной нейронной сети.

Настройка, или так называемое «обучение» ИНС происходит чаще всего тоже не в реальном времени управления, а заранее (обучение с учителем) и предполагает наличие «обучающей выборки» - пар входных сигналов и требуемого учителем отклика. Получая входной вектор, нейронная сеть выдает результат, который сравнивается с желаемым откликом. Затем вычисляется ошибка между полученным и желаемым откликами, и на основании некоторого алгоритма корректируются весовые коэффициенты ИНС.

Обучение ИНС с учителем требует наличия заранее известной обучающей выборки, а также выбора топологии ИНС. Выбор топологии ИНС является нетривиальной задачей, требующей от ее проектировщика зачастую нестандартного мышления и чутья, поэтому в настоящее время активно развиваются механизмы самоорганизующихся нейронных сетей с возрастающей топологией, которые используют генетические алгоритмы для определения лучшей и минимально необходимой топологии нейронной сети, а также ее параметров.

Впервые использование генетического алгоритма для настройки весов ИНС с заранее определенной топологией предложено в 1994 году в работе [42]. Дальнейшее развитие данного подхода привело к созданию различных вариантов алгоритмов, отличающихся друг от друга используемым эволюционным методом и выбором генетического представления (Таблица 1.3.1).

Следует отметить, что ИНС по своей природе – это системы распознавания образов (либо аппроксимации функций), обучаемые с учителем, а не собственно системы управления. Задача же управления не сводится только к задаче распознавания. Принимаемые решения по управлению зависят от распознанной ситуации, но не только от нее. Особенно при адаптивном управлении, где нужно научиться не только распознавать образы текущих состояний, но и реагировать на них. Обычно ИНС используют для построения простых систем управления, когда заранее известно, что следует делать при распознавании того или иного образа.

Т.е., это управляющие системы, обеспечивающие работу по рефлекторному принципу «стимул – реакция», что является простым частным случаем управления.

Таблица 1.3.1. Некоторые нейроэволюционные методы автоматического создания ИНС.

Метод	Способ кодирования	Используемый эволюционный алгоритм	Эволюционирующие параметры ИНС	Ссылка
Neuro-genetic evolution	Прямой	Генетический алгоритм	Весы ИНС	[42]
NeuroEvolution of Augmenting Topologies (NEAT)	Прямой	Генетический алгоритм	Структура ИНС и ее параметры	[45]
Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT)	Косвенное кодирование	Генетический алгоритм	Параметры ИНС. Изначальная структура фиксирована и полностью связана	[46]

Среди множества известных подходов к организации процессов самообучения и адаптации стоит отметить методы, разработанные представителями отечественной научной школы.

Одним из первых методов адаптации является автоматическая настройка параметров системы управления, предложенная Цыпкиным Я.З. [117]. При этом подходе изменяются только коэффициенты, но не сам закон управления.

Начиная с 60-х годов, ведутся исследования моделей поведения, основанных на конечных автоматах, среди которых можно выделить работы М.Л. Цетлина [116], В.И. Варшавского [62] и А.Н. Рапопорта [102]. Функционирование автоматов данного типа основано на выявлении оптимальных действий в зависимости от сигналов обратной связи – штрафа и поощрения – , получаемых от внешней среды. Находясь в одном из состояний, автомат совершает соответствующее действие  $y_i \in \{y_1, \dots, y_n\}$ . При получении поощрительного сигнала автомат считает совершенное действия правильным и повторит его на следующем шаге. Получая сигнал штрафа, автомат напротив, считает совершенное действие ошибкой и, в

зависимости от глубины памяти автомата, либо совершит то же самое действие, либо сменит его на другое. Число  $n$ , называемое глубиной памяти автомата, задает количество состояний, в которых автомат совершает одно и то же действие. Его смысл заключается в том, что чем больше  $n$ , тем более инерционен автомат и тем большая последовательность сигналов штрафа требуется для смены его действий. Очевидно, что чем больше инерционность автомата, тем ближе он к тому, чтобы выбрав наилучшее в данной среде действие, продолжать выполнять только его. Строго доказано, что при достаточно больших значениях  $n$ , автомат обеспечивает поведение, близкое к наилучшему.

В качестве примера на рис. 1.3.6 приведены диаграммы переходов автомата с линейной тактикой (М.Л. Цетлин) и «доверчивого» автомата (В.И. Кринский) [62].

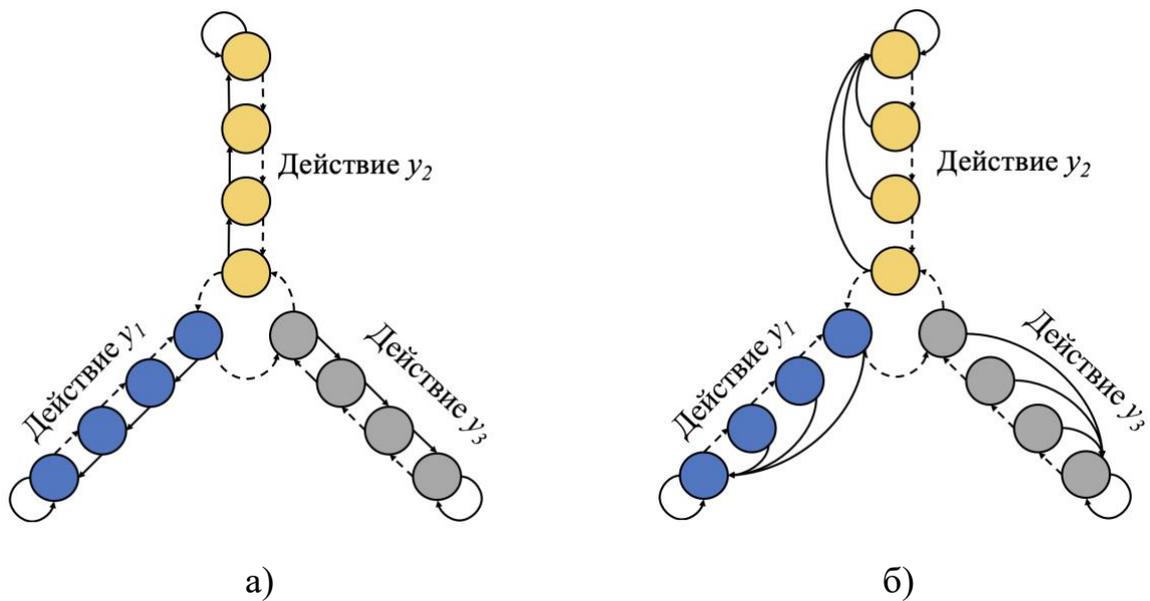


Рис. 1.3.6. Диаграммы переходов а) автомата с линейной тактикой, б) «доверчивого» автомата [62].

Дальнейшее развитие методов самообучения и адаптации в рамках отечественной научной школы представлено в концепции метода Автономного Адаптивного Управления (ААУ) А.А. Жданова. Приведем его краткое описание. На рис.1.3.7 показана общая структура системы, в которой, в отличие от

традиционного подхода, принятого в теории управления, управляющая система (УС) является частью объекта управления ОУ. Исходя из этого, УС одновременно осуществляет две функции: управление и самообучение [67, 68]. Поскольку система ААУ является биологически инспирированной, то в ней присутствует некоторая системы целеполагания (мотивации) и качественной оценки совершаемых действий, а также состояний, в которых она может находиться. В биологических системах такую роль играет «Аппарат эмоций» [67]. Дальше по тексту мы будем называть такие качественные оценки – «эмоциональными».

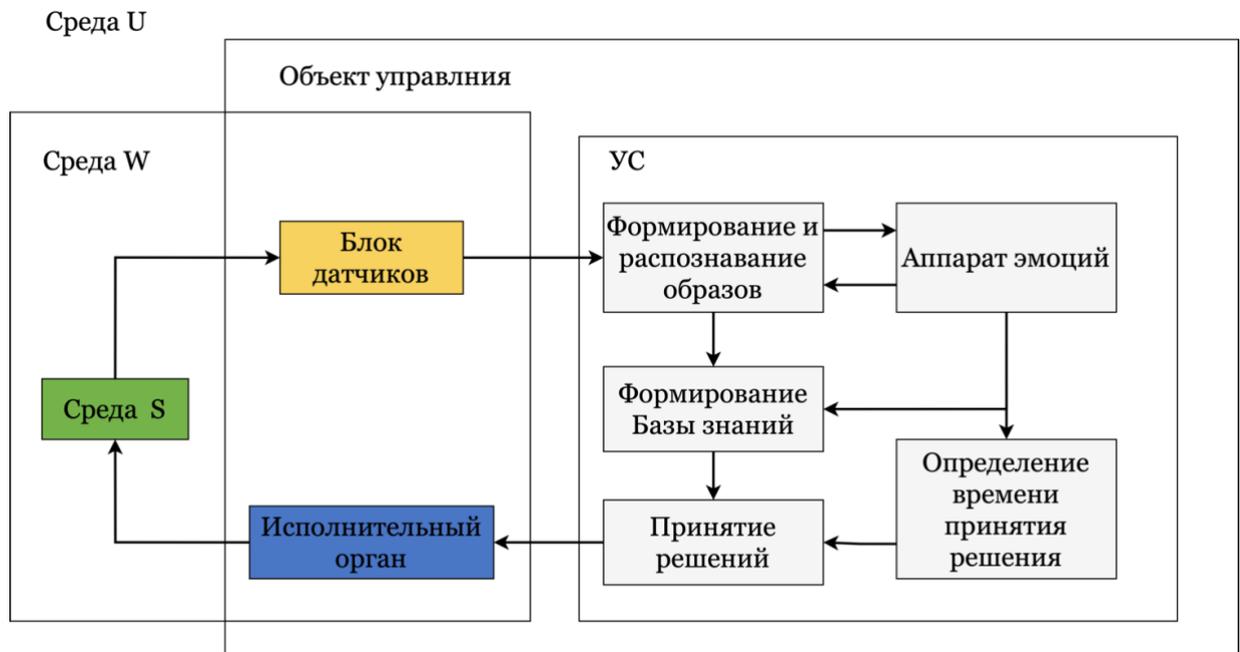


Рис. 1.3.7. Общая структура системы ААУ [67].

Подсистема «Формирования и Распознавания образов» (ФРО), получая данные от датчиков, решает задачи самообучаемой системы распознавания. В «Базе знаний» накапливаются эмпирические знания о возможностях перехода между распознаваемыми (в терминах образов) состояниями системы. «Аппарат эмоций» выполняет 4 важные функции:

1. Определяет качественные («эмоциональные») оценки образов.
2. Определяет качество текущего состояния через сумму «эмоциональных» оценок.

3. Задаёт универсальную целевую функцию – переход в состояние с максимальной из возможных в текущем состоянии «эмоциональных» оценок.
4. Определяет внутреннее время системы управления.

Подсистема «Принятия решений» выбирает лучшее из действий, которое, согласно текущему содержанию «Базы знаний», и доступному времени, переведет систему в состояние с наилучшими «эмоциональными» оценками.

Принятое решение о действии сообщается «Исполнительным органам» и запоминается для дальнейшего анализа.

Одновременно происходит накопление новой информации для дообучения или переобучения системы распознавания, «Базы Знаний» и «Аппарата эмоций». Подсистема «Формирования Базы Знаний» выявляет новые знания о причинно-следственных связях «образов» и «действий». Управление повторяется циклически с заданной постоянной времени.

Современные исследователи в области МР обычно используют эволюционные оптимизационные алгоритмы (ГА), комбинируя их с традиционными алгоритмами управления [3,14,24,49], рассмотренными в предыдущем параграфе. С одной стороны, сложности такого подхода связаны с длительными вычислениями, моделированием «популяции» МР и оценкой каждой из «особей» функцией пригодности. С другой стороны, эти подходы применяются на этапе разработки системы управления для ее оптимизации, а работа в режиме реального времени в этих случаях практически невозможна.

Метод ААУ показал хорошие результаты применительно к различным объектам управления, как при моделировании, так и в реальных условиях на физических прототипах [67-73]. Его отличительной особенностью, по сравнению с нейронными сетями, эволюционными алгоритмами и другими методами адаптации, является возможность обучения и переобучения непосредственно в процессе функционирования объекта управления.

Настоящая работа посвящена развитию метода ААУ путем его применения к управлению МР, что в конечном счете, подчинено задаче повышения качества и эффективности функционирования МР и развитию систем управления многозвенными конструкциями. Применение метода ААУ непосредственно связано с разработкой нового способа построения адаптивной системы управления МР.

### **1.3. Программные средства моделирования модульных роботов**

Программное обеспечение для моделирования является необходимым инструментом для разработки и программирования систем управления МР (в том числе и адаптивных систем), позволяющих разрабатывать, отлаживать и верифицировать алгоритмы управления, эмулируя физические свойства робота и окружающей среды, а также проводить обучение робота.

В настоящее время можно выделить два направления разработки программных комплексов для моделирования МР. Первое предполагает использование робототехнических симуляторов общего назначения, например Webots [34,44], Gazebo [33] или V-Rep [8]. Другой подход связан с разработкой отдельными исследовательскими группами собственных программно-инструментальных средств на базе различных игровых движков (Unity [22], Delta3D [53], JME [4]) или гибрида графических движков (OGRE3D) и библиотек физической симуляции (PhysX, ODE) [6,35, 50, 51].

Робототехнические симуляторы общего назначения, основные характеристики которых приведены в табл. 1.3.1, обладают широкими возможностями по моделированию роботов, а также программированию их поведенческих, моторно-двигательных и сенсорных функций. Предоставляемые опции позволяют создавать программные модели роботов и среды их

функционирования на основе встроенных библиотек типовых элементов. Кроме того, существуют встроенные специализированные программы для решения кинематических задач, определению расстояний, обнаружению столкновений, планированию маршрутов, построению 2D и 3D графиков, возможность программирования контроллеров, управление реальными образцами роботов и т.д. Однако анализ основных характеристик робототехнических симуляторов общего назначения показывает, что при всем многообразии их возможностей, для моделирования модульных робототехнических систем необходима разработка специализированных программных модулей.

Таблица 1.3.1. Основные характеристики робототехнических симуляторов  
общего назначения.

Название	Поддерживаемый язык программирования	Физический движок	Операционная система	Лицензия	Отличительные особенности
Gazebo	C++	ODE / Bullet	Linux	Свободная Apache 2.0	Интеграция с ROS
V-Rep	Lua	ODE / Bullet / Vortex / Newton	Windows, Linux, MacOS	Платная	Интеграция с ROS. API для программирования на C/C++/ Python/ Java/ Matlab
Webots	C++	Основанный на ODE	Windows, Linux, MacOS	Свободная MIT	Интеграция с ROS. API для программирования на C/C++/ Python/ Java/ Matlab
Microsoft Robotics Studio	C#	PhysX	Windows	Свободная	—
RoboDK	Python	—	Windows, Linux, MacOS, Android	Платная	API для программирования на C# и Matlab
MORSE	Python	Bullet	Linux, MacOS	Свободная BSD	—
OpenRave	C++/Python	ODE/Bullet	Windows, Linux, MacOS	Свободная LGPL	API для программирования на C/C++/ Python/ Matlab

При отказе от симуляторов общего назначения разработчики реконфигурируемых роботов вынуждены обращаться к созданию собственных средств моделирования, выбирая удобные для себя языки программирования и типы используемых движков. В таблице 1.3.2 приведены основные характеристики наиболее популярных движков и библиотек, обеспечивающих моделирование физических закономерностей в программных средах.

Таблица 1.3.2. Основные характеристики библиотек физической симуляции.

Название	Язык разработки	Поддерживаемые операционные системы	Лицензия	Отличительные особенности
PhysX	C++	Windows, Linux, MacOS, Android, IOS	Платная	Аппаратная поддержка для ускорения физических расчетов
Bullet Physics Library (Bullet)	C / C++	Windows, Linux, MacOS, Android, IOS	Свободная zlib	Механизм обнаружения столкновений для объектов с выпуклой оболочкой
Open Dynamics Engine	C / C++	Windows, Linux, MacOS	Свободная BSD и LGPL	Предпочтение скорости и стабильности в ущерб физической точности
Newton Dynamics	C++	Windows, Linux, MacOS	Свободная zlib	Обработка тел с высоким отношением масс (до 400:1)

В качестве основы для создания средств моделирования МР достаточно часто выбираются игровые движки, которые позволяют быстро и эффективно разрабатывать приложения с трехмерной графикой и физической симуляцией. При этом удобство и простота моделирования обеспечиваются за счет высокоуровневых языков программирования, а также наличия встроенных компонентов для трехмерной визуализации и симуляции физических свойств и закономерностей. Сравнительные характеристики современных игровых движков приведены в табл. 1.3.3.

Основные различия специализированных симуляторов МР заключаются в используемых физических движках, наличии (или отсутствии) библиотек трехмерных моделей типовых модулей, степенью детализации моделируемых конструкций, а также дополнительными программными компонентами.

Многочисленные примеры из мировой практики (рис. 1.3.1) позволяют утверждать, что робототехнические симуляторы общего назначения и специализированные программные комплексы собственной разработки используются приблизительно в равной мере в задачах моделирования МР. Результаты сравнительного анализа современных средств моделирования мехатронно-модульных реконфигурируемых роботов приведены в табл. 1.3.4.

Таблица 1.3.3. Сравнительные характеристики игровых движков.

Название	Язык разработки	Физический движок	Поддерживаемые операционные системы	Лицензия
Unity3D	C#/Javascript	PhysX	Windows, Linux, MacOS	Свободная / платная
Unreal Engine	C++	PhysX	Windows, Linux, MacOS	Платная
Torque	TorqueScript	PhysX	Windows, Linux, MacOS	Свободная MIT
Delta3D	Python	Bullet/ODE	Windows, Linux, MacOS	Свободная LGPL
Blender Game Engine	Python	Bullet	Windows, Linux, MacOS	Свободная GNU
Jmonkey Engine	Java	Bullet	Windows, Linux, MacOS	Свободная BSD

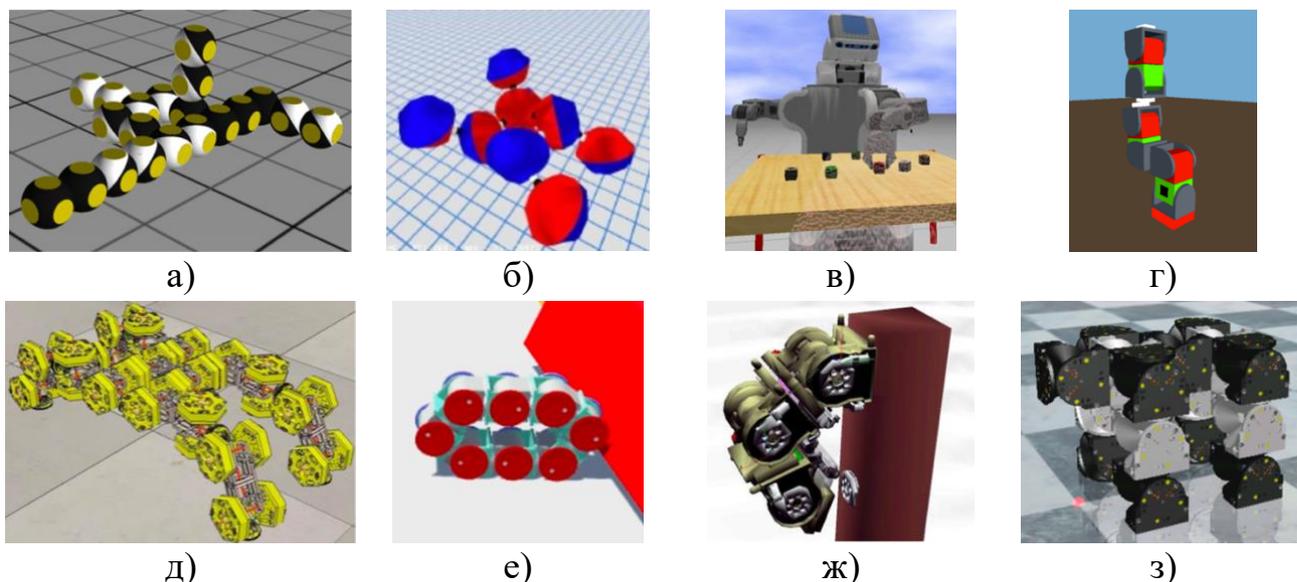


Рис. 1.3.1. Примеры моделирования модульных роботов в различных средах: а) Roombots (Webots) [34,44]; б) ATRON (USSR) [4]; в) Cubelets (Gazebo) [33]; г) Superbot (ReMod3D) [6]; д) ULGEN (V-Rep)[8]; е) SMORES-EP (VSPARC) [22]; ж) SYMBRION/REPLICATOR (Robot3D)[53]; з) Ubot (UbotSim) [57].

Таблица 1.3.4. Сравнительные характеристики средств моделирования модульных роботов.

ПО	Среда разработки	Физический движок	Типовой мехатронный модуль	Возможность добавления новой модели модуля	Средства описания конфигурации и модульного робота	Возможность добавления новых программных компонентов	Лицензия
Webots	-	ODE	Нет	Да	На усмотрение разработчика	Да	Свободная MIT
V-Rep	-	ODE / Bullet / Vortex / Newton	Нет	Да	На усмотрение разработчика	Да	Платная
Gazebo	-	ODE/Bullet	Нет	Да	На усмотрение разработчика	Да	Свободная Apache 2.0
USSR	Jmonkey	ODE	ATRON, MTRAN, Odin, CkBot	Да	XML	Нет	-
ReMod3D	-	PhysX	SuperBot, ATRON, Smores, Roombots	Нет	-	Да (C++)	-
VSPARC	Unity	PhysX	SMORES-EP	Нет	XML	Нет	-
Robot3D	Delta3D	ODE	Backbone, Scout, Active Wheel	Нет	XML	Нет	-
UbotSim	-	PhysX	Ubot	Нет	XML	Нет	-

Обобщение итогов проведенного аналитического обзора средств моделирования МР позволяет сделать следующие основные выводы:

- использование робототехнических симуляторов общего назначения для моделирования МР предполагает необходимость разработки дополнительных программных компонентов;

- средства моделирования собственной разработки, как правило, создавались для проведения тех исследований, которыми занималась конкретная научная группа. В частности, они ориентированы на конкретную конструкцию типового модуля и на алгоритмы управления, рассмотренные в параграфе 1.2 и др.

- исследования в области адаптивного управления МР требуют разработки нового способа построения программного обеспечения, с учетом дальнейших перспектив исследований как в области адаптивного управления, так и в области МР. Это подразумевает, во-первых, выработку архитектуры системы управления, во-вторых – организацию взаимодействия моделирующего комплекса непосредственно с физическим образцом МР, в-третьих – применение метода Автономного Адаптивного Управления для управления модульными роботами.

### **Выводы по главе**

В настоящей главе рассмотрены традиционные алгоритмы управления модульными роботами, которые в подавляющем своем большинстве не могут дообучаться или переобучаться непосредственно в процессе управления, поэтому они не имеют возможности «на ходу» приспособливаться к изменениям свойств среды или робота и в результате принципиально не могут обеспечить высокого качества управления модульными роботами в тех типичных случаях, когда а) математически формализованный закон управления построить трудно или невозможно, либо б) свойства объекта управления, окружающей среды или миссии модульного робота изменяются мало предсказуемым способом.

Также были рассмотрены методы и алгоритмы искусственного интеллекта и самообучения. В этом классе алгоритмов наибольший интерес представляет метод Автономного Адаптивного Управления А.А. Жданова, позволяющий повысить функциональные возможности модульных роботов, а именно, возможности адаптации и самообучения непосредственно в процессе функционирования. Применение метода Автономного Адаптивного Управления неразрывно связано с разработкой способа построения адаптивной системы управления модульными роботами.

Кроме того, были рассмотрены программные средства программного моделирования, анализ которых показал необходимость разработки нового способа построения программного обеспечения моделирования модульных роботов, включая архитектуру программной системы, организацию взаимодействия моделирующего комплекса непосредственно с физическим образцом модульного робота, а также применение в основе алгоритмов управления метода Автономного Адаптивного Управления.

## **ГЛАВА 2. АДАПТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ МОДУЛЬНЫМИ РОБОТАМИ**

В данной главе предложен способ построения адаптивной системы управления модульными роботами, разработана математическая модель, упрощенно описывающая работу системы, построенной по методу Автономного Адаптивного Управления. Также описаны предложенные алгоритмы адаптивной системы управления. Программная реализация и результаты экспериментальных исследований будут изложены в следующих главах.

### **2.1. Способ построения адаптивной системы управления модульными роботами**

Современные системы управления интеллектуальными роботами в подавляющем большинстве случаев организованы иерархически. Это обуславливается не столько сложностью конструкции самого робота, сколько сложностью и многообразием совершаемых им целенаправленных действий. Смысл иерархической системы заключается в функциональном разделении задач между уровнями иерархии. Так, например, в большинстве случаев на самом низком уровне решаются задачи управления исполнительными механизмами. На следующем уровне решается в какой последовательности ими управлять. На более высоких уровнях решаются задачи построения модели среды, человеко-машинного взаимодействия и планирования действий. На практике разделение системы управления на уровни позволяет вносить изменения точно на отдельных уровнях, не изменяя систему в целом.

Принимая во внимание функциональные возможности МР, в настоящем диссертационном исследовании мы остановились на классической иерархической системе управления [76, 78, 94]. Она состоит из 3-х уровней:

1. Уровень планирования поведения
2. Уровень построения траекторий движения
3. Приводной уровень

Опираясь на данную 3-х уровневую систему, в данной диссертационной работе предлагается следующий способ построения адаптивной системы управления.

1. На уровне *планирования поведения* МР получает от оператора целевое задание, поставленное в удобной для человека форме, например голосом или в виде текста. На данном уровне осуществляется следующее:
  - 1.1. Определение модульным роботом поставленной оператором задачи, описанной в удобной для человека форме представления информации (голос/текст). Несмотря на широкий круг задач, решаемых современными роботами, все они используют ограниченный набор возможных функциональных действий: манипуляционных и транспортных. Таким образом поставленные задачи можно свести к одной из следующих:
    - манипуляционная задача (захват предмета, его перемещение относительно неподвижного основания робота);
    - транспортная задача (перемещение робота из текущей в требуемую точку пространства);
    - реконфигурационная задача (реконфигурация МР из одной кинематической схемы в другую).
  - 1.2. Выбирается оптимальная конфигурация МР для выполнения поставленного задания. В процессе своего функционирования МР непрерывно взаимодействует со средой. Поэтому одним из ключевых аспектов при выборе той или иной конфигурации является распознавание формы

препятствия [92]. Классификация форм препятствий – это отдельное направление исследований, которое проводилось в привязке к разработке транспортных средств различного типа и назначения. Так, согласно М.Г. Беккеру [60], несмотря на разнообразие препятствий, встречающихся в процессе движения, все они могут быть приведены к одному из двух типов формы – «уступ» (контрэскарп) или «выступ» (эскарп). Они могут встречаться как отдельно, так и в сочетании, и, в последнем случае, образуют новые виды - «канавы» или «насыпи». Классификацию Беккера можно дополнить препятствием типа «арка», характерной особенностью которой является наличие горизонтального блока, ограничивающего конфигурацию модульного робота по высоте. На рис. 2.1.1. приведен внешний вид типовых препятствий.

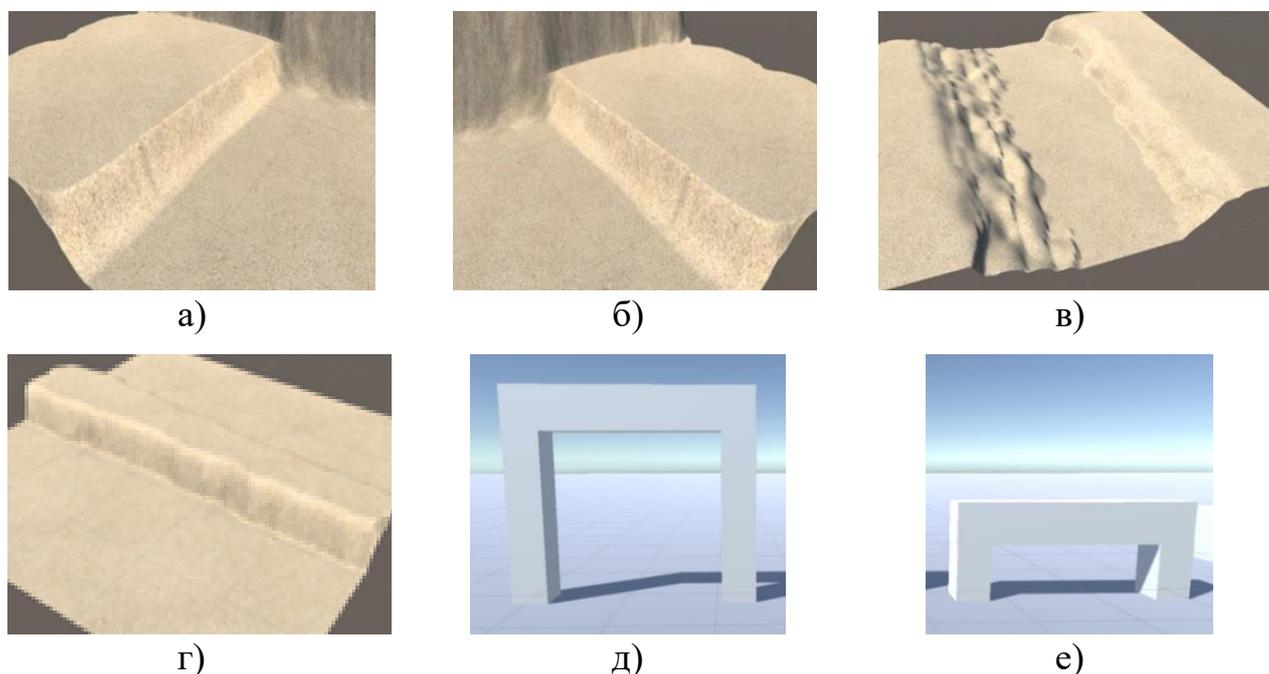


Рис. 2.1.1. Внешний вид типовых препятствий: а) «уступ», б) «выступ», в) «канавы», г) «насыпь», д)-е) «арка».

Все пять типов препятствий могут снижать показатели проходимости модульного робота, а в ряде случаев, при неправильном выборе конфигурации,

вызывать и невозможность их преодоления. В таблице 2.1.1 представлено предполагаемое соответствие классов ситуаций и используемых типов конфигураций.

Кроме того, стоит отметить, что существует также класс непреодолимых препятствий, например таких, как высокие стены, при возникновении которых необходимо перестраивать маршрут движения МР.

Таблица 2.1.1. Соответствие типа задачи и условий внешней среды конфигурации модульного робота.

Тип задачи	Условия внешней среды	Конфигурация модульного робота
Манипуляционная задача. Перемещение объектов	–	«Манипулятор»
Транспортная задача. Перевозка грузов.	–	«Шагающая платформа»
	Низкая арка	«Гусеница»
	Высокая арка	
Транспортная задача. Прибытие в точку.	Уступ	«Гусеница», «Колесо» или «Шагающая платформа»
	Выступ	
	Канавы	
	Насыпь	

1.3. Формируется план действий, приводящий к успешному выполнению поставленной задачи.

Стоит отметить, что данный уровень подразумевает проведение моделирования функционирования робота на борту, за счет специальных программных средств для моделирования МР, о которых пойдет речь в следующей главе).

2. На уровне *построения траекторий движения* адаптивно формируется закон изменения обобщенных координат МР, для достижения целевой точки с учетом особенностей внешней среды. Адаптация достигается за счет применения метода ААУ. В данной диссертационной работе главный акцент сделан на данном уровне. Алгоритмы, лежащие в его основе, будут рассмотрены в следующем параграфе.
3. *Приводной* уровень осуществляет управление исполнительными механизмами роботами, обеспечивая реализацию элементарных траекторий движения. Адаптация на этом уровне предполагает изменение параметров или закона управления в ответ на внешние возмущения, например, изменение температурных характеристик или нагрузки на валу двигателя. Приводной уровень системы управления робототехническими системами на сегодняшний день является самым проработанным. Для качественного управления исполнительными механизмами часто используют классические решения, основанные на ПИД или LQR регуляторах [81, 82]. Активно используются и интеллектуальные методы, основанные на нечеткой логике [76,78,109,110]. Метода ААУ примененный к управлению исполнительными механизмами рассмотрен в работах [72,73] и показывает хорошие результаты. Поскольку этот уровень достаточно проработан, в диссертационном исследовании мы не будем специально на нем останавливаться и будем использовать простую наиболее подходящую для нашего случая модель, основанную на гармоническом осцилляторе (см. параграф 1.2 настоящей работы).

Первые 2 уровня должны располагаться на борту одного из мехатронных модулей робота (для удобства постановки экспериментов эти уровни могут располагаться на внешнем вычислительном устройстве). Будем называть такой модуль главным. В зависимости от ситуации, функции главного модуля могут быть переданы другому модулю. В работе [88] рассматриваются возможности передачи

функций главного модуля. Для возможности передачи функций главного модуля другому, в каждый модуль записывается одно и то же программное обеспечение. Исполнительный уровень, учитывая конструктивные особенности МР, является физически распределенным.

Ключевым отличием предлагаемой системы управления является введение механизма адаптации на основе метода ААУ на уровне *построения траекторий движения*. Таким образом, структуру адаптивной системы управления МР можно представить в виде следующей схемы (рис. 2.1.2).

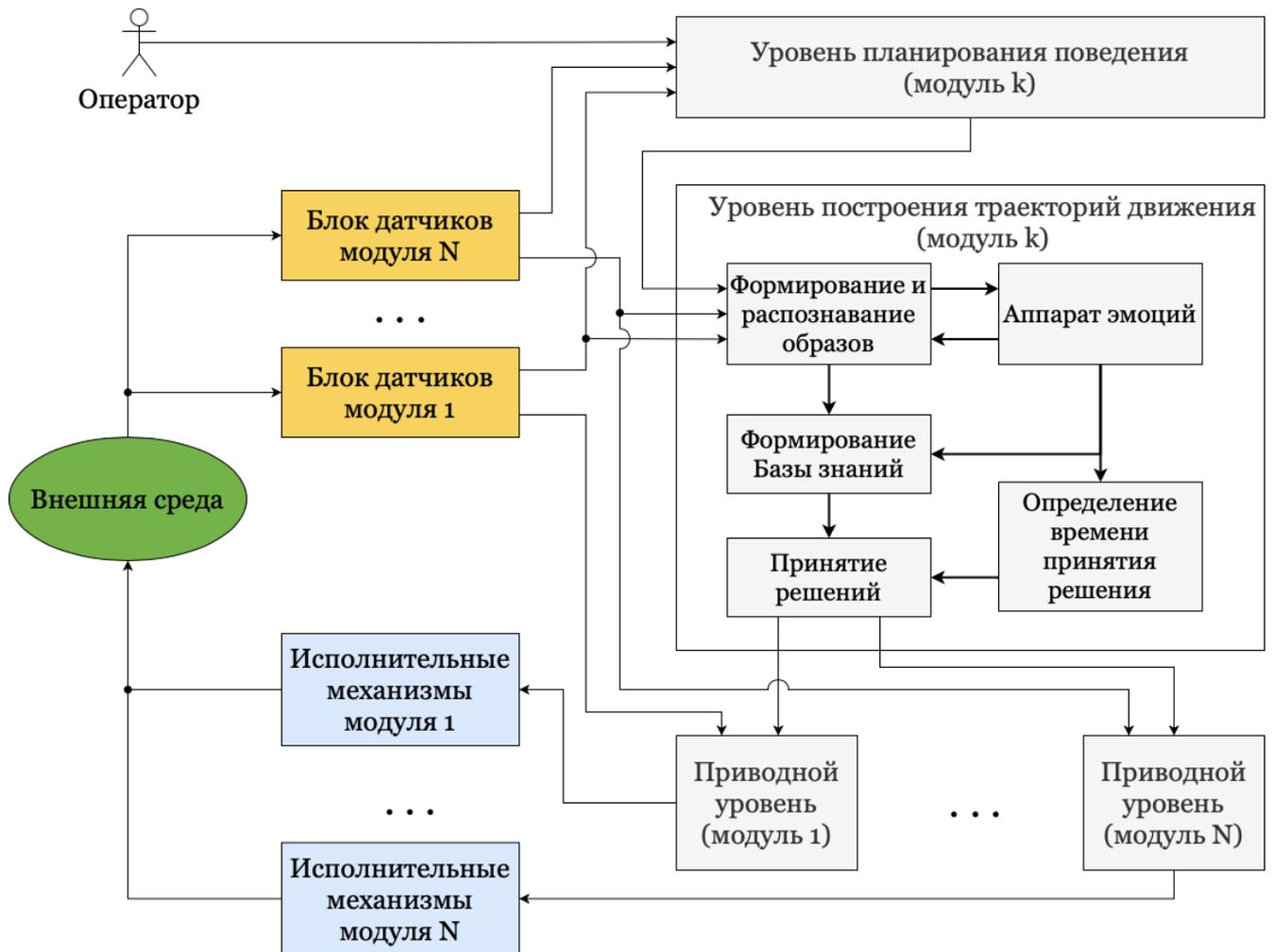


Рис. 2.1.2. Структура адаптивной системы управления модульными роботами.

Как видно из схемы (рис. 2.1.2), уровень планирования поведения и уровень построения траекторий движения получают сенсорные данные с каждого из модулей. Исполнительный уровень каждого из модулей является независимым, поэтому получает сенсорные данные с блока датчиков только того модуля, к которому он относится.

Согласно методу ААУ, система управления имеет 2 контура: контур управления и контур обучения. Такт функционирования системы управления состоит из следующих операций:

В контуре управления:

- чтение данных с блока датчиков;
- распознавание образов, на основе полученной сенсорной информации;
- оценка «аппаратом эмоций» распознанных образов;
- расчет текущей общей «эмоциональной» оценки
- принятие решения о действии на основе распознанных и сформированных образов, общей «эмоциональной» оценки, её производной и базы знаний.

В контуре обучения:

- формирование новых образов, на основе полученной сенсорной информации
- качественная оценка «аппаратом эмоций» новых, сформированных образов;
- формирование новых знаний, на основе распознанных и сформированных образов и совершенных на предыдущем шаге действий.

Исходя из перечисленных операций, ключевыми моментами для применения метода ААУ к управлению МР является определение:

- набора сенсоров, входящих в «блок датчиков»;
- способа формирования и распознавания образов в блоке «ФРО»;
- способа расчёта «эмоциональной» оценки «аппаратом эмоций»;
- формы представления знаний в «базе знаний»;

- набора возможных действий исполнительных механизмов.

Коротко сформулируем основные положения предложенного способа построения адаптивной системы управления МР:

1. Адаптивная система управления должна строиться иерархически (для функционального разделения задач между уровнями иерархии) и содержать 3 уровня – уровень планирования поведения, уровень построения траекторий движения, приводной уровень.
2. Верхние 2 уровня (уровень планирования поведения и уровень построения траекторий движения) должны располагаться на борту одного их мехатронных модулей.
3. Адаптивность системы управления должна обеспечиваться за счет использования метода ААУ.
4. На каждом из типовых модулей стоит одно и то же программное обеспечение.
5. Система управления должна включать в свой состав средства программного моделирования для проведения обучения или переобучения без участия реального физического робота.

## **2.2. Математическая модель метода автономного адаптивного управления**

В данном параграфе рассмотрим разработанную в диссертационном исследовании математическую модель, упрощенно описывающую работу метода ААУ. Предлагаемая модель, далее будем называть ее «Самообучающийся Динамический Конечный Автомат» (СДКА), основывается на теории конечных

автоматов и расширяет ее. Принципиальные отличия от традиционных КА – введение динамической функции выходов и функции формирования новых состояний, а также замыкание выходов автомата на его вход.

СДКА описывается кортежем:

$$(S, A, D, F, \psi, E, s_0, f, d) \quad (10)$$

где

$S$  – конечное пополняемое множество состояний;

$A$  – совмещенное конечное множество входных и выходных символов (множество действий);

$D$  – динамическая функция выходов;

$F$  – функция формирования новых состояний;

$\psi$  – функция переходов;

$E$  – функция качественной оценки состояний

$f$  – коэффициент самообучения,

$d$  – коэффициент глубины поиска решения

$s_0$  – начальное состояние автомата  $s_0 \in S$ .

Предложенная математическая модель соотносится с блоками системы ААУ следующим образом, как показано в таблице 2.2.1.

Таблица 2.2.1. Интерпретация СДКА в терминах метода ААУ.

Параметр	Описание в терминах СДКА	Описание в терминах метод ААУ
$S$	Конечное пополняемое множество состояний	Память распознанных образов
$A$	Совмещенное конечное множество входных и выходных символов (множество действий)	Множество возможных действий, которые может совершить система

D	Динамическая функция выходов	Аппарат эмоций и блок принятия решений о действии
$\psi$	Функция переходов	Блок формирования новых знаний
F	Функция формирования новых состояний	Блок формирования новых образов

Важно отметить, что выход СДКА замкнут на вход, т.е. символ  $a_i \in A$ , продуцированный на выход, поступает на вход автомата и провоцирует переход в новое состояние и новый выходной символ. Данный процесс описывается законами функционирования СДКА. Вход  $a_t \in A$  в момент времени  $t$  однозначно определяется предыдущем состоянием:

$$a_t = D(s(t-1), f) \quad (11)$$

где  $D$  – динамическая функция переходов, которая определяет выход автомата на основе предыдущего состояния на основе многокритериального выбора:

1. Если отношение мощности множества  $B \subseteq A$  (отражающее известные возможные переходы из состояния  $s(t-1)$ ) к мощности множества  $A$  больше  $f$ , то  $a_t$  выбирается путем равновероятностного выбора из множества  $C = A \setminus B$ .
2. В противном случае  $a_t$  выбирается таким, что

$$E(s(t+d)) \rightarrow \max(E(S)),$$

где  $E$  – многокритериальная функция качественной оценки состояний.

Новое состояние  $s(t)$  в момент времени  $t$  однозначно определяется состоянием  $s(t-1)$  и входным символом  $a_t$ :

$$s(t) = \psi(s(t-1), a_{t-1}), \quad (12)$$

где  $\psi$  – функция переходов;

**Определение 1.** Состояние  $p$  ( $p \in S$ ) называется устойчивым, для  $\forall s (s \in S) \exists$  последовательность  $(a_n)$  такая, что осуществляется переход в состояние  $p$ . Множество устойчивых состояний будем обозначать  $P$  ( $P \subseteq S$ )

**Теорема 1 (теорема сходимости).** Существует последовательность входных символов  $(a_1, a_2, \dots, a_n)$  такая, что СДКА переходит в устойчивое состояние  $p_b$  такое, что  $E(p_b) = \max(E(P))$ .

*Доказательство.* Пусть в некоторый момент времени  $t$  автомат находится в состоянии  $s(t)$ . Рассмотрим динамическую функцию переходов  $D$ . Согласно определению 1, из состояния  $s(t)$  может быть осуществлен переход в любое из состояний  $p \in P$ . Тогда очевидно, что  $\exists d$  такое, что  $E(s(t + d)) = \max(E(P))$  по определению работы динамической функции переходов. При этом  $s(t + d) \in P$ . Что и требовалось доказать.

В следующем параграфе рассмотрим пример применения предложенной математической модели к описанию одного из способов применения метода ААУ к управлению МР.

### **2.3. Алгоритмы адаптивной системы управления движением модульных роботов в конфигурациях «манипулятор» и «шагающая платформа»**

В данном параграфе рассмотрим подробнее уровень построения траекторий движения адаптивной системы управления. На этом уровне адаптивно формируется закон изменения обобщенных координат для достижения целевой точки. Наименее разработанными конфигурациями МР являются «манипулятор» и

«шагающая платформа», поэтому в рамках диссертационного исследования сделан акцент именно на этих конфигурациях.

Задачу по нахождению закона изменения обобщенных координат для достижения целевой точки можно формализовать следующим образом. Начальная конфигурация  $C_0$  модульного манипуляционного робота задана множеством обобщенных координат, однозначно определяющих его положение в пространстве:

$$C_0 = \{q_1, q_2, \dots, q_n\}, \quad (13)$$

где  $n$  – количество модулей, входящих в состав модульного манипуляционного робота.

Цель задана координатами в трехмерном пространстве  $(x_u, y_u, z_u)$ , относительно локальной системы координат модульного робота. Необходимо найти такую последовательность конфигураций  $\{C_0, \dots, C_k\}$ , что:

$$\sqrt{(x_k - x_u)^2 + (y_k - y_u)^2 + (z_k - z_u)^2} \rightarrow \min, \quad (14)$$

где  $(x_k, y_k, z_k)$  – координаты конца кинематической цепи, определяемой конфигурацией  $C_k$ .

В диссертационной работе мы предлагаем два способа решения данной задачи, каждый из которых основан на методе ААУ. Блок датчиков должен содержать сенсоры углового положения каждого из модулей робота, датчики, позволяющие оценить окружающую обстановку (расстояния до препятствий и целевой точки, их координаты относительно локальной системы координат робота).

*1-й способ.* В этом случае образцами системы ФРО могут быть собой наборы обобщенных координат  $C = \{q_1, q_2, \dots, q_n\}$ , определяющие текущую

конфигурацию модульного робота. В качестве примера на рис. 2.3.1 приведены различные наборы координат и соответствующие им конфигурации модульного реконфигурируемого робота.

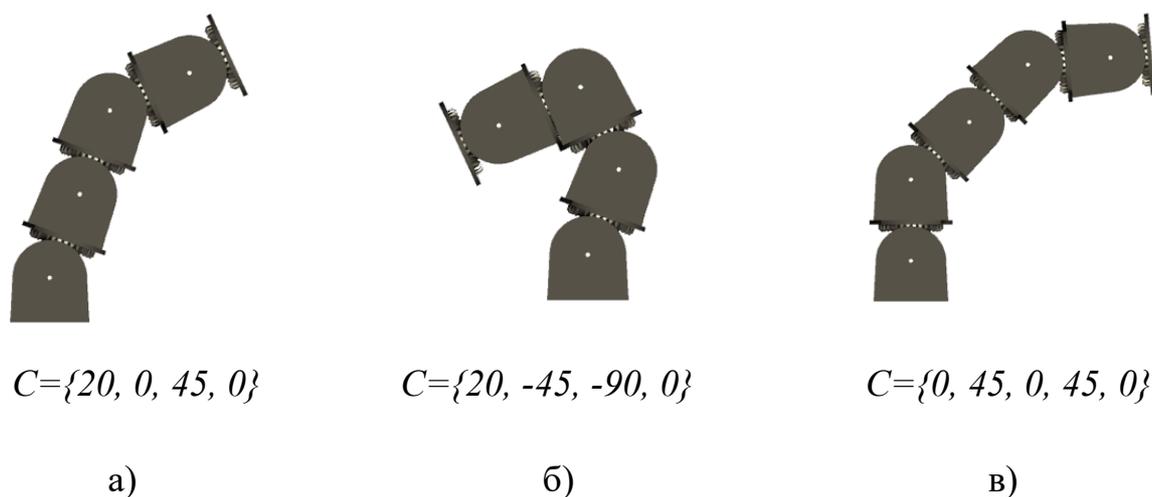


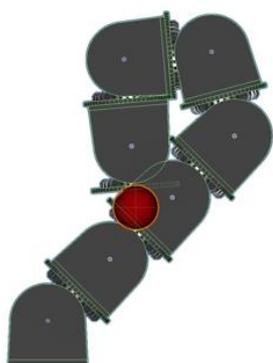
Рис. 2.3.1. Примеры конфигураций манипуляционного модульного робота.

Нормированная «эмоциональная» оценка распознанного образа текущей конфигурации зависит от расстояния  $d$  конца кинематической цепи до целевой точки, размера её  $\varepsilon$ -окрестности и рассчитывается по формуле:

$$E = \begin{cases} \varepsilon/d, & d \geq \varepsilon \\ 1, & d < \varepsilon \end{cases} \quad (15)$$

Значение нормированной «эмоциональной» оценки будет находиться в диапазоне  $[0;1]$ , что позволяет легко перевести ее в любую другую шкалу.

Стоит отметить, что в процессе своего функционирования система может совершить ряд действий, которые приведут к столкновению модулей друг с другом (рис. 2.3.2). В этом случае такой «образ» текущей конфигурации робота имеет минимально возможную «эмоциональную» оценку, даже несмотря на то, что робот отработал целевую точку.



$$C = \{45, 0, 0, -60, -80, -90, 0\}$$

Рис. 2.3.2. Пример отработки точки со столкновением между модулями робота.

Типовыми действиями реконфигурируемого модульного робота являются либо фиксация текущего углового положения  $i$ -го модуля, либо его изменение на величину  $\Delta$  (как в положительном, так и в отрицательном направлении) (16). Таким образом, количество возможных действий робота равно  $3n$ , где  $n$  – число звеньев.

$$q_i \rightarrow \begin{cases} q_i + \Delta \\ q_i - \Delta \\ q_i \end{cases}, \quad (16)$$

Одним из ключевых факторов, влияющих на скорость обучения в методе ААУ, является способ выбора действий. Например, можно перебирать действия одно за другим, или выбирать действия случайно. В работе [68] рассмотрены способы выбора действий с применением генетических алгоритмов и градиентного спуска.

В нашем случае применяется метод случайного выбора действий с возвращением в ситуацию с наилучшей «эмоциональной» оценкой. Адаптивная система управления запоминает наилучшую «эмоциональную» оценку, которая была получена в процессе функционирования системы. В случае если МР находится некоторое время  $\tau$  в ситуации с не самой лучшей «эмоциональной» оценкой, то происходит переход в ситуацию с наилучшей «эмоциональной» оценкой.

Возвращение в такую ситуацию возможно благодаря структуре «образов», которые представляют собой, как было сказано выше, набор обобщенных координат. Таким образом, система может просмотреть весь список образов, выбрать и перейти в образ с наилучшей оценкой, отработав требуемые углы в сочленениях.

Недостатком такого подхода является то, что с уменьшением величины  $\Delta$  количество образов существенно возрастает. При этом большое значение величины  $\Delta$  может не обеспечить требуемой точности.

Данный способ применения метода ААУ к управлению МР можно формализовать, используя описанную в предыдущем параграфе математическую модель СДКА. Применяя модель, установим, что множеству  $S$  соответствуют образы системы ФРО – наборы обобщенных координат робота, множество  $A$ , определяется типовыми действиями робота (16), «эмоциональная оценка» участвует в работе динамической функции выходов  $D$ .

Работа системы происходит следующим образом. В начальный момент времени СДКА находится в состоянии  $s_0$  (рис. 2.3.3а), затем функция  $D$  инициирует некоторое действие  $a_1$ , которое может либо перевести автомат в новое состояние  $s_1$  (рис. 2.3.3б) или оставить в текущем состоянии  $s_0$  (рис. 2.3.3в). Затем процесс выбора действия повторяется, и тем самым происходит автоматическое построение СДКА.

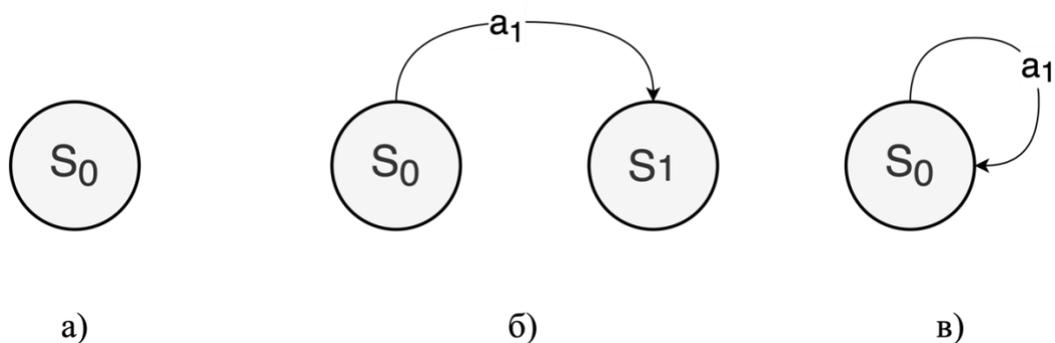


Рис. 2.3.3. Пример работы СДКА применительно к 1-му способу управления методом ААУ.

Динамическая функция выходов  $D$  регулирует работу автомата таким образом, чтобы он оказался в состоянии с наилучшей качественной («эмоциональной») оценкой, а сама структура автомата представляет собой Базу Знаний.

*2-й способ.* В этом случае образами системы ФРО является положение схвата робота относительно целевой точки, которое может описываться комбинацией понятий «левее»/«правее» и «выше»/«ниже» (рис. 2.3.4).

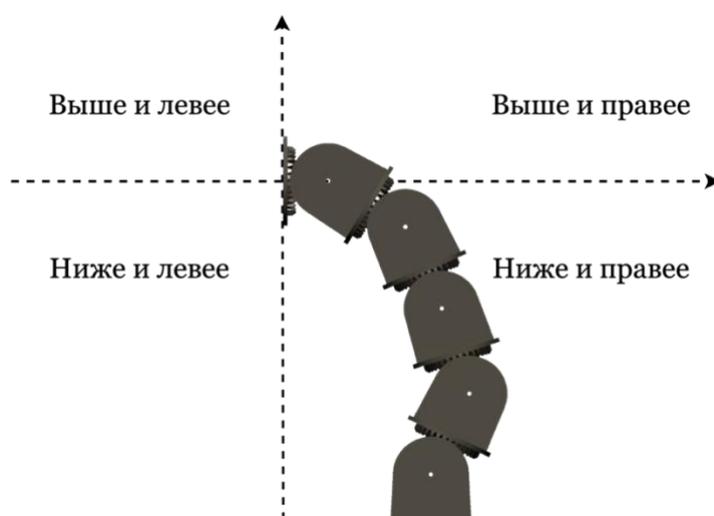


Рис. 2.3.4. Образы системы ФРО, в зависимости от расположения схвата робота относительно целевой точки.

Типовыми действиями в данном случае так же является фиксация текущего углового положения  $i$ -го модуля или изменение его на величину  $\Delta$ . Однако, в отличие от предыдущего способа, значение  $\Delta$  может быть сколь угодно мало и ограничено лишь разрешающей способностью конкретного привода.

Процесс работы циклический и итеративный, т.е. в начале действие совершает 1-й модуль, затем 2-ой, и т.д., а затем процесс повторяется.

«Эмоциональная» оценка основана на изменении рассогласования между схватом и целевой точкой, и рассчитывается следующим образом:

$$E = \begin{cases} 0, & r_T - r_{II} \geq 0 \\ 1, & r_T - r_{II} < 0 \end{cases}, \quad (17)$$

где  $r_T$  - значение рассогласования между концом кинематической цепи и целевой точкой на текущем шаге;

$r_{II}$  - значение рассогласования между концом кинематической цепи и целевой точкой на предыдущем шаге.

Недостатком такого подхода является отсутствие информации в структуре образов о текущей конфигурации манипулятора. Это в свою очередь делает невозможным планирование движения. Кроме того, существуют такие взаимные положения манипулятора и целевой точки, при которых задача прихода в целевую точку не будет решена.

Стоит отметить, что оба таких подхода могут применяться не только к МР в конфигурации «манипулятор», но и в конфигурации «шагающая платформа». Для этого каждую из конечностей шагающего МР можно рассматривать как отдельный манипулятор (рис.2.3.5).



Рис. 2.2.5. Представление конечности шагающей платформы в виде модульного манипулятора.

При этом задача движения сводится к двум подзадачам [90]:

1. задача перестановки отдельной конечности в целевую точку. Целевая точка рассчитывается автоматически на каждом шаге – смещается относительно конца конечности на заданную величину шага  $d$ ;
2. задача последовательной перестановки конечностей, необходимых для передвижения платформы.

Первая подзадача решается одним из изложенных выше способов. Вторая – выбором варианта походки [83, 93], т.е. перестановки конечностей. В качестве допущения будем рассматривать варианты так называемой «осторожной» походки, когда на каждом этапе движения перемещается только одна конечность. В этом случае потеря или сохранение устойчивости шагающей платформы могут рассматриваться в качестве «эмоциональной» оценки.

### **Выводы по главе**

В настоящей главе изложен способ построения адаптивной системы управления модульными роботами. А именно: 1) система должна иметь 3-х уровневую иерархическую структуру, включающую в себя уровень планирования поведения, уровень формирования траекторий движения и приводной уровень; 2) верхние два уровня располагаются на борту одного из мехатронных модулей 3) адаптивность системы обеспечивается за счет применения метода Автономного Адаптивного Управления; 4) программное обеспечение, заранее загружаемое на борт мехатронных модулей должно быть однотипным 5) средства программного моделирования, расположенные на уровне планирования поведения, обеспечивают возможность обучения и переобучения на программных моделях, без риска поломки реального физического образца робота;

Также предложена математическая модель, упрощенно описывающая функционирование системы, построенной по методу автономного адаптивного

управления. Модель, названная «Самообучающийся Динамический Конечный Автомат», опирается на теорию конечных автоматов и расширяет её.

Отличительной особенностью разработанного подхода к построению системы управления является введение механизма адаптации на уровне построений траекторий движения, что позволяет системе управления адаптивно обучаться движению в требуемую точку. Описанные алгоритмы могут применяться для модульных роботов в конфигурациях «манипулятор» и «шагающая платформа». Преимуществами такого подхода являются одновременные процессы управления и обучения, возможность переобучения, отсутствие необходимости предварительного обучения.

Программная реализация и экспериментальные исследования будут даны ниже, в следующих главах.

### **ГЛАВА 3. СРЕДСТВА МОДЕЛИРОВАНИЯ МОДУЛЬНЫХ РОБОТОВ И ИХ ПРОГРАММНОЕ ВЗАИМОДЕЙСТВИЕ С МАКЕТНЫМ ОБРАЗЦОМ МОДУЛЬНОГО РОБОТА**

В данной главе описаны разработанные программные средства, позволяющие моделировать модульных роботов и управлять ими. Предложен способ описания конфигураций модульных роботов и алгоритм автоматического построения программных моделей. Описана реализация метода Автономного Адаптивного Управления. Кроме того, описано взаимодействие программных средств моделирования и разработанного макетного образца модульного робота.

#### **3.1. Способ построения программных средств моделирования модульных роботов**

В 1-ой главе диссертационного исследования был проведен сравнительный анализ программных средств, который показал необходимость разработки нового способа построения программного обеспечения для моделирования МР, с учетом дальнейших перспектив исследований как в области адаптивного управления, так и в области МР.

В рамках диссертационной работы предложен новый способ структурной организации программного комплекса для моделирования МР (рис. 3.1.1). Основными компонентами комплекса являются следующие:

- 3-мерный графический движок и библиотеку физической симуляции для моделирования МР максимально приближенно к реальной среде;
- библиотека типовых мехатронных модулей;

- библиотека алгоритмов управления МР, включая реализацию метода ААУ;
- библиотека дополнительных алгоритмов, включая алгоритмы построения программной модели МР по ее XML описанию.
- библиотека моделей внешней среды;
- API для взаимодействия со сторонними программными комплексами (на основе HTTP протокола);
- драйвер для управления макетным образцом МР (на основе виртуального COM порта).

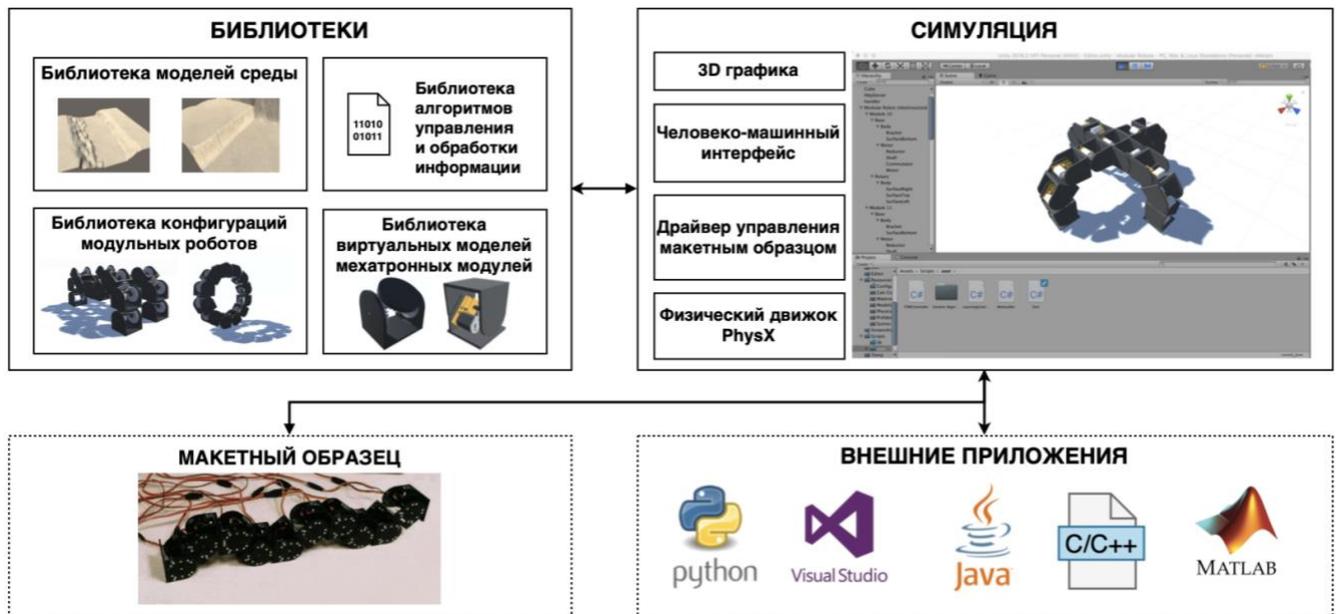


Рис. 3.1.1. Структурная организация программного комплекса для моделирования модульных роботов.

Программный комплекс представляет собой систему классов, взаимосвязь которых представлена на рис. 3.1.2.

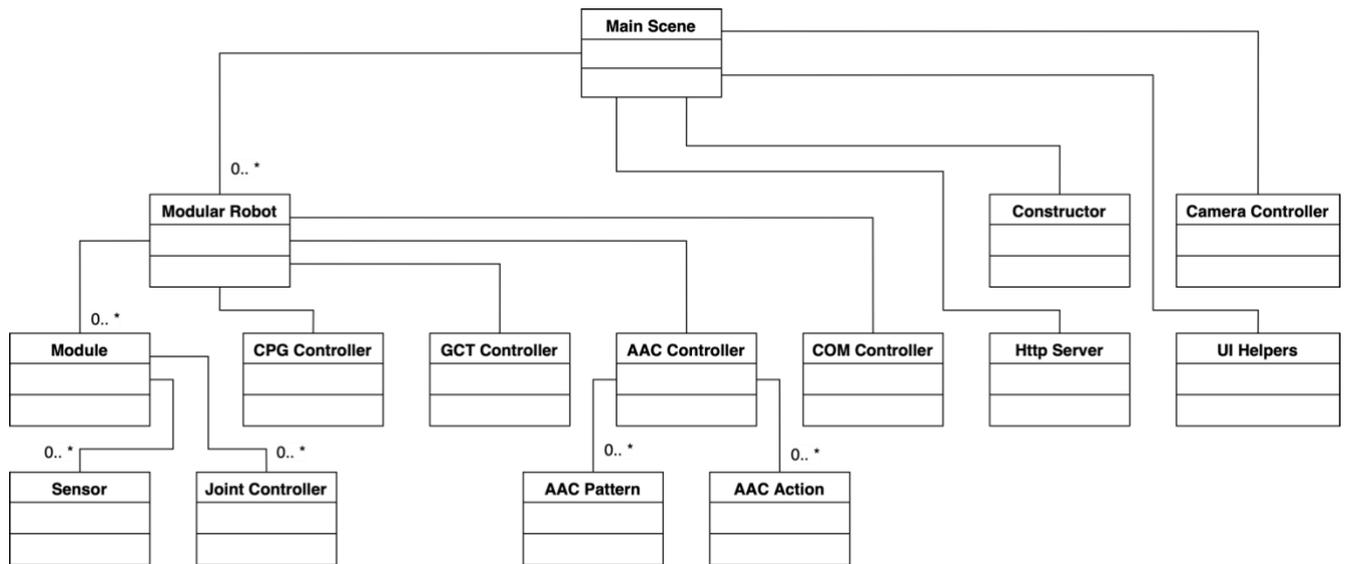


Рис. 3.1.2 UML-диаграмма классов программного комплекса.

- Класс *Modular Robot* представляет собой описание МР и реализует механизмы для управления им.
- Класс *Module* описывает отдельный мехатронный модуль МР. Класс *Modular Robot* содержит в себе список объектов *Module*.
- Класс *Joint Controller* описывает вращательное звено мехатронного модуля. В зависимости от конкретного типа мехатронного модуля, объектов типа *Joint Controller* может быть несколько в классе *Module*.
- Класс *Sensor* описывает датчик мехатронного модуля.
- Класс *GCT Controller* реализует управление МР на основе управляющих таблиц (Gait Control Tables, сокращенно GCT).
- Класс *CPG Controller* реализует управление МР на основе ЦГУА (Central Pattern Generator, сокращенно CPG).
- Класс *AAC Controller* реализует управление на основе метода ААУ (Autonomous Adaptive Control, сокращенно AAC).
- Класс *AAC Pattern* описывает образ системы ААУ. Класс *AAC Controller* содержит список объектов *AAC Pattern*.

- Класс *AAC Action* описывает действие системы ААУ. Класс *AAC Controller* содержит список объектов *AAC Action*.
- Класс *Http Server* содержит реализацию HTTP Сервера для взаимодействия программного комплекса с внешними приложениями.
- Класс *COM Controller* содержит реализацию драйвера для управления макетным образцом МР через виртуальный COM порт. (Макетный образец подключается через USB или Bluetooth).
- Класс *Constructor* реализует механизм построение программной модели в режиме «визуального конструктора».
- Класс *Camera Controller* реализует управление виртуальной камерой в программном комплексе.
- Класс *UI Helpers* содержит дополнительные утилиты, упрощающие работу с пользовательским интерфейсом (User Interface, сокращенно UI) (текстовые поля, кнопки, поля ввода и пр.).
- Класс *Main Scene* реализует интерфейс программы.

В рамках данной диссертационной работы было разработано программное обеспечение для моделирования модульных роботов на основе вышеизложенного способа. Основой разработанного ПО программного комплекса послужила кроссплатформенная среда разработки Unity3D, которая компилирует исходный код, написанный на C# на целевую платформу (Windows, Linux, MacOS). В программном обеспечении используются протоколы HTTP, протокол обмена информацией между макетным образцом и средой, стандартные драйверы целевой ОС, для работы с виртуальным COM-портом. Макетный образец МР подключается к программному обеспечению через USB/Bluetooth.

В программном коде используются стандартные библиотеки платформы .NET и Unity3D:

- *System* (базовая библиотека .NET).

- *System.Collections* / *System.Collections.Generic* (контейнерные типы данных, коллекции).
- *System.Linq* (операции с использованием языка LINQ).
- *System.Text* (работа с кодировкой, форматирование строк);
- *System.Xml* (взаимодействие с XML);
- *System.IO* / *System.IO.Ports* (операции с файлами и портами);
- *System.Net* (работа с сетью, HTTP);
- *System.Threading* (многопоточные операции).
- *UnityEngine* (базовая библиотека Unity3D)
- *UnityEngine.UI* (работа с пользовательским интерфейсом)
- *UnityEngine.EventSystems* (работа с событиями Unity3D)

Размер разработанного программного обеспечения составляет ~2000 строк кода на языке C#. Целевые платформы – Windows, Linux, MacOS.

Минимально необходимые системные требования приведены в таблице 3.1.1.

Таблица 3.1.1. Минимально необходимые требования к аппаратному обеспечению для запуска разработанного ПО для моделирования МР

	<b>Windows</b>	<b>Linux</b>	<b>MacOS</b>
Версия операционной системы	Windows 7 (SP1+) / Windows 10	Ubuntu 20.4 / Ubuntu 18.04 / CentOS 7	High Sierra 10.13 или выше
Процессор	Архитектура X64 с поддержкой набора инструкций SSE2		
Графика	Поддержка DirectX 10 / DirectX 11 / DirectX 12	Поддержка OpenGL 3.2 или выше / Vulkan	Поддержка Metal
Место на диске	200 МБ		
Оперативная память	4Гб		

### 3.2. Компоненты программного обеспечения для моделирования адаптивной системы управления движением модульных роботов

В данном параграфе описаны основные компоненты разработанного программного комплекса.

**Программные модели модулей и робота.** Трехмерные модели типовых мехатронных модулей могут разрабатываться с использованием различных САПР, таких как Компас 3D, Blender, Autodesk 3DMax, Maya, Inventor и др. В общем случае, создаваемая модель модуля должна детально отражать конструкцию, кинематические и динамические параметры реального образца. После завершения разработки в соответствующей программной среде 3-мерная модель модуля импортируется в программный комплекс с добавлением необходимых физических параметров.

Базовый состав библиотеки типовых мехатронных модулей включает программные модели двух реальных образцов (рис. 3.2.1), которые были разработаны в ходе выполнения настоящей диссертационной работы.



Рис. 3.2.1. Разработанные программные модели типовых мехатронных модулей 4L1(а) и 4U2(б).

Формирование программных моделей мехатронно-модульных роботов может осуществляться двумя способами. 1-й из них предполагает пошаговую модульную сборку конструкции, выполняемую пользователем интерактивным образом на экране компьютера (рис. 3.2.2) в режиме «визуального конструктора».

На каждом шаге пользователь выбирает используемые стыковочные площадки и их относительную ориентацию для соединения модулей.



Рис. 3.2.2. Пример формирования программной модели мехатронно-модульного робота в режиме «визуального конструктора».

Создание программной модели робота 2-м способом предполагает ее автоматическое построение по файлу описания. Наиболее удобным способом представления конфигурации робота являются древовидные структуры, описанные таким образом, чтобы они были понятны и человеку, и вычислительному устройству. К такому типу представлений относятся языки разметки основанные на XML. Они находят широкое применение в различных областях [61,63]

В настоящей работе мы предлагаем в качестве основных элементов-сущностей XML описания конфигурации МР следующие:

- модульный реконфигурируемый робот (<robot>);
- типовой мехатронный модуль (<module>);
- сопряжение двух мехатронных модулей (<connection>).

Подробное описание словаря элементов предлагаемого способа описания конфигураций МР представлено в табл. 3.2.1.

Таблица 3.2.1. Словарь элементов описания конфигураций МР.

Элемент	Описание элемента	Атрибуты	Описание атрибута
robot	Корневой элемент XML документа, описывающий модульного робота.	name	Имя робота
		refId	Идентификатор модуля (референтного), от которого будет строиться программная модель робота.
		moduleType	Основной тип мехатронных модулей, из которых состоит робот.
		position	Координаты x, y, z, определяющие положение референтного модуля.
		rotation	Углы Эйлера, определяющие ориентацию референтного модуля.
modules	Контейнер для элементов <module>	-	-
module	Элемент описывающий мехатронный модуль.	id	Уникальный идентификатор.
		qi	Угол в i-м сочленении модуля.
		type (опционально)	Тип мехатронного модуля. Параметр необходим в том случае, когда робот состоит из разных типов мехатронных модулей.
connections	Контейнер для элементов <connection>	-	-
connection	Элемент, описывающий соединение между двумя мехатронными модулями.	from	Идентификатор 1-го модуля.
		to	Идентификатор 2-го модуля.
		surfaceFrom	Имя интерфейсной площадки 1-го модуля.
		surfaceTo	Имя интерфейсной площадки 2-го модуля.
		tilt (опционально)	Угол соединения модулей (если параметр не указан, то по умолчанию равен 0).

Разработанный способ позволяет описывать конфигурации различных типов МР вне зависимости от особенностей их кинематической структуры и конструкции мехатронных модулей. В качестве наглядного примера на рис. 3.2.3, 3.2.4 приведены примеры описания конфигураций МР «Y1» (UPM, Испания) (в конфигурации шагающей платформы с 3-мя конечностями) и «АРАКС» (РТУ МИРЭА) (в конфигурации 4-звенного манипулятора).

В реальных системах построение описания конфигурации происходит при помощи алгоритма обхода дерева в ширину. Главный модуль посылает информационный сигнал соседним, соединенным с ним, модулям. Они, в свою очередь, запоминают, что получили сигнал опроса своего состояния от главного модуля и посылают сигнал своим соседям. Запоминание модулем получения сигнала опроса аналогично «окраске» посещенных вершин в дереве во избежание заикливания работы алгоритма.

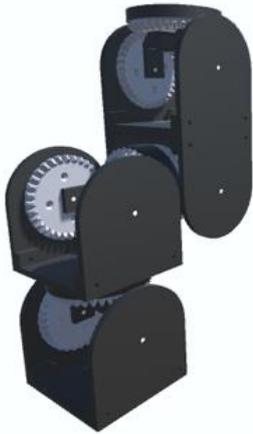


```
<robot name="3Pod" refId="0" moduleType="Y1"
position="0,0,0" rotation="0,0,0">
<modules>
  <module id="11" q1="0" />
  <module id="12" q1="0" />
  <module id="21" q1="0" />
  <module id="22" q1="0" />
  <module id="31" q1="0" />
  <module id="32" q1="0" />
  <module id="0" type="Y1T" />
</modules>
<connections>
  <connection from="0" to="11" surfaceFrom="B" surfaceTo="B"/>
  <connection from="11" to="12" surfaceFrom="T" surfaceTo="T"/>
  <connection from="0" to="21" surfaceFrom="L" surfaceTo="B"/>
  <connection from="21" to="22" surfaceFrom="T" surfaceTo="B"/>
  <connection from="0" to="31" surfaceFrom="R" surfaceTo="B"/>
  <connection from="31" to="32" surfaceFrom="T" surfaceTo="B"/>
</connections>
</robot>
```

a)

б)

Рис. 3.2.3. Модульный робот «Y1» [15] (UPM, Испания) в конфигурации шагающей платформы с 3-мя конечностями: а) натуральный образец; б) описание конфигурации разработанным способом.



а)

```

<robot name="Manipulator4" refId="1"
moduleType="3R1" position="0,0,0" rotation="0,0,0">
<modules>
  <module id="1" q1="0" q2="0" />
  <module id="2" q1="90" q2="0" />
  <module id="3" q1="-90" q2="0" />
  <module id="4" q1="0" q2="0" />
</modules>
<connections>
  <connection from="1" to="2" surfaceFrom="T" surfaceTo="B"/>
  <connection from="2" to="3" surfaceFrom="T" surfaceTo="B"/>
  <connection from="3" to="4" surfaceFrom="T" surfaceTo="B"/>
</connections>
</robot>

```

б)

Рис. 3.2.4. Модульный робот «АРАКС» (РТУ МИРЭА) в конфигурации 4-звенного манипулятора: а) трехмерная модель; б) описание конфигурации разработанным способом.

В случае построения программных моделей, требуется расчет вектора положения  $P$  и кватерниона ориентации  $Q$  каждого модуля системы относительно референтного:

$$P_i = \{a_i, b_i, c_i\}, \quad Q_i = \{x_i, y_i, z_i, w_i\}, \quad i \in \{2, \dots, n\}, \quad (18)$$

где  $P_i$  – трехмерный вектор положения  $i$ -го модуля;

$a_i, b_i, c_i$  – компоненты вектора  $P_i$ ;

$Q_i$  – кватернион, определяющий ориентацию  $i$ -го модуля;

$x_i, y_i, z_i, w_i$  – компоненты кватерниона  $P_i$ ;

$n$  – количество модулей.

Использование кватернионов обусловлено простотой вычисления относительных вращений и позволяет избежать проблемы так называемого «шарнирного замка», возникающей при использовании углов Эйлера. Расчет ориентации  $i+1$ -го модуля относительно предыдущего описывается следующей формулой:

$$Q_{i+1} = Q_{S_{i+1}} \cdot \prod_{j=0}^m Q_j(q_j) \cdot Q_i, \quad (19)$$

где  $Q_{i+1}$  – кватернион, определяющий ориентацию  $i+1$ -го модуля;

$Q_{S_{i+1}}$  – кватернион, реализующий вращение системы координат, в направлении присоединяемой интерфейсной площадки;

$Q_i$  – кватернион, определяющий ориентацию  $i$ -го модуля;

$Q_j$  – кватернион, определяющий ориентацию в зависимости от текущего угла;

$q_j$  в  $j$ -ом сочленении  $i$ -го модуля;

$m$  – количество сочленений в  $i$ -ом модуле.

С каждым модулем связана своя локальная система координат (рис. 3.2.5), расположенная в геометрическом центре модуля.

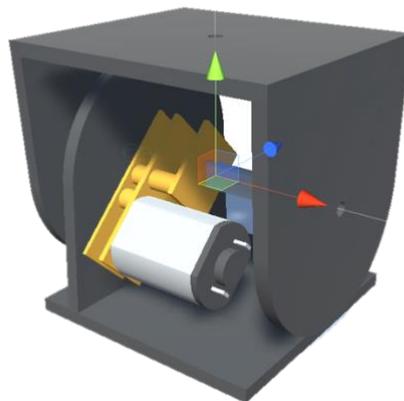


Рис. 3.2.5. Расположение локальной системы координат типового мехатронного модуля.

Тогда позиция  $(i+1)$ -го модуля относительно  $i$ -го модуля будет определяться по формуле:

$$P_{i+1} = P_i + \hat{v} \cdot (S_i + S_{i+1}), \quad (20)$$

где  $S_i$  – расстояние от центра координат  $i$ -го модуля до интерфейсной площадки, к которой присоединен  $i+1$  модуль (см. рис. 3.2.6);  
 $S_{i+1}$  – расстояние от центра координат  $(i+1)$ -го модуля до интерфейсной площадки, к которой присоединен  $i$ -й модуль (см. рис. 3.2.6);  
 $\hat{v}$  – единичный вектор, характеризующий направление соединения двух модулей.

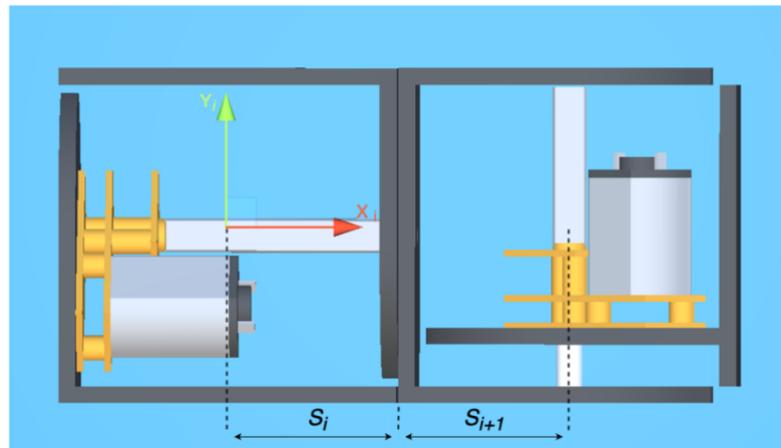


Рис. 3.2.6. Иллюстрация определения позиции  $i+1$ -го модуля, относительно локальной системы координат  $i$ -го модуля.

Ориентация  $i+1$  модуля, относительно  $i$  определяется следующим соотношением:

$$Q_{i+1} = Q_s \cdot Q_t \cdot Q_i , \quad (21)$$

где  $Q_s$  – кватернион, совмещающий направление стыковочных интерфейсных площадок,

$Q_t$  – кватернион, представляющий поворот присоединяемой площадки

$Q_i$  – кватернион, задающий ориентацию  $i$ -го модуля.

В качестве примера рассмотрим стыковку модуля В нижней площадкой к левой площадке модуля А под углом  $90^\circ$  (рис. 3.2.7). Модуль А расположен в некоторой точке пространства с начальной ориентацией  $Q_A$  (рис. 3.2.7а). Затем по формуле 20 рассчитывается вектор положения модуля В, и модуль устанавливается в эту точку с ориентацией  $Q_A$  (рис. 3.2.7б). Для ориентации нижней площадки модуля В кватернион  $Q_A$  умножается слева на  $Q_t$  (рис. 3.2.7в). Для поворота модуля относительно левой площадки на  $90^\circ$  произведение  $Q_t Q_A$  умножается слева на  $Q_s$  (рис. 3.2.7г).

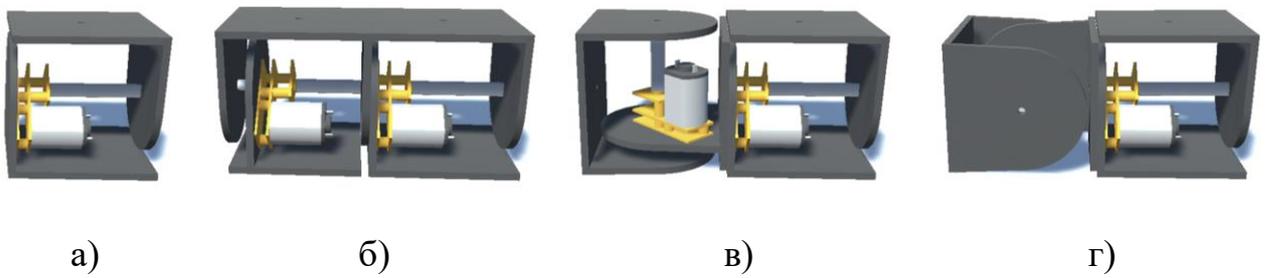


Рис. 3.2.7. Иллюстрация определения ориентации  $i+1$ -го модуля, относительно локальной системы координат  $i$ -го модуля.

Современные языки программирования, включая C#, на котором разрабатывалось ПО для моделирования и программирования МР, позволяют удобно работать с XML документом, выполняя поиск и обращение к конкретным узлам дерева (Листинг 3.1).

Кроме того, дополнительно разработаны функции для автоматического создания типовых конфигураций «гусеница» (листинг 3.2) и «колесо» с возможностью чередования модулей. Чередование предполагает изменение ориентации последующего модуля на  $90^\circ$  (рис. 3.2.8). Отличие алгоритма построения МР в конфигурации «колесо» от алгоритма построения «гусеницы» заключается в повороте степени свободы нужных модулей на  $90^\circ$ .

---

**Листинг 3.1.** Алгоритм построения программной модели МР по файлу описания.

---

**ВВОД:** XML документ

Очередь ids

modules  $\leftarrow$  <Получение элементов с тегом «module»>

connections  $\leftarrow$  <Получение элементов с тегом «connection»>

<Установка референтного модуля>

<Добавляем id референтного модуля в очередь ids>

**ПОКА** (Очередь ids не пустая)

    m  $\leftarrow$  <Извлекаем элемент из очереди ids>

**ПОКА** (Очередь connections [currentId] не пустая)

        m2  $\leftarrow$  <Извлекаем информацию о присоединяемом модуле>

**ЕСЛИ** (m2 не установлен) **ТО**

            <Устанавливаем m2 относительно m>

            <Добавляем id модуля m2 в очередь ids>

---

---

**Листинг 3.2.** Алгоритм построения программной модели МР в конфигурации «гусеница».

---

**ВВОД:** количество модулей n (целое), положение p (вектор3), чередование alt (лог)

i  $\leftarrow$  1

<Устанавливаем модуль i в положение p>

**ПОКА** (i  $\leq$  n)

    tilt  $\leftarrow$  0

**ЕСЛИ** (alt = ИСТИНА) **ТО**

**ЕСЛИ** (i mod 2 = 0) **ТО**

            tilt  $\leftarrow$  90

**ИНАЧЕ**

            tilt  $\leftarrow$  270

    <Устанавливаем модуль i относительно i-1>

    i  $\leftarrow$  i + 1

---

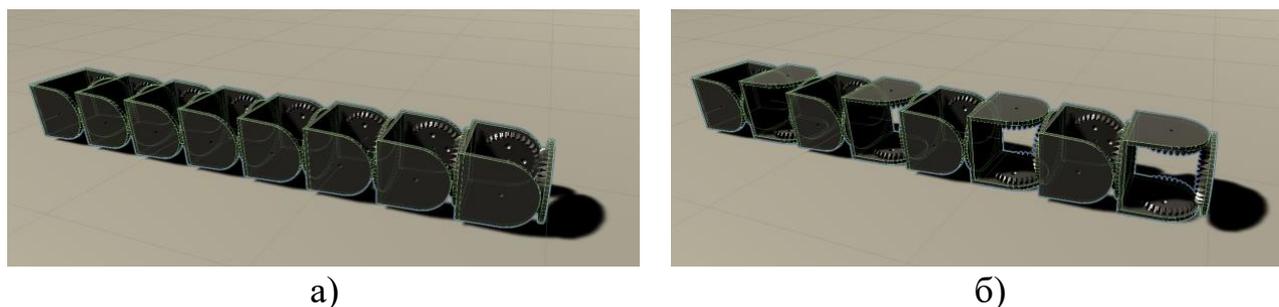


Рис.3.2.8. Пример МР в конфигурации «гусеница» а) без чередования модулей, б) с чередованием модулей.

**Алгоритмы управления модульными роботами.** Библиотека алгоритмов, входящая в состав разработанного программного комплекса, реализует различные методы управления МР. Исполнительный уровень основан на 2-х подходах – управляющих таблицах и гармонических осцилляторах.

Уровень построения траекторий движения основан на методе ААУ, реализующем адаптивное управление МР. Алгоритм содержит две ветви, реализующие функции обучения и управления (рис. 3.2.9).

В начале работы программы инициализируются функции получения данных с датчиков (функция S), выполнения действий (функция A) и расчета «эмоциональной» оценки (функция E). Функция S содержит в себе ссылки на функции получения данных с датчиков программной модели, а именно углов в сочленениях и датчика расстояния до целевой точки. Функция A содержит ссылки на все возможные действия, которые может совершить система (16). Функция E, может задаваться одним из 2 предложенных способов (15), (17).

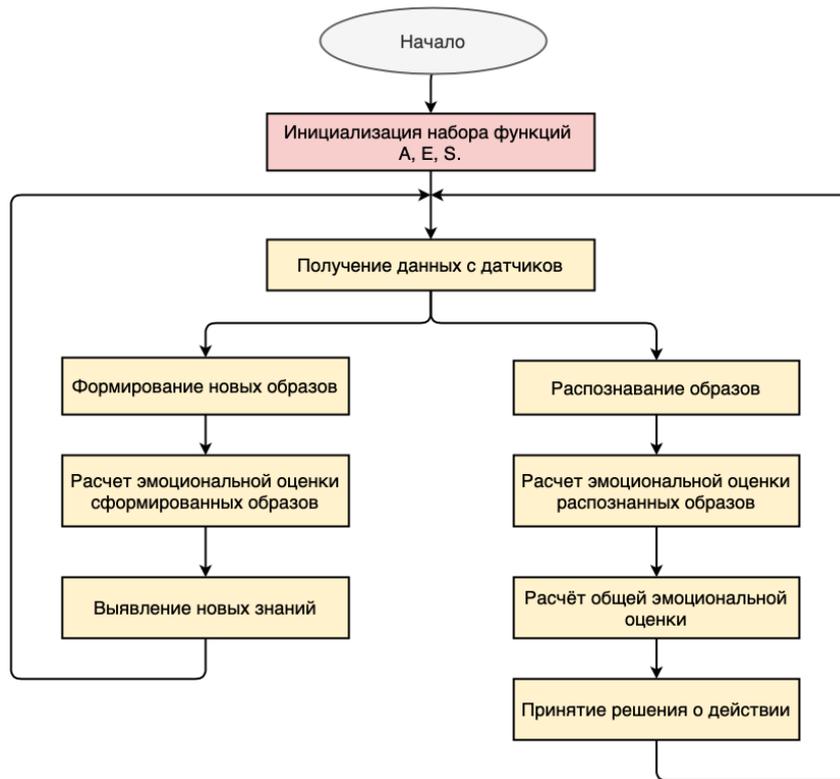


Рис. 3.2.9. Блок-схема программы, реализующей метод Автономного Адаптивного управления модульными роботами.

Контур обучения состоит из 3-х шагов:

1. Формирование новых образов;
2. Расчёт «эмоциональной оценки» сформированных образов;
3. Выявление новых знаний на основе распознанных и сформированных образов на текущем шаге и действия, совершенного на предыдущем шаге.

Контур управления состоит из 4-х шагов:

1. Распознавание образов;
2. Расчёт «эмоциональной» оценки распознанных образов;
3. Расчёт общей «эмоциональной» оценки, как суммы оценок распознанных образов;
4. Принятие решения о действии на основе базы знаний.

Подробнее рассмотрим работу каждого контура. Контур *обучения* начинает свой такт работы запуском подсистемы формирования новых образов, используя данные, полученные с датчиков. Формируются новые образы, если выполняются условия их формирования, и тогда создается новый программный объект «Образ», описанный на языке признаков. В зависимости от применяемого способа (см. параграф 2.2.) образами могут быть текущие показания датчиков или положение цели относительно схвата манипулятора. Этот объект записывается в «Память сформированных образов» (алфавит классов). Каждому из сформированных образов рассчитывается значение «эмоциональной» оценки.

С появлением сформированных образов, складываются условия для формирования «элементарных знаний», которые представляют собой выявленные в предыстории статистически неслучайные последовательности событий следующего вида:

$$((\{O_{t-3}^i\} \& A_{t-2}^j) \rightarrow \{O_{t-1}^k\} \rightarrow E_t^l) \mid ((M_i > N) \vee (|E_t^l| > E)), \quad (22)$$

где  $\{O_{t-3}^i\}$  – множество распознанных в момент  $t-3$  образов,

$A_{t-2}^j$  – действие, совершенное в момент  $t-2$ ,

$\{O_{t-1}^k\}$  – множество образов, распознанных в момент  $t-1$ ,

$E_t^l$  – «эмоциональная оценка», распознанная в момент  $t$ .

Критерии формирования элементарного знания:

$M_i > N$  – число наблюдений  $M$  такой последовательности больше заданного порога  $N$ , либо

$|E_t^l| > E$  – важность прецедента («эмоциональная» оценка) очень большая.

Совокупность элементарных знаний вида (22), структурированная некоторым способом, представляет собой Базу Знаний (БЗ). БЗ, на основании эмпирического опыта УС показывает, чем закономерно заканчиваются действия  $A^j$ ,

совершенные в различаемых УС условиях  $\{O^i\}$ , а именно, какие образы  $\{O^k\}$  будут распознаны и какие «эмоциональные» оценки  $E^l$  за этим последуют. Поскольку знания находятся в предыстории, но подтверждены статистически ( $M_i > N$ ), то это дает основания полагаться на них при прогнозировании на будущее. Т.е., если при распознанном в текущий момент образе совершить действие, то можно надеяться, что в результате его будут распознаны те же образы и «эмоциональные» оценки, которые многократно распознавались в предыдущих прецедентах. Если же элементарное знание вида (22) попало в БЗ по одному прецеденту, но принесло некогда большую по модулю «эмоциональную» оценку  $E^l$ , то также есть смысл надеяться на него. Заметим, что в методе ААУ знания постоянно обновляются и максимально актуальны в любой момент времени в отличие, например, от ИНС, где обучение проводилось по заранее заготовленной обучающей выборке еще до начала управления, и знания, возможно, могли утратить свою актуальность.

БЗ можно структурировать и сохранять в разных нотациях, например, в табличном виде [67]. В данном случае БЗ представляет собой ассоциативный массив, позволяющий хранить пары («ключ» - «значение»). Ключом этого массива является объект «Образ», а значением ассоциативный массив пар «Действие» - «Образ». Ассоциативный массив в современных языках программирования реализуется, как правило, на основе хэш-таблицы, поэтому операции вставки, поиска и удаления элементов выполняются в среднем за время  $O(1)$  [111].

Функция выявления новых знаний записывает текущий распознанный образ как результат выполнения действия на предыдущем шаге, тем самым формируя Базу знаний.

В контуре *управления*, данные, полученные от датчиков, передаются в подсистему распознавания образов. В ней распознаются образы, уже известные системе, и описанные на языке признаков.

Подсистема «аппарат эмоций», используя инициализированную в начале работы программы функцию расчёта «эмоциональной» оценки, в контуре управления:

- рассчитывает «эмоциональную» оценку  $e^t$  для каждого из распознанных образов;
- рассчитывает общую оценку  $E^t$  текущего распознанного состояния системы;
- вычисляет 1-ю производную «эмоциональной» оценки  $\dot{E}^t$ ;
- обновляет значения самой худшей и самой лучшей «эмоциональной» оценки, которые появлялись в процессе функционирования системы;
- обновляет записи «эмоциональной» оценки текущего распознанного образа в базе знаний.

Затем функция принятия решения о действии (листинг 3.3), используя алгоритм поиска в глубину (листинг 3.4), ищет действие, которое приведет к наилучшей «эмоциональной» оценке. В случае если текущий столбец Базы знаний недостаточно заполнен, то на основе механизма равновероятностного случайного выбора выбирается случайное действие. Действие выполняется и алгоритм переходит опять к шагу 1.

---

### Листинг 3.3. Алгоритм принятия решения о действии (метод ААУ)

---

**ВВОД:** требуемая заполненность столбца базы знаний  $f$  (вещественное)

$bestActions \leftarrow$  <Поиск списка действий с наилучшим результатом>

$unknownActions \leftarrow$  <Поиск списка действий с неизвестным результатом>

**ЕСЛИ** (Количество( $unknownActions$ ) / Количество( $actions$ )  $> f$ ) **ТО**

<выполняем случайное действие из  $unknownActions$ >

**ИНАЧЕ ЕСЛИ** (Количество( $bestActions$ )  $> 0$ ) **ТО**

<выполняем случайное действие из  $bestActions$ >

<запоминаем выполненное действие>

---

---

**Листинг 3.4.** Поиск списка действий с наилучшим результатом (метод ААУ)
 

---

**ФУНКЦИЯ** Search(actions, p, a, d)

**ВВОД:** список возможных действий actions, образ p, действие a, глубина поиска d

**ЕСЛИ** (d = 1) **ТО**

**ВЕРНУТЬ** <Эмоциональная оценка при выполнении действия a при образе p>

E ← <Минимально возможная эмоциональная оценка>

**ПОКА** (i < Количество(actions))

**ЕСЛИ** (БазаЗнаний[p][a] ≠ ∅) **ТО**

e ← Search (БазаЗнаний[p][a], actions[i], d - 1)

**ЕСЛИ** (e > E) **ТО**

E ← e

i ← i + 1

---

**Взаимодействие с внешними приложениями.** Взаимодействие разработанного программного обеспечения с внешними приложениями и пользовательскими программами основано на клиент-серверной архитектуре (рис. 3.2.10). Организация клиент-серверной архитектуры позволяет абстрагировать написание пользовательских приложений от среды моделирования, используя привычные для разработчика среду и язык программирования, включая такие распространенные средства, как Matlab, Microsoft Visual Studio, Python IDE и др. При этом взаимодействие созданного компонента с моделирующим комплексом осуществляется по определённому протоколу через программный интерфейс приложения (API). В рамках данной схемы информационного взаимодействия, построенного в соответствии с протоколом HTTP, разработанный программный комплекс поддерживает обработку стандартных POST-запросов, параметры которых приведены в таблице 3.2.2.

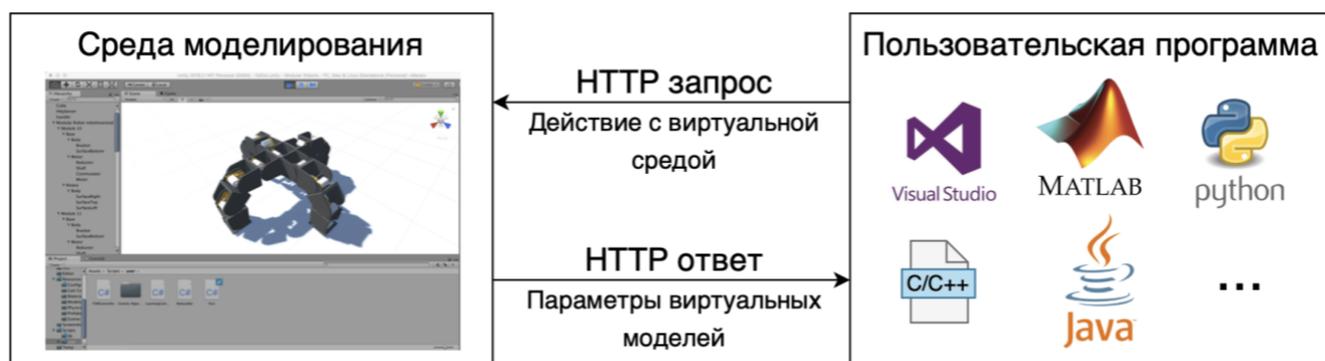


Рис. 3.2.10. Организация взаимодействия моделирующего комплекса с внешними пользовательскими приложениями на основе клиент-серверной архитектуры.

Таблица 3.2.2. Описание API методов и их параметров для взаимодействия с внешними приложениями.

Название запроса	Описание запроса	Параметры запроса	Описание параметров
Create	Создание робота на трехмерной сцене	XML	Имя файла из библиотеки конфигураций / Xml описание конфигурации робота
		RobotName	Название (идентификатор) робота, который будет создан на трехмерной сцене
Delete	Удаление робота из трехмерной сцены	RobotName	Название (идентификатор) робота, к которому будет применен данный запрос
		Control Method	Тип метода управления (ААУ, управляющие таблицы, гармонический осциллятор и др.)
Start Control	Запуск управления роботом	Control Method Params	Параметры, специфичные для метода управления (например способ расчета «эмоциональной» оценки, параметры гармонического осциллятора и др.)
		RobotName	Название (идентификатор) робота, к которому будет применен данный запрос
Stop Control	Остановка управления роботом	RobotName	Название (идентификатор) робота, к которому будет применен данный запрос
Get Environment	Получение информации о роботе и программной среде		—

### 3.3. Программное взаимодействие средств моделирования с макетным образцом модульного робота

В рамках данного диссертационного исследования был разработан макетный образец модульного робота. В основу макета положен типовой модуль (рис. 3.3.1а), технические спецификации которого приведены в Таблице 3.3.1. Детали (рис. 3.3.1б) спроектированы в ПО Autodesk Inventor и распечатаны на 3D принтере. 2-сторонняя печатная плата разработана в ПО Altium Designer. Ее внешний вид, 3D модель, трассировка и расположение компонентов приведены на рис.3.3.2.



Рис. 3.3.1. а) Внешний вид разработанного макетного образца типового мехатронного модуля, б) разработанные детали конструкции.

Таблица 3.3.1. Технические характеристики типового модуля.

Параметр	Значение
Габариты	40×40×40 (мм)
Питание	5В (внешнее)
Микроконтроллер	STM32F030
Степени свободы	1 (вращательная)
Угол поворота	[-90°; 90°]
Момент	0,23 Н·м
Контактные площадки	4
Механическое сцепление	Постоянные магниты
Передача информации (между модулями)	UART / CAN
Передача информации (между роботом и управляющим компьютером)	USB / Bluetooth

Выбор оптимальной конструкции типового модуля является открытой проблемой, требующей разработки специального математического критерия. Поскольку такого критерия не существует на настоящий момент, наш выбор был основан на статистических данных. Подавляющее большинство типовых модулей, разработанных различными научными группами, имеют похожий «кубический» вид. Основанные отличия заключаются в габаритах, количестве степеней свободы (1 или 2), типом интерфейсных площадок, используемыми типами двигателей и аппаратными характеристиками.

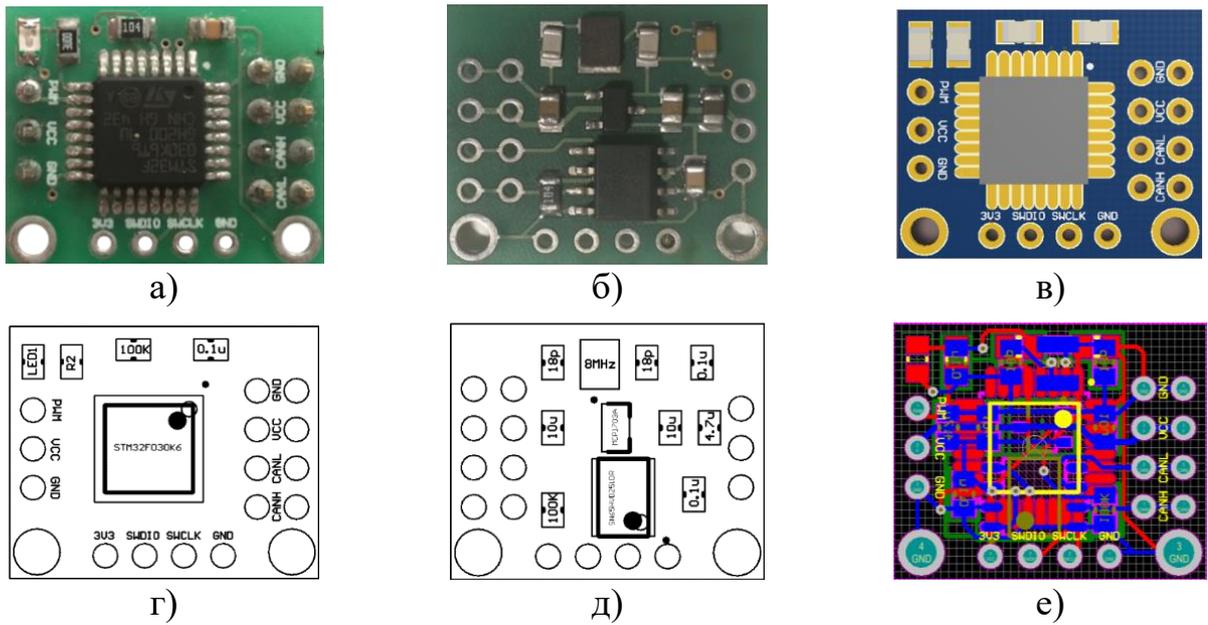


Рис. 3.3.2. Плата управления типовым модулем: а) внешний вид сверху, б) внешний вид снизу, в) 3-мерная модель, г) расположение компонентов сверху, д) расположение компонентов снизу, е) трассировка.

На внешних сторонах модуля расположены 4 контактные площадки, на каждой из которых установлено четыре пары магнитов для фиксации модулей между собой.

Модуль поделен на две L-образные части, которые перемещаются друг относительно друга вокруг поперечной оси в диапазоне  $[-90^\circ, 90^\circ]$  градусов. Движение частей друг относительно друга обеспечивается сервоприводом GS-9025MG. Управление углом поворота осуществляется на основе метода широтно-импульсной модуляции. Длительность импульса пропорциональна углу поворота рулевой машинки и лежит в диапазоне от 1 до 2мс, а период следования импульсов составляет 20мс. Управление ШИМ сигналом в микроконтроллерах серии STM32 может осуществляться путем настройки таймера-счетчика в режим ШИМ, и записи соответствующего значения скважности в регистр CCR. При этом, особенность сервоприводов состоит в том, что поворот на заданный угол осуществляется с максимально возможной скоростью. Поэтому, для управления скоростью движения, используется

алгоритм, приведенный в листинге 3.5. Таким образом, настраивая частоту вызова данной функции (частоту прерываний таймера-счетчика), можно управлять скоростью движения сервопривода.

---

**Листинг 3.5.** Управление скоростью движения сервопривода на МК STM32

---

```
void TIM16_IRQHandler(void)
{
    if (currentAngle == targetAngle)
        return;
    if ((currentAngle < ANGLE_MAX) && (currentAngle > ANGLE_MIN))
    {
        if (currentAngle < targetAngle)
            currentAngle++;
        else if (currentAngle > targetAngle)
            currentAngle--;
        TIM14->CCR1=currentAngle;
        counter=0;
    }
    counter++;
}
```

---

Связь между модулями осуществляется на основе общей шины по CAN протоколу, обеспечивающим высокую скорость обмена информацией и надежную защиту пакетов данных от повреждений. Каждый модуль имеет в своем составе преобразователь UART-CAN, благодаря которому он подключается к общей шине.

Проведение экспериментальных исследований на макетном образце МР связано со сложностью предварительной подготовки к эксперименту, переноса алгоритмов управления из модельного комплекса на бортовой вычислитель и т.д. В этом случае становятся существенными временные затраты на перенос разработанных алгоритмов на борт робота, а также существует вероятность допущения ошибок при программировании.

Исходя из этого предлагается подход, заключающийся в интеграции средств управления макетным образцом в разработанный программный комплекс программного моделирования МР (рис.3.3.3) [104].

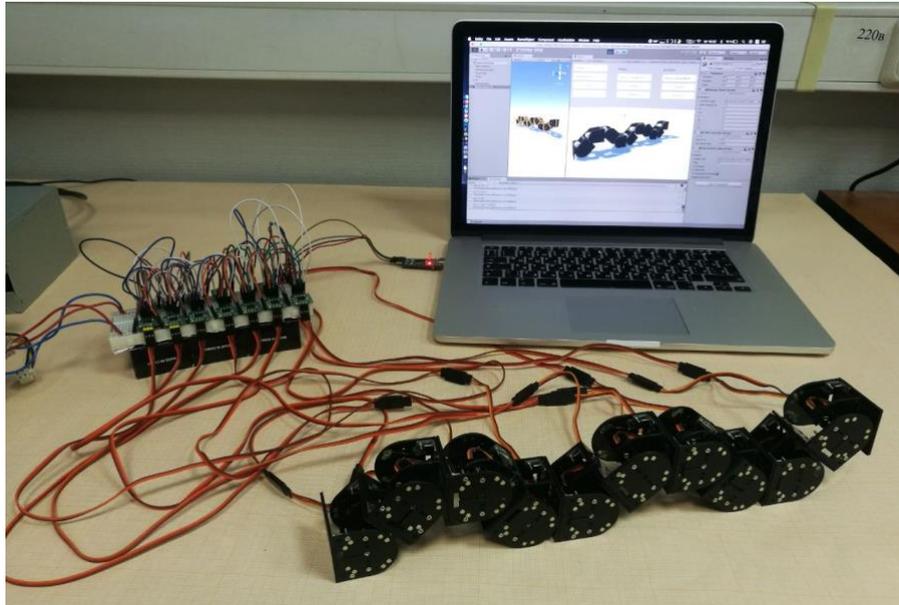


Рис. 3.3.3. Демонстрация одновременного управления программной моделью и макетным образцом модульного робота.

Ключевым достоинством такой реализации является неразделимость программного кода для управления как программной моделью, так и реальным роботом. При этом комплекс подключается к общей CAN-шине робота через преобразователь, например USB/CAN или Bluetooth/CAN (рис. 3.3.4). Стоит отметить, что при таком подходе функции главного модуля исполняет внешний вычислитель (разработанный программный комплекс), из-за удобства постановки экспериментов. Однако в реальных системах разработанное программное обеспечение должно быть загружено непосредственно на борт МР, тем самым обеспечивая автономность робота.

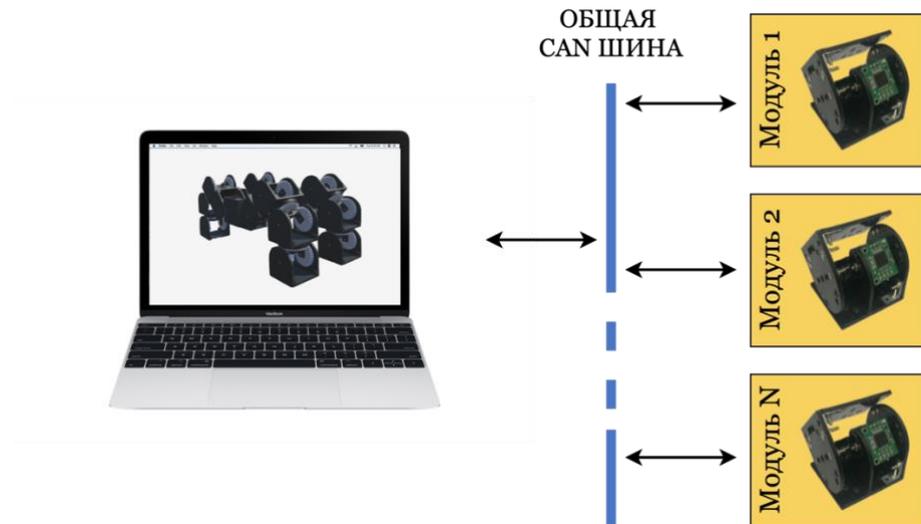


Рис. 3.3.4. Структурная схема системы на основе общей шины данных.

Одновременное управление программной моделью и макетным образцом достигается за счет реализованного драйвера, который преобразует управляющий сигнал алгоритма управления в специальное сообщение (рис. 3.3.5).

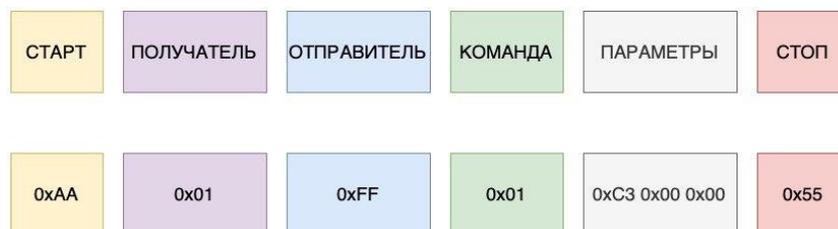


Рис. 3.3.5. Структура сообщения межмодульного протокола передачи данных.

Оно направляется в CAN-шину и доставляется требуемому адресату (модулю). После этого сообщение преобразуется в низкоуровневые инструкции управления приводом модуля. За стартовым байтом следуют уникальные однобайтовые идентификаторы получателя и отправителя сообщения. Идентификаторы в диапазоне 1 – 254 присваиваются модулям на стадии загрузки прошивки на бортовой микроконтроллер. Адрес 0 является широкоэмитерным

(передача сообщения на все узлы системы), а адрес 255 – адресом центра управления (программного комплекса для моделирования и управления МР).

В сообщении указывается команда, которую получатель должен выполнить. Дополнительно могут указываться параметры команды, например для команды поворота угла, в качестве параметра передается угловое значение. Полный список команд представлен в Таблице 3.3.2.

Таблица 3.3.2. Список команд управления мехатронным модулем

<b>Список команд управления</b>		
<b>Управление углом поворота двигателя</b>		
<i>Байт</i>	<i>Название</i>	<i>Значение</i>
0x01	COM_ANGLE_SET	Задать угол поворота двигателя
0x02	COM_ANGLE_MINUS	Повернуть вал двигателя на заданный угол в отрицательном направлении
0x03	COM_ANGLE_PLUS	Повернуть вал двигателя на заданный угол в положительном направлении
0x04	COM_ANGLE_GET	Получить значение текущего угла поворота
0x06	COM_ANGLE_MIN	Получить значение минимально возможного угла
0x07	COM_ANGLE_CENTER	Получить значение угла, соответствующего центральному положению вала
0x08	COM_ANGLE_MAX	Получить значение максимально возможного угла
<b>Информационные команды</b>		
<i>Байт</i>	<i>Название</i>	<i>Значение</i>
0x80	COM_INFO_IS_ONLINE	Запрос, доступен ли модуль
0xDA	COM_INFO_ONLINE_OK	Ответ, доступен ли модуль
0x81	COM_INFO_VERSION	Запрос номера текущей версии бортового ПО
<b>Управление светодиодом, расположенном на плате управления</b>		
<i>Байт</i>	<i>Название</i>	<i>Значение</i>
0x50	COM_LED_OFF	Выключить светодиод
0x51	COM_LED_ON	Включить светодиод
0x52	COM_LED_TOGGLE	Переключить светодиод
0x53	COM_LED_ONOFF	Включить светодиод на X мс.
0x54	COM_LED_BLINK	Мигать светодиодом с периодом в X мс.

Таблица 3.3.2. (Продолжение)

Управление скоростью двигателя		
Байт	Название	Значение
0x21	COM_SET_SPEED	Установить величину скорости двигателя
0x22	COM_RESET_SPEED	Установить величину скорости двигателя по умолчанию

В качестве примера, на рис. 3.3.6 - 3.3.8 показаны фрагменты экспериментов по управлению разработанным модульным роботом в конфигурации «гусеница», «колесо» и «манипулятор».

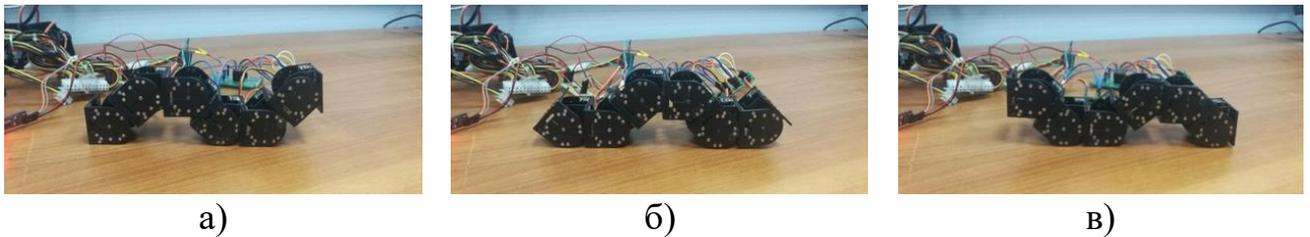


Рис. 3.3.6. Фрагменты экспериментов по управлению модульным роботом в конфигурации «гусеница».

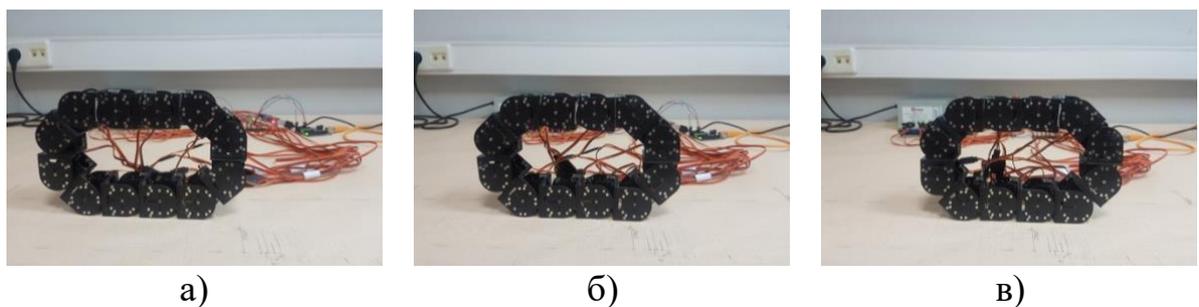


Рис. 3.3.7. Фрагменты экспериментов по управлению модульным роботом в конфигурации «колесо».



а)



б)



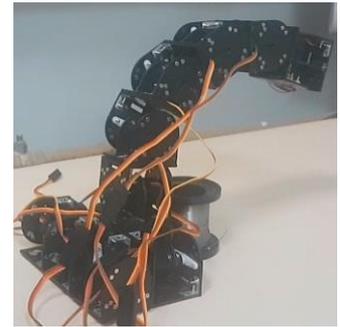
в)



г)



д)



е)

Рис. 3.3.8. Фрагменты экспериментов по управлению модульным роботом в конфигурации «манипулятор».

Таким образом, был разработан макетный физический образец МР, который управляется одновременно с программной моделью робота. Программные модели в разработанном ПО адекватны реальному образцу, за счет моделирования физических свойств робота, и эксперименты могут быть поставлены не на макетном образце, а на программных моделях.

### **Выводы по главе**

В 3-й главе был представлен новый способ построения программного обеспечения для моделирования модульных роботов, описана его структурная организация и система классов. Разработано программное обеспечение на основе предложенного способа, описаны его основные компоненты.

Также предложен способ описания кинематической структуры модульных роботов на основе XML, и алгоритмы построения программных моделей модульных роботов по файлу описания.

Одним из ключевых элементов в составе библиотеки алгоритмов комплекса является реализация метода Автономного Адаптивного Управления применительно к управлению модульными роботами.

Кроме того, был описан способ взаимодействия разработанного макетного образца модульного робота с предложенными средствами программного моделирования. Программные модели адекватны реальному роботу, за счет моделирования физических свойств реального робота. Это в свою очередь позволяет проводить эксперименты на программных моделях, сохраняя достоверность получаемых экспериментальных данных.

## **ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ АЛГОРИТМОВ АДАПТИВНОГО УПРАВЛЕНИЯ МОДУЛЬНЫМИ РОБОТАМИ**

В данной главе описаны результаты проведенных экспериментальных исследований предложенной во 2-й главе адаптивной системы управления на разработанных и описанных в 3-й главе программном обеспечении.

### **4.1. Экспериментальные исследования адаптивной системы управления движением модульных роботов в конфигурациях «манипулятор» и «шагающая платформа»**

Цель экспериментов состояла в исследовании работоспособности разработанных способов применения метода ААУ применительно к управлению МР в конфигурациях «манипулятор» и «шагающая платформа». В основу экспериментов легло описанное в предыдущей главе программное обеспечение для моделирования МР. Рассмотрены 2 способа управления МР в конфигурации «манипулятор» (см. параграф 2.2) и управление «шагающей платформой». Стоит отметить, что моделирование производилось с ограниченным набором датчиков, а именно, датчиков угла поворота в модулях, поскольку целью экспериментов является демонстрация способностей системы ААУ к самообучению на примере движения МР. Расстояние до цели определялось программно в виртуальной среде.

Предложенный 1-ый способ (параграф 2.2) исследовался при различных начальных параметрах  $\Delta$  и разном количестве модулей в составе кинематической цепи. На рис. 4.1.1 показаны, полученные в результате экспериментов, зависимости времени обучения от количества модулей в

кинематической цепи при разных начальных значениях параметра  $\Delta$ . Видна закономерность между количеством модулей, точностью и временем обучения при фиксированном значении  $\Delta$ . Чем больше модулей находится в кинематической цепи, тем больше время обучения. При этом видно, что чем выше значение  $\Delta$ , тем быстрее осуществляется обучение.

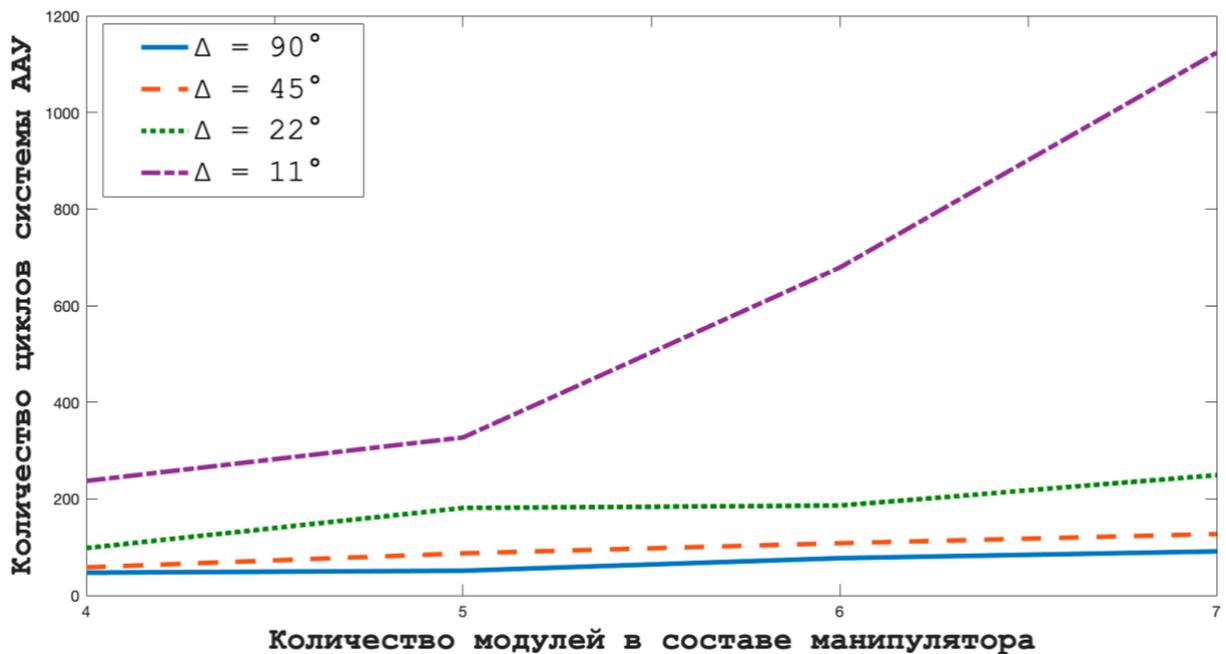


Рис. 4.1.1. Зависимости времени обучения от количества модулей в кинематической цепи при разных начальных значениях параметра  $\Delta$  (1-й способ).

На рис. 4.1.2. изображены зависимости точности достижения целевой точки от значения параметра  $\Delta$  при различном количестве модулей в составе манипулятора. Из экспериментальных данных видно, что чем меньше значение величины  $\Delta$ , тем выше точность управления. Однако, как показывает предыдущий график (рис. 4.1.1), уменьшение значения величины  $\Delta$  приводит к увеличению времени обучения.

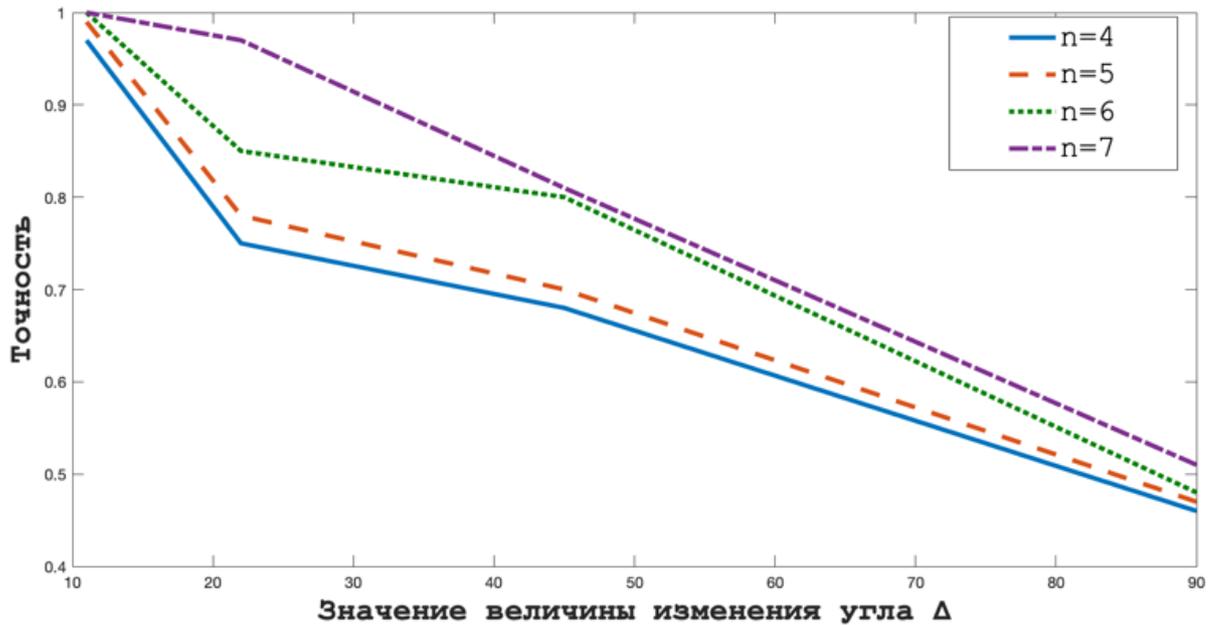


Рис. 4.1.2. Зависимости точности достижения целевой точки от значения параметра  $\Delta$  при различном количестве модулей в составе манипулятора (1-й способ).

На рис. 4.1.3 показан «процесс» обучения, из которого видно, что с течением времени функционирования системы ААУ точность возрастает.

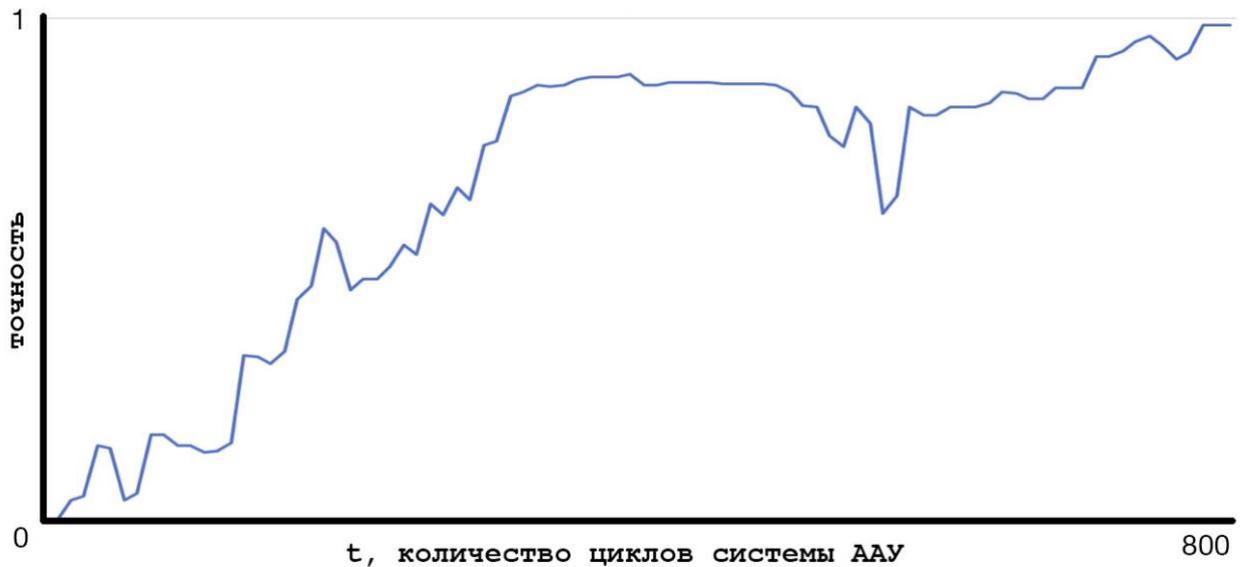


Рис. 4.1.3. Зависимость точности управления от времени обучения при  $\Delta = 5$ ,  $n = 5$  (1-й способ).

Резкие скачки на графике (рис. 4.1.3) связаны с выбором системой одного из действий. Когда график идет вверх, система выбирала те действия, которые ведут к положительному результату, то есть уменьшению рассогласования между схватом и целевой точкой. В случае отрицательных скачков, система выбирала те действия, которые увеличивали это рассогласование. Однако, если ситуация с течением времени не улучшалась, система возвращалась к наилучшей ситуации (благодаря структуре образов) и продолжала поиск наилучших действий.

На рис. 4.1.4 проиллюстрирована способность модульного манипулятора к самообучению.

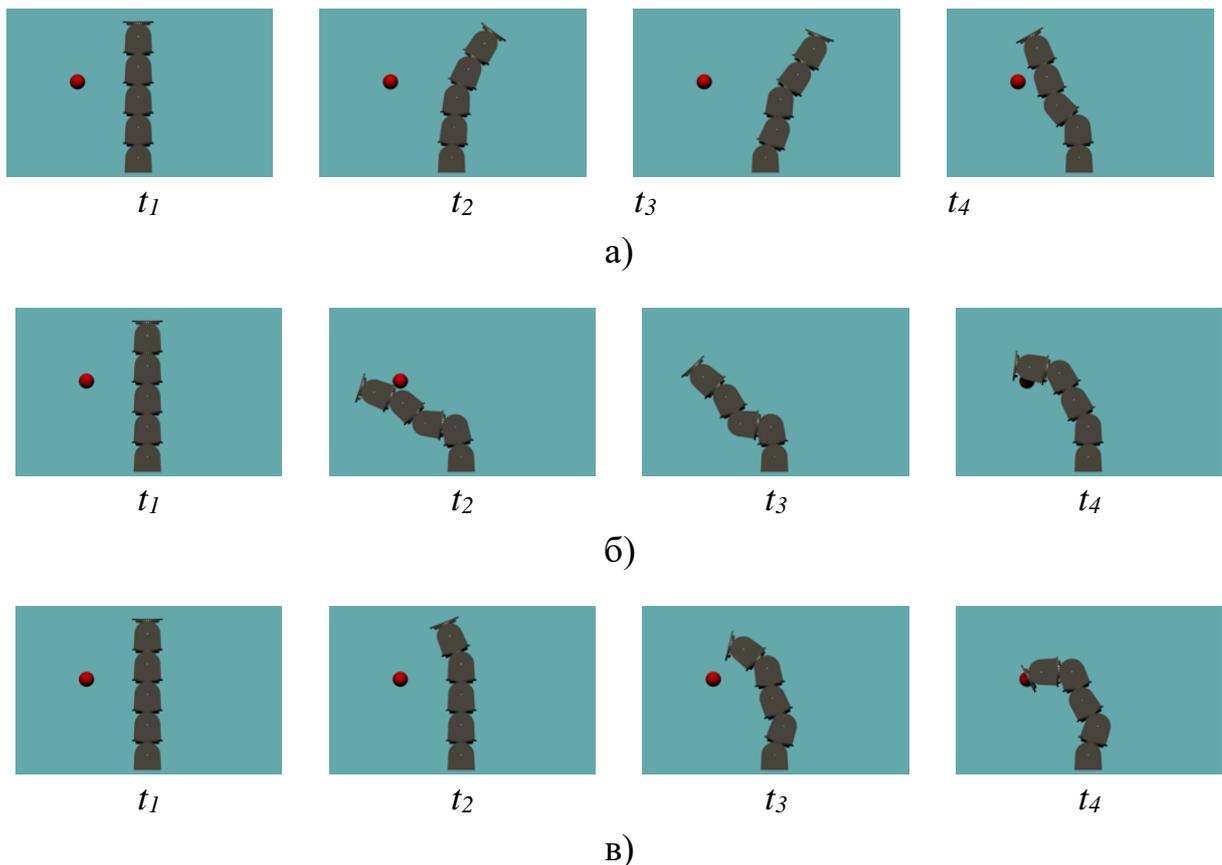


Рис. 4.1.4. Демонстрация способности самообучения модульного манипулятора (1-й способ): а) БЗ робота пустая; б) в БЗ частично заполнена знаниями; в) робот обучился, БЗ заполнена

Как видно из рис. 4.1.4а, при пустой БЗ, робот из исходной позиции движется хаотично. Когда через некоторое время БЗ робота уже частично заполнена знаниями, робот из этой же исходной позиции сразу же, хотя и не очень точно, начинает двигаться в направлении целевой точки (рис. 4.1.4б). В полностью обученном состоянии робот быстро и точно приходит в целевую точку (рис. 4.1.4в). Видео-демонстрацию можно посмотреть в сети Интернет по адресу [https://youtu.be/J0v\\_ZsfIDa8](https://youtu.be/J0v_ZsfIDa8).

На рис. 4.1.5. показан результат обучения робота с изначально заклиненным модулем. Этот модуль не может изменять свой угол поворота, однако у системы об этом нет информации. В процессе своего функционирования система может попытаться повернуть заклиненный модуль на величину  $\Delta$ , но это не даст никакого эффекта. В результате, несмотря на поломку модуля, робот обучается приходить в целевую точку.

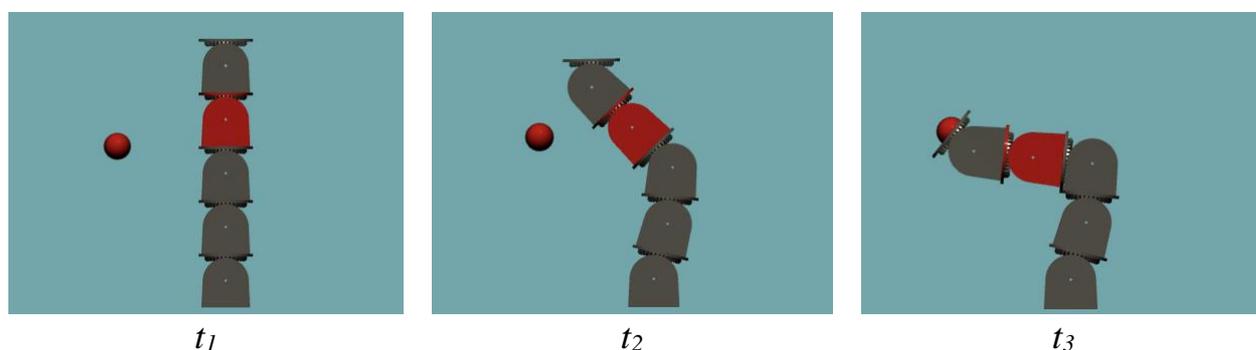


Рис. 4.1.5. Демонстрация результата обучения модульного манипулятора с заклиненным модулем.

Способность к переобучению продемонстрирована на рис. 4.1.6. В начальный момент времени  $t_1$  (рис. 4.1.6а) БЗ знаний робота пустая, и он начинает двигаться хаотично. К моменту времени  $t_4$  робот обучился приходить в целевую точку. Затем робот возвращается в целевую точку, и имитируется поломка модуля (рис.4.1.6б) (выделен красным цветом). Если теперь робот попытается прийти в целевую точку, используя полученные знания, то, из-за заклинивания модуля, он промахнется (рис.4.1.6в). Затем

робот начинает переобучение (рис.4.1.6г) и постепенно переобучается перемещать схват в целевую точку, несмотря на выход из строя одного из модулей. Видео- демонстрацию можно посмотреть в сети Интернет по адресу [https://youtu.be/JFzwZSUw\\_nM](https://youtu.be/JFzwZSUw_nM) .

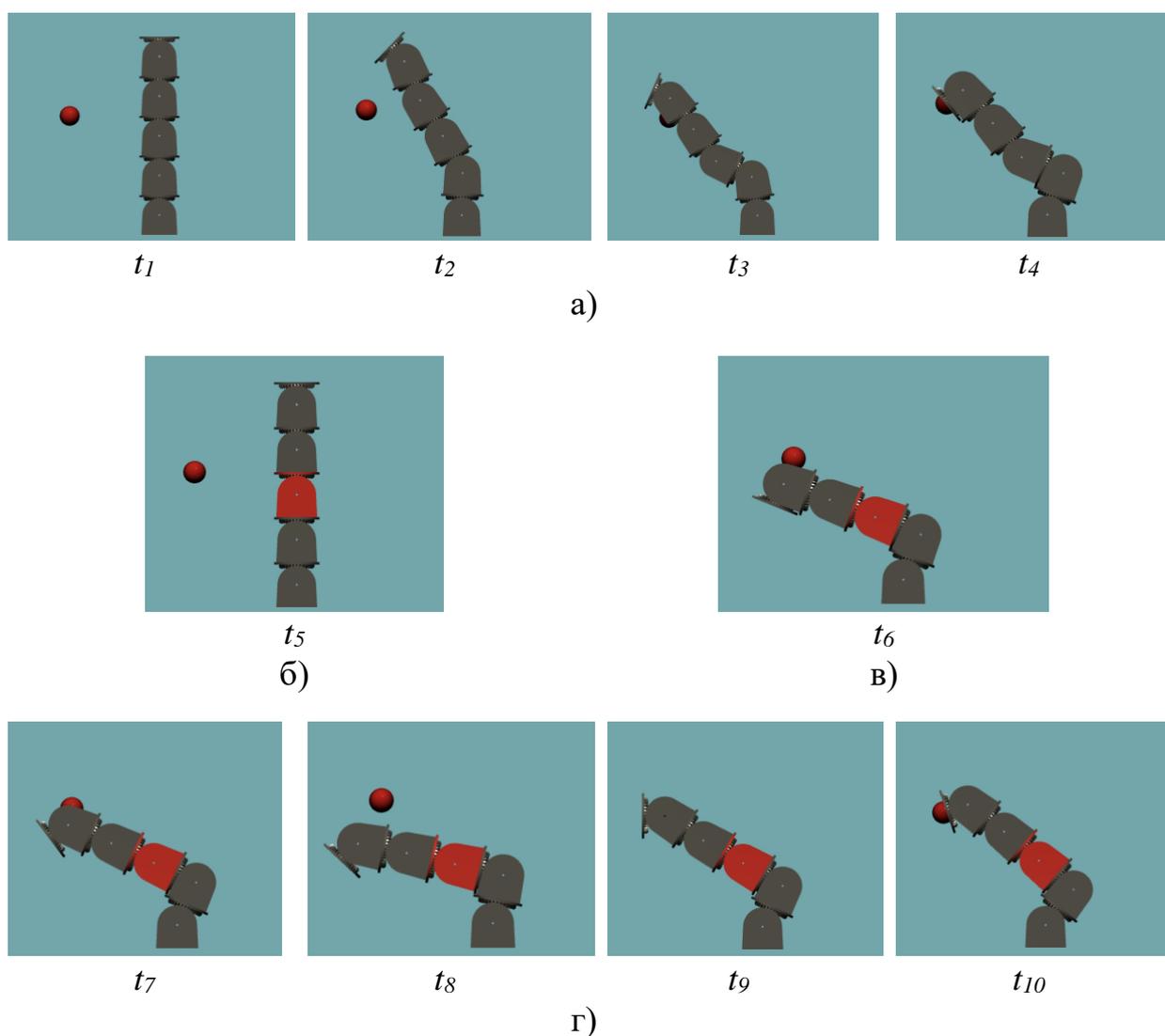


Рис. 4.1.6. Демонстрация способности переобучения модульного манипулятора (1-й способ) : а) БЗ робота пустая, и робот самообучается приходить в целевую точку; б) имитируется выход из строя (заклинивание) одного из модулей; в) робот промахивается, пытаясь переместить схват в целевую точку, используя полученные знания; г) робот переобучается и в результате приходит в целевую точку, несмотря на поломку одного из модулей

2-ой способ управления в отличие от предыдущего требует меньше памяти, поскольку количество образов существенно ниже. На рис. 4.1.7. показан процесс обучения робота при использовании данного способа. Можно видеть, что робот постепенно обучается выполнять действия для передвижения в целевую точку. Из-за малого количества образов скорость обучения при использовании 2-го способа выше, чем при использовании 1-го.

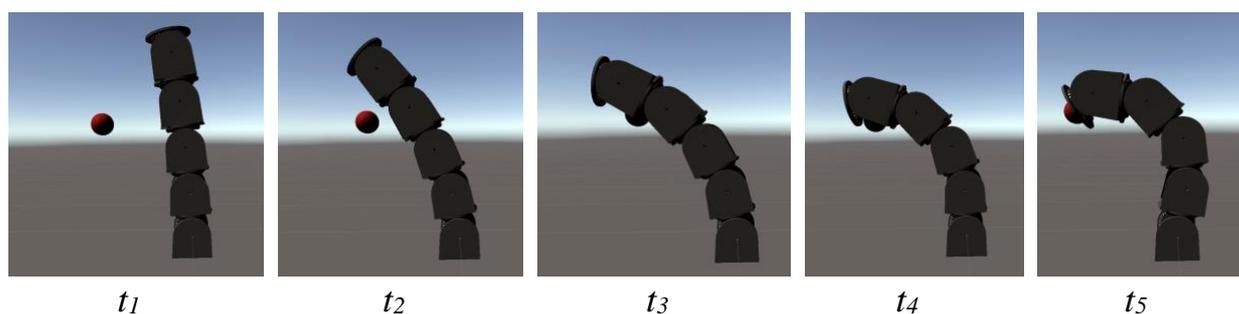


Рис. 4.1.7. Демонстрации обучения управлению модульным манипуляционным роботом на основе метода ААУ (2-ой способ).

Как было сказано выше, данные способы управления могут применяться к МР в конфигурации шагающей платформы. При этом задача декомпозируется на 2 этапа. Первый предполагает нахождение целевого состояния конечности, рассматривая конечность как манипулятор. На следующем этапе ищется наиболее эффективный порядок перестановки конечностей.

Для платформы с 4-мя конечностями, приведенной для примера на рис.4.1.8., не все варианты осторожной походки являются равнозначными с точки зрения поддержания устойчивости. Так, при последовательном перемещении конечностей по схеме «1-2-3-4» изменение положения центра тяжести платформы на стадии выполнения 1-го шага, приводит к ее опрокидыванию. В то же время другой вариант походки с перемещением конечностей по схеме «4-3-1-2» позволяет сохранить устойчивость шагающей платформы в процессе ее движения (рис. 4.1.9).

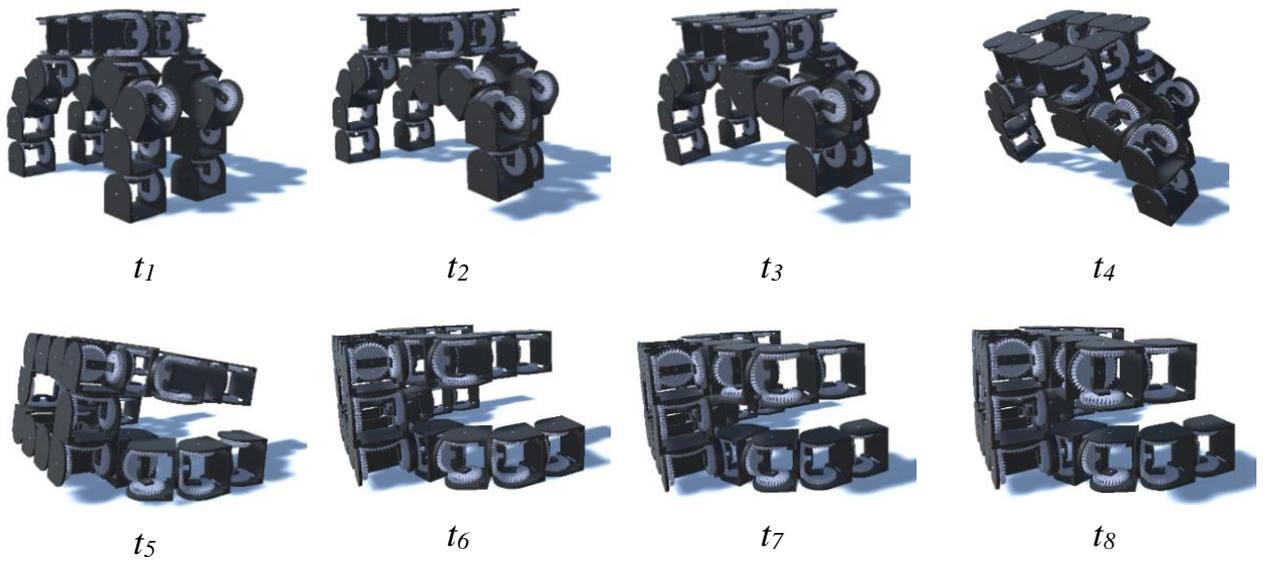


Рис. 4.1.8. Моделирование движения шагающей платформы по принципу «осторожной походки» по схеме «1-2-3-4», при которой происходит потеря устойчивости.

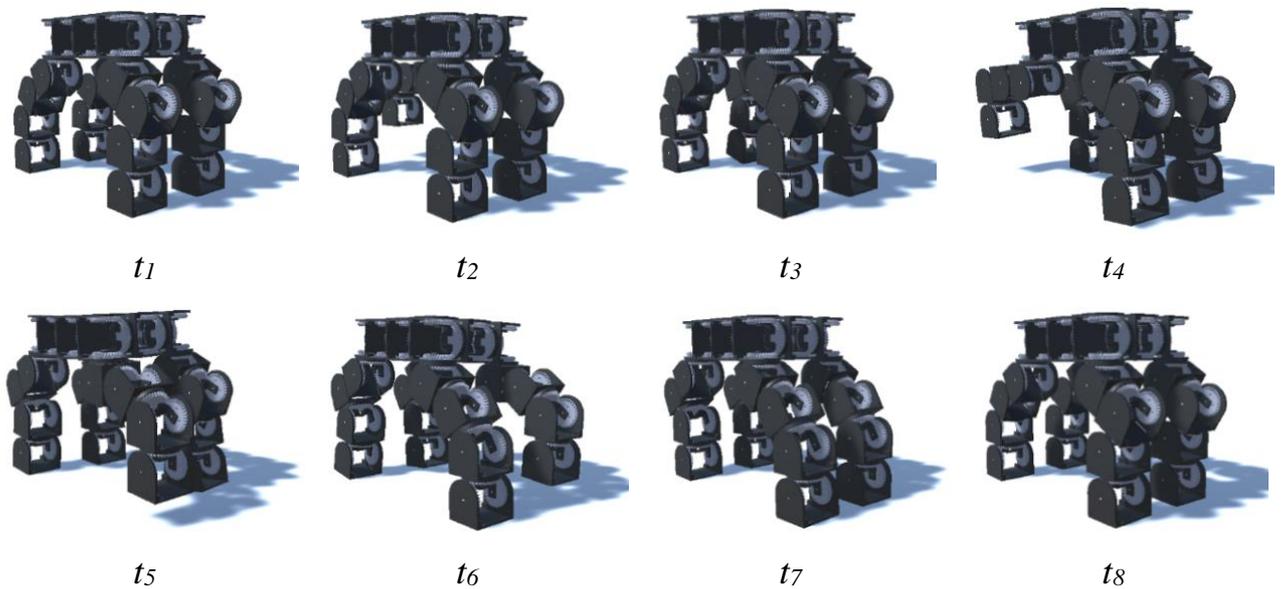


Рис. 4.1.9. Моделирование движения шагающей платформы по принципу «осторожной походки» по схеме «4-3-1-2», при которой не происходит потеря устойчивости.

Поскольку существует возможность опрокидывания платформы, то программное обеспечение, которое должно закладываться на борт модульной шагающей платформы, должно иметь частично заполненную БЗ. Для этого, сначала, необходимо провести обучение методом ААУ на моделях, а затем перенести полученную в результате обучения БЗ на борт робота.

Таким образом, проведенные экспериментальные исследования свидетельствуют о работоспособности метода ААУ применительно к управлению МР в конфигурациях «манипулятор» и «шагающая платформа». Показаны процесс самообучения модульных роботов и переобучения в случае выхода из строя отдельного модуля.

#### **4.2. Преимущества предложенных алгоритмов адаптивного управления модульными роботами по сравнению с другими методами управления**

Подводя итоги работы, стоит сравнить предложенный способ управления с уже существующими подходами.

Традиционно, при управлении манипуляционными роботами решается обратная задача кинематики. Но получение точных аналитических решений становится практически невозможным с увеличением количества степеней свободы. Метод ААУ не использует аналитических решений обратной задачи. Однако 1-й способ применения метода ААУ, предложенный в параграфе 2.2 и экспериментально проверенный в параграфе 4.1. фактически получает точное автоматическое решение обратной задачи кинематики в виде образов системы ФРО, независимо от количества степеней свободы. Кроме того, это позволяет решать задачу планирования траектории движения, как это происходит в методе RRT [29]. Однако метод RRT не имеет адаптивных возможностей и в

случае выхода из строя одного из модулей, нужно полностью перестраивать его дерево решения.

Численные методы, такие как метод покоординатного спуска в его геометрической интерпретации и метод градиентного спуска, [76] позволяют получить приблизительное решение. Однако, в ряде случаев, при определенных взаимных расположениях целевой точки и манипулятора данные алгоритмы не работают, поскольку не обеспечивают сходимость решения. 2-й способ применения метода ААУ, также предложенный в параграфе 2.2 и экспериментально проверенный в параграфе 4.1, за счет возможности совершения случайных действий с небольшой вероятностью (аналог оператора мутации в генетическом алгоритме) позволяет выходить из области локальных экстремумов и обеспечивает сходимость решения.

Для решения задачи генерации походки шагающих МР находят применение самообучаемые автоматы [31], генетические алгоритмы [94] и обучение с подкреплением [28]. Последние два способа, несмотря на их достоинства, обладают существенным недостатком, заключающимся в необходимости полного переобучения, в случае изменения параметров среды или выхода из строя отдельных модулей. Кроме того, работа данных методов в режиме реального времени, в отличие от предложенного подхода, невозможна.

Подход на основе самообучающихся автоматов [31] обеспечивает возможность переобучения, однако, за счет полного исследования рабочей зоны данный метод достаточно медленный. Предложенный подход на основе метода ААУ обеспечивает обучение в 10-100 раз быстрее (в случае 4-звенной конечности), чем подход, основанный на конечных автоматах. С увеличением количества модулей в кинематической цепи, разрыв во времени обучения будет еще больше.

Стоит еще раз отметить, что оба предложенных способа, основанные на методе ААУ, в отличие от всех вышеописанных методов обладают

возможностью переобучения в процессе функционирования МР, тем самым адаптируясь к таким изменениям, как выход из строя какого-либо модуля или изменения направления тока в обмотке двигателя. Кроме того, процесс управления неразрывен с процессом обучения, что позволяет работать в режиме реального времени и переобучаться в случае выхода из строя отдельных модулей.

### **Выводы по главе**

В данной главе представлены результаты проведенных программных экспериментов по исследованию работоспособности и эффективности применения метода Автономного Адаптивного Управления к управлению модульными роботами на разработанном программном обеспечении. Экспериментально исследованы 2 способа адаптивного управления, при которых модульный манипуляционный робот непосредственно в процессе управления постепенно и успешно самообучается точнее перемещать схват в заданную точку. Проведены эксперименты по работоспособности предложенного подхода в случае выхода из строя одного из модулей, включая процессы самообучения при изначально пустой Базе Знаний, и процесс переобучения при частично заполненной Базе Знаний.

Также представлены результаты успешных экспериментов по исследованию адаптивного управления движением модульных роботов в конфигурации «шагающей платформы». При этом конечности робота рассматривались как совокупность манипуляторов.

Описаны преимущества предложенных способов адаптивного управления по сравнению с существующими методами и алгоритмами управления манипуляционными и шагающими модульными роботами,

которые состоят в возможности адаптации к различного рода изменениям свойств работа или окружающей среды. Контур обучения работает одновременно с контуром управления, что позволяет самообучаться и переобучаться (например, в случае выхода из строя отдельных модулей) в режиме реального времени.

## ЗАКЛЮЧЕНИЕ

Современные программные комплексы и системы управления активно применяют методы искусственного интеллекта. Одним из перспективных подходов, на наш взгляд, является метод Автономного Адаптивного Управления, отличительные особенности которого заключаются в возможности обучения и переобучения одновременно с процессом управления, в постепенном автоматическом повышении качества управления, обусловленном автоматическим учетом реальных текущих свойств объекта управления, в возможности дообучения и переобучения непосредственно в процессе управления.

Диссертационная работа была посвящена проблемам алгоритмического и программного обеспечения адаптивной системы управления многозвенными объектами на примере модульных роботов. Модульная робототехника сегодня является активно развивающейся областью как теоретических, так и практических исследований. Благодаря своим функциональным возможностям, области применения модульных роботов очень широки – начиная с бытовой и учебной сферы и заканчивая задачами исследования космоса. Современные алгоритмы управления модульными роботами, как показывает проведенный в 1-ой главе работы анализ публикаций, не обеспечивают адаптации к изменяющимся параметрам среды или различного вида поломкам, например, выходу из строя одного из модулей. Поэтому необходимо привлечение методов искусственного интеллекта. В этом классе алгоритмов наибольший интерес представляет метод Автономного Адаптивного Управления А.А. Жданова, позволяющий повысить функциональные возможности модульных роботов, а именно, возможности адаптации и самообучения непосредственно в процессе функционирования. Применение метода Автономного

Адаптивного Управления неразрывно связано с разработкой способа построения адаптивной системы управления модульными роботами.

Также в 1-й главе был проведен сравнительный анализ современных средств моделирования модульных роботов, который показал необходимость разработки нового способа построения программного обеспечения для моделирования модульных роботов, организацию взаимодействия моделирующего комплекса непосредственно с физическим образцом МР, а также применение в основе алгоритмов управления метода Автономного Адаптивного Управления.

Во 2-ой главе настоящей работы изложены разработанные способ построения адаптивной системы управления модульными роботами и математическая модель, упрощенно описывающая работу метода Автономного Адаптивного Управления А.А. Жданова, а также описаны разработанные алгоритмы адаптивного управления роботами в конфигурациях «манипулятор» и «шагающая платформа».

В 3-ей главе представлен способ построения программного обеспечения для моделирования модульных роботов. Описано разработанное на основе предложенного способа программное обеспечение, его основные компоненты, включая программную реализацию метода Автономного Адаптивного Управления применительно к модульным роботам. Также предложен способ описания конфигураций модульных роботов и алгоритмы автоматического построения программных моделей по файлу описания, описано программное взаимодействие средств моделирования и разработанного макетного образца модульного робота. Предложенное программное обеспечение позволяет одновременно управлять как программными моделями, так и реальным роботом, верифицировать созданные программные модели и алгоритмы.

4-я глава посвящена экспериментальным исследованиям на разработанном программном обеспечении. Проведенные эксперименты

подтверждают работоспособность и эффективность разработанных адаптивных алгоритмов управления движением модульных роботов в конфигурациях «манипулятор» и «шагающая платформа». В ходе экспериментов показаны возможности обучения и переобучения модульного робота. Также в 4-й главе описаны преимущества предложенных методов адаптивного управления по сравнению с существующими методами управления.

В ходе выполнения диссертационной работы получены следующие результаты:

1. разработан способ построения адаптивной системы управления многозвенными объектами на примере модульных роботов;
2. предложена математическая модель, упрощенно описывающая работу системы, построенной на основе метода Автономного Адаптивного Управления;
3. разработан способ построения программных средств моделирования модульных роботов;
4. разработано программное обеспечение для моделирования модульных роботов;
5. предложен способ описания конфигураций модульных роботов и алгоритм автоматического построения программных моделей по файлу описания;
6. разработан способ взаимодействия средств программного моделирования с разработанным макетным образцом модульного робота;
7. разработано и экспериментально проверено алгоритмическое и программное обеспечение адаптивной системы управления модульными роботами в конфигурациях «манипулятор» и «шагающая платформа».

На защиту выносятся следующие положения :

1. Способ построения адаптивной системы управления модульными роботами.
2. Способ построения программного обеспечения для моделирования модульных роботов.
3. Алгоритмы адаптивного управления модульными роботами в конфигурациях «манипулятор» и «шагающая платформа».
4. Способ описания конфигураций модульных роботов и алгоритм автоматического построения 3-мерной модели роботов по файлу описания.

Полученные результаты могут быть применены при разработке модульных роботов различного типа и назначения. Предложенные способ построения адаптивной системы управления, алгоритмы адаптивного управления, способ описания конфигураций и способ построения программного обеспечения для моделирования модульных роботов могут быть внедрены при проектировании модульных робототехнических систем. Разработанное программное обеспечение для моделирования модульных роботов можно применять в учебных и научно-исследовательских целях.

Дальнейшие перспективы данного исследования связаны с тремя направлениями. Первое из них – это развитие метода Автономного Адаптивного Управления применительно к другим объектам управления. Второе – развитие предложенной адаптивной системы управления модульными роботами, реализация адаптивных возможностей в других конфигурациях модульного робота и на других уровнях системы управления, а также её применение для других классов многозвенных конструкций. Кроме того, потенциальным развитием может быть комбинация двух предложенных способов управления модульными роботами с целью объединения их преимуществ и нивелирования

недостатков. Третье – переход к групповому, распределенному управлению модульными роботами.

В заключение я хочу выразить огромную благодарность своему учителю Александру Аркадьевичу Жданову за научное руководство, оказанную помощь и поддержку, а также Марине Анатольевне Шестаковой за ценные замечания по тексту диссертации. Кроме того, выражаю благодарность и признательность сотрудникам кафедры проблем управления РТУ МИРЭА и лично Сергею Викторовичу Манько.

**СПИСОК ЛИТЕРАТУРЫ**

1. A drone program taking flight. [Электронный ресурс]. — URL: <https://blog.aboutamazon.com/transportation/a-drone-program-taking-flight> (Дата обращения 10.12.2020)
2. *Vaca J., Ferre M. & Aracil R.* A heterogeneous modular robotic design for fast response to a diversity of tasks // *Robotics and Autonomous Systems*. 2012. Vol. 60. No. 4. P. 522 – 531.
3. *Bongard J. C.* Evolutionary Robotics // *Communications of the ACM*. 2013. Vol. 56. No 8. pp. 74-83.
4. *Christensen D., Brandt D., Stoy K., Schultz U.* A unified simulator for self-reconfigurable robots // *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008. pp. 870–876.
5. *Cohen M. M.* Mobile lunar and planetary bases // in proc 2003 AIAA Space Conference, San Diego, 2004.
6. *Collins, T., Ranasinghe, N. O., Wei-Min Shen.* ReMod3D: A high-performance simulator for autonomous, self-reconfigurable robots // *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013.
7. *Demin A.V., Vityaev E.E.* Adaptive Control of Modular Robots // *Proc. Of First International Early Research Career Enhancement School on Biologically Inspired Cognitive Architectures*, 2018.
8. *Ercan H., Boyraz P.* Design of a Modular Mobile Multi Robot System: ULGEN (Universal-Generative Robot) // *Proceedings of Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. 2016. pp. 8–15.

9. *Everist, J., Shen, W. M.* Comparative Study of Locomotion Methods for Modular and Self-Reconfigurable Robots. In In Proc. of Eighth International Symposium on Distributed Autonomous Robotic Systems, DARS 2006
10. *Fukuda T., Ueyama T., Kawauchi Y. & Arai F.* Concept of cellular robotic system (CEBOT) and basic strategies for its realization // Computers Elect Engng. 1987. Vol. 18. No. 1. P. 11 – 39.
11. *Garcia R.F.M., Lyder A., Christensen D.J. & Stoy K.* Reusable Electronics and Adaptable Communication as Implemented in the Odin Modular Robot // IEEE International Conference on Robotics and Automation. 2009. P. 1152 – 1158.
12. *Geoffrey Brown.* Discovering the STM32 Microcontroller // Indiana University, 2016. – 244p.
13. *Golovinsky A., Yim M., Ying Zhang, Eldershaw C. and Duff D.* PolyBot and PolyKinetic/spl trade/ System: a modular robotic platform for education // IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 1381-1386 Vol.2, doi: 10.1109/ROBOT.2004.1308017.
14. *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
15. *Gonzalez-Gomez J., Gonzalez I., Gomez-Arribas FJ. y Boemo E.* Evaluación de un Algoritmo de Locomoción de Robots Ápodos en Diferentes Procesadores Embebidos en FPGA // V Jornadas de Computacion Reconfigurable y Aplicaciones. JCRA 2005. Dentro del Primer Congreso Espanol de Informatica, CEDI 2005. Granada, Septiembre 2005.
16. *González-Gómez, J, Aguayo E. y Boemo E.* Locomoción de un Robot Ápodo Modular con el Procesador MicroBlaze // IV Jornadas sobre Computación

Reconfigurable y Aplicaciones, JCRA04, Escuela Técnica Superior de Ingenierías. Universidad Autónoma de Barcelona, Septiembre 2004.

17. *González-Gómez, J. G. et al.* “Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules.// 9th International Conference on Climbing and Walking Robots. CLAWAR06. Brussels, September 2006.
18. *Haarnoja T., Aurick Zhou, Sehoon Ha, Tan J., Tucker G. and Levine S.* Learning to Walk via Deep Reinforcement Learning, ArXiv abs/1812.11103 – 2019.
19. *Hancher M.D. & Hornby G.S.* A modular robotic system with applications to space exploration // 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06). Pasadena, CA: Publisher «IEEE», 2006. P. 132 – 140.
20. *Harada, K., Susilo, E., Watanabe, T., Kawamura, K., Fujie, M., Menciassi, A., & Dario, P.* Modular Robotic Approach in Surgical Applications – Wireless Robotic Modules and a Reconfigurable Master Device for Endoluminal Surgery, 2012
21. *Ijspeert A.J., Crespi A., Cabelguen J.-M.* Simulation and Robotics Studies of Salamander Locomotion: Applying Neurobiological Principles to the Control of Locomotion in Robots // Neuroinformatics, Volume 3, Issue 3, pps. 171-196, Fall 2005.
22. *Jing G., Tosun T., Yim M., Kress-Gazit H.* Accomplishing high-level tasks with modular robots // Autonomous Robots, 2018. vol. 42, pp 1337-1354.
23. *Joshua P. Davis, John P. Mayberry, and Jay P. Penn.* On-Orbit Servicing: Inspection, Repair, Refuel, Upgrade, and Assembly of Satellites in Space // The Aerospace Corporation, April 2019.

24. *Kamimura A., Kurokawa H., Yoshida E., Murata S., Tomita K. and Kokaji S.* Automatic Locomotion Pattern Generation for Modular Robots // Proc. IEEE Int. Conf. on Robotics and Automation (ICRA2003), pp. 714–720, 2003
25. *Kamimura A., Kurokawa H., Yoshida E., Murata S., Tomita K. and Kokaji S.* Distributed Adaptive Locomotion by a Modular Robotic System, M-TRAN II From Local Adaptation to Global Coordinated Motion Using CPG Controllers // Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems doi: [10.1109/IROS.2004.1389763](https://doi.org/10.1109/IROS.2004.1389763)
26. KUKA omniMove. [Электронный ресурс]. — URL: <https://www.kuka.com/en-gb/products/mobility/mobile-platforms/kuka-omnimove> (Дата обращения 10.12.2020)
27. *Lawrence S. J., G. Taylor, R. Lentz, L. M. Martel, Wei Shen, P. Will, M. Sims, S. Colombano, D. Kortenkamp, B. Damer and W. Chun.* Superbots on the Lunar Surface: A Habitat Operations and Maintenance System (HOMS). (2005).
28. *Lee J., Oh J.* Walking Pattern Generation for Planar Biped Using Q-Learning, Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC), IFAC, Luxenburg, Austria, July, 2008, pp. 3027-3032.
29. *Liang Y., Mu H., Chen D., Wei X. and Wang M.* PR-RRT\*: Motion Planning of 6-DOF Robotic Arm Based on Improved RRT Algorithm, 2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Xi'an, 2020, pp. 417-422, doi: [10.1109/CYBER50695.2020.9279164](https://doi.org/10.1109/CYBER50695.2020.9279164).
30. *Lyder A.H., Stoy K., Mendoza-Garcia R.F., Larsen J.C. & Hermansen P.* On sub-modularization and morphological heterogeneity in modular robotics // Intelligent Autonomous Systems of Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg. 2013. Vol. 193. No. 12. P. 649 – 661.

31. *Manko S., Shestakov E.* Automatic Synthesis Gait Scenarios for Reconfigurable Modular Robots Walking Platform Configuration // In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. ICR 2018. Lecture Notes in Computer Science, vol 11097. Springer, Cham
32. *Matsuoka K.* Mechanisms of frequency and pattern control in the neural rhythm generators // *Biolog. Cybern.*, 56, pp. 345–353, 1987.
33. *McEvoy M., Strzpek J.* Robotic Assembly of Modular Robots in Simulation. 2012.] или V-Rep [Ercan H., Boyraz P. Design of a Modular Mobile Multi Robot System: ULGEN (Universal-Generative Robot) // Proceedings of Asia-Pacific Conference on Intelligent Robot Systems (ACIRS). 2016. pp. 8–15.
34. *Moeckel R., Perov Y., The Nguyen A., Vespignani M., Bonardi S., Pouya S., Sproewitz A., Van den Kieboom J., Wilhelm F., Ijspeert A. J.* Gait optimization for roombots modular robots - matching simulation and reality // Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. pp. 3265– 3272.
35. *Moll M., Will P., Krivokon M., Shen, W.* Distributed Control of the Center of Mass of a Modular Robot // Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-06). 2006. pp. 4710-4715.
36. *Murata S., Yoshida E., Kamimura A., Kurokawa H., Tomita K. & Kokaji S.* M-TRAN: selfreconfigurable modular robotic system // IEEE/ASME Transactions on Mechatronics. 2002. Vol. 7 (4). P. 432 – 441.
37. *Østergaard E.H., Kassow K., Beck R. & Lund H.H.* Design of the ATRON lattice-based self-reconfigurable robot // *Autonomous Robots*. 2006. Vol. 21 (2). P. 165 – 183.
38. *Oung, R., Bourgault, F., Donovan, M., & D'Andrea, R.* (2010). The Distributed Flight Array. 2010 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2010.5509882

39. *Qiao G., Song G., Zhang J., Sun H., Wang W. & Song A.* Design of Transmote: a Modular Self-Reconfigurable Robot with Versatile Transformation Capabilities // Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics. 2012. P. 1331 – 1336.
40. *Reesman R.* Assurance Through Insurance and On-Orbit Servicing // The Aerospace Corporation, February 2018 (<https://aerospace.org/sites/default/files/2018-05/OnOrbitServicing.pdf>)
41. *Romanov A.M., Romanov M.P., Shestakov E.I.* A novel architecture for control systems of modular reconfigurable robots // Proceedings of 2017 IEEE 2nd International Conference on Control in Technical Systems, CTS 2017 2. 2017. pp. 131-134.
42. *Ronald E., Schoenauer M.* Genetic lander: An experiment in accurate neuro-genetic control. In: Davidor Y., Schwefel HP., Männer R. (eds) Parallel Problem Solving from Nature — PPSN III. PPSN 1994. Lecture Notes in Computer Science, vol 866. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-58484-6\\_288](https://doi.org/10.1007/3-540-58484-6_288)
43. *Shen W., Salemi B., Will P.* Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. IEEE Trans. Robotics Autom., 18. pp. 700 – 712, 2002.
44. *Sproewitz A., Asadpour M., Bourquin Y., Ijspeert A.* An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching. in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, May 2008, pp. 3508 –3513.
45. *Stanley K.O. and Miikkulainen R.* Evolving neural networks through augmenting topologies // Evolutionary Computation 10(2):99-127, 2002.
46. *Stanley K.O., D'Ambrosio D.B., Gauci J.* A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks // Artificial Life. 15 (2): 185–212, 2006

47. *Støy K., Shen W.-M. and Will P.* Global Locomotion from Local Interaction in Self-Reconfigurable Robots // *Intelligent Autonomous Systems 7*, IOS Press, 2002
48. Thinkerbots – Roboter bauen & programmieren – Roboter für den Unterricht [Электронный ресурс]. — URL: <https://www.tinkerbots.de/?lang=en> (Дата обращения 10.12.2020)
49. *Valsalam V. K. Miikkulainen R.* Modular neuroevolution for multilegged locomotion // *In Proceedings of GECCO. 2008.* pp. 265–272
50. *Vonásek V., Fišer D., Košnar K., Přeučil L.* A Light-Weight Robot Simulator for Modular Robotics // *Proc. of International Workshop on Modelling and Simulation for Autonomous Systems MESAS 2014: Modelling and Simulation for Autonomous Systems, 2014,* pp. 206-216.
51. *Wang X., Zhu Y., Zhao J.* A Dynamic Simulation and Virtual Evolution Platform for Modular Self-reconfigurable Robots // *Proceedings of IEEE International Conference on Information and Automation (ICIA). 2013.* pp. 457–462
52. *Weghe, M.V., Ferguson, D. and Srinivasa, S.S.* Randomized path planning for redundant manipulators without inverse kinematics. In *2007 7th IEEE-RAS International Conference on Humanoid Robots, 2007,* pp. 477-482
53. *Winkler L., Vonasek V., Worn H., Preucil L.* Robot3D - A Simulator for Mobile Modular Self-Reconfigurable Robots // *Proceedings of IEEE International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI). 2012.* pp. 464 – 469
54. *Yim M., Duff D., and Roufas K.* PolyBot: A modular reconfigurable robot. // *In Proc. of the IEEE Int. Conference on Robotics and Automation, 2000.*

55. *Yim M., Roufas K., Duff D. et al.* Modular Reconfigurable Robots in Space Applications // *Autonomous Robots* 14, 225–237 (2003).  
<https://doi.org/10.1023/A:1022287820808>
56. *Zhang Y., Yim M., Eldershaw C., Duff D., Roufas K.* Phase Automata: a programming model of locomotion gaits for scalable chain-type modular robots // *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, USA, 2003.*
57. *Zhu Y., Zhao J., Cui X. et al.,* "Design and implementation of ubot: A modular self-reconfigurable robot," in *Proc.2013*, pp.1217-1222.
58. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. / [Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест и др.] – М.: Издательский дом «Вильямс», 2011. – 1296 с.
59. *Афанасьев В.Н., Колмоновский В.Б., Носов В.Р.* Математическая теория конструирования систем управления.// М.: «Высшая школа», 1998.
60. *Беккер, М.Г.* Введение в теорию систем местность – машина / М.Г. Беккер – М.: Машиностроение, 1973. – 520 с.
61. *Боронников А.Б.* Место XML-технологий в среде современных информационных технологий // *Программные продукты и системы*, № 2 2005г.
62. *Варшавский В.И., Поспелов Д.А.* Оркестр играет без дирижера: размышления об эволюции некоторых технических систем и управления ими. / В.И. Варшавский, Д.А. Поспелов – М.: Наука, 1984. – 208 с
63. *Голубева Л.А., Ильин В.П., Козырев А.Н.* О программных технологиях в геометрических аспектах математического моделирования. УДК 519.632 *Вестник НГУ. Серия: Информационные технологии Т.10 Вып.2 2012*

64. *Горелик А.Л., Гуревич И.Б., Скрипкин В.А.* Современное состояние проблемы распознавания. // М.: Радио и связь, 1985, 160 с.
65. *Гренандер У.* Лекции по теории образов. // М.: Мир, 1979, 1 том; 1981, 2 том; 1983, 3 том.
66. Демин А. В. Адаптивное управление роботами с модульной конструкцией // Системы управления, связи и безопасности — №4, 2015.
67. *Жданов А. А.* Автономный искусственный интеллект: Монография. 2-е изд. / А.А. Жданов – М.: БИНОМ. Лаборатория знаний, 2009. – 359 с.
68. *Жданов А. А.* Метод автономного адаптивного управления. // Известия Академии Наук. Теория и системы управления, 1999, № 5, с. 127-134.
69. *Жданов А.А., Земских Л.В., Беляев Б.Б.* Применение генетических алгоритмов для оптимизации нейросетевой базы знаний адаптивной системы стабилизации углового движения космического аппарата. // Сб. тезисов докладов XXV академических чтений по космонавтике, М: Война и мир», 2001 – с. 128-129.
70. *Жданов А.А.* Метод автономного адаптивного управления. // Известия Южного федерального университета. Технические науки 38.3 (2004).
71. *Жданов А.А.* О понятии автономного искусственного интеллекта // Сб. научн. тр. Искусственный интеллект в технических системах. – М.: Гос. ИФТП 1997.
72. *Жданов А.А., Земских Л.В., Беляев Б.Б.* Система стабилизации углового движения космического аппарата на основе нейроподобной системы автономного адаптивного управления // Космические исследования. 2004. Т. 42. № 3. С. 280-294.

73. *Жданов А.А., Липкевич Д.Б.* ADCAS - система автономного адаптивного управления активной подвеской автомобиля // Труды Института системного программирования РАН. 2004. Т. 7. С. 119-160.
74. *Журавлев Ю.И.* Об алгебраическом подходе к решению задач распознавания и классификации. // Проблемы кибернетики. М.: Наука. 1978.-вып.33. – с.5-68.
75. *Загоруйко Н.Г.* Прикладные методы анализа данных и знаний. // Новосибирск. Изд-во института математики. 1999.
76. Интеллектуальные системы автоматического управления / под ред. И.М. Макарова, В.М. Лохина – М.: ФИЗМАТЛИТ, 2001. – 576 с.
77. Информационные устройства робототехнических систем: Учеб. пособие / С.А. Воротников – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005 – 384 с.
78. Искусственный интеллект и интеллектуальные системы управления / И.М. Макаров, В.М. Лохин, С.В. Манько, М.П. Романов – М.: Наука, 2006. – 333 с
79. Искусство схемотехники: Пер. с англ. – Изд. 7-е. / П. Хоровиц, У. Хилл – М.: Издательство БИНОМ, 2012. – 704 с.
80. *Казаков К.А., Семенов В.А.* Обзор современных методов планирования движения // Труды ИСП РАН, 2016 г., с. 241-294.
81. *Квашнин В. В., Бабаиш А. В.* и др. Управление векторным электроприводом механической системы с переменными массами // Проблемы региональной энергетики. 2019. №S1-3 (42).
82. *Кибардин В.В., Ковалева О. А., Язев В.Н.* Критерии оптимального управления и lqr-оптимизация в электроприводе // Вестник КрасГАУ. 2015. №12.

83. *Лапшин В. В.* Механика и управление движением шагающих машин / В. В. Лапшин. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2012,— 199 с.
84. *Лебедев А.Е., Жданов А.А.* Динамическая сегментация пространства признаков для систем автономного адаптивного управления и систем обучения с подкреплением // XII Всероссийская научно-техническая конференция "Нейроинформатика - 2010": Сборник научных трудов. В 2-х частях. Ч.1. М.: НИЯУ МИФИ, 2010. с.182-190.
85. Лохин В.М., Манько С.В., Диане С.А.К., Шестаков Е.И. Использование аппарата конечных автоматов в задачах группового управления автономными роботами. В книге: Фундаментальные проблемы группового взаимодействия роботов. Материалы отчетного мероприятия РФФИ по конкурсу "офи-м" (тема 604) в рамках международной научно-практической конференции. 2018. С. 24-25.
86. Лохин В.М., Манько С.В., Шестаков Е.И., Диане С.А.К. Комплексное применение аппарата теории конечных автоматов в задачах группового управления автономными роботами. В книге: ДЕСЯТАЯ ВСЕРОССИЙСКАЯ МУЛЬТИКОНФЕРЕНЦИЯ ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ (МКПУ-2017). Материалы 10-й Всероссийской мультikonференции в 3-х томах. Ответственный редактор: И.А. Каляев. 2017. С. 297-299.
87. Лохин В.М., Манько С.В., Шестаков Е.И., Малько А.Н. Автоматический синтез сценариев походки мехатронно-модульных роботов в конфигурации шагающей платформы. Свидетельство о регистрации программы для ЭВМ № RU 2018619700.
88. *Макаров И.М.* Управление движением мехатронно-модульных роботов с адаптивной кинематической структурой [Текст] / И.М. Макаров, В.М. Ло-

- хин, С.В. Манько, М.П. Романов, М.В. Кадочников // Мехатроника, автоматизация, управление. – No3. – 2008.
89. Манипуляционные системы роботов: основы устройства, элементы теории. / Кобринский А.А., Кобринский А.Е. – М.: Наука, 1985. – 344с.
90. *Манько С.В., Шестаков Е.И.* Автоматический синтез сценариев походки реконфигурируемых мехатронно- модульных роботов в модификации шагающей платформы // Российский технологический журнал. 2018 т.6 No4, с. 26-41
91. Матрично-геометрические методы в механике с приложениями к задачам робототехники. / Величенко В.В. – М.: Наука, 1988. – 280с.
92. Методы, модели и алгоритмы группового управления автономными роботами на основе комплексного применения аппарата теории конечных автоматов: Итоговый отчет о НИР / Лохин В.М. – Москва, РТУ МИРЭА, 2019 – 98 с.
93. Механика и управление движением автоматического шагающего аппарата / Охоцимский Д.Е., Голубев Ю.Ф. — М.: Наука, 1984.
94. Мехатронно-модульные роботы с адаптивной кинематической структурой / М.В. Кадочников – LAP Lambert Academic Publishing, 2012.
95. Нейронные сети. Учебное пособие. / Е.И. Горожанина. – Самара. ФГБОУ ВО ПГУТИ, 2017. – 84 с.
96. Обзор мирового опыта коммерческой доставки грузов с помощью беспилотников. [Электронный ресурс]. — URL: <https://habr.com/ru/post/402475> (Дата обращения 10.12.2020)
97. Основы кибернетики: В 2-х т. Т.1. Математические основы кибернетики. Учеб. пособие для студентов вузов. / Л.Т. Кузин – М.: Энергия, 1973 – 504 с.

98. Основы кибернетики: В 2-х т. Т.2. Основы кибернетических моделей. Учеб. пособие для студентов вузов. / Л.Т. Кузин – М.: Энергия, 1979 – 584 с.
99. От амёбы до робота: модели поведения / М.Г. Гаазе-Рапопорт, Д.А. Поспелов – М.:Едиториал УРСС, 204 – 296с.
100. *Панченко Т. В.* Генетические алгоритмы [Текст] : учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. — Астрахань : Издательский дом «Астраханский университет», 2007
101. Программирование на языке Си: учеб. пособие / В.В. Подбельский, С.С. Фомин – 2-е изд., перераб. и доп. М.: Финансы и статистика, 2009. – 660с.
102. *Рапопорт А.Н.* Автоматные модели поисковой оптимизации и управления. Киров: ВятГТУ, 1999. УДК 519.863+681.5
103. Роботы-змеи будут заниматься инспекцией АЭС. [Электронный ресурс]. — URL: <http://www.fainaidea.com/technologii/roboty-zmei-budut-zanimatsya-inspekts-30711.html> (Дата обращения 10.12.2020)
104. *Романов А.М., Манько С.В., Шестаков Е.И., Малько А.Н., Чуу В.-Ю.* Способы описания и средства моделирования мехатронно-модульных реконфигурируемых роботов // Труды ФГУП" НПЦАП". Системы и приборы управления. – 2019. – No. 2
105. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия – Телеком, 2004
106. C# 4.0: полное руководство.: Пер. с англ. / Г. Шилдт – М.: ООО «И.Д. Вильямс», 2012. – 1056 с.
107. Самообучающиеся системы. / Николаенко С.Н., Тулупьев А.Л. – М.: МЦНМО, 2009. – 288 с.

108. *Семенов С.С., Жданов А.А., Романов А.А.* Использование метода «Автономного адаптивного управления» в многоуровневых управляющих системах
109. *Синюк К.В., Анчутин В.А.* Применение нечеткой логики для систем управления линейным синхронным двигателем с постоянными магнитами // Символ науки. 2018. №3.
110. *Стжелецки Р., Демидова Г. Л. и др.* Алгоритмы управления электромеханическими объектами с использованием регуляторов на основе нечеткой логики // Научно-технический вестник информационных технологий, механики и оптики, 2019, том. 19, №1, сс. 1–14. DOI: 10.17586/2226-1494-2019-19-1-1-14,
111. Структуры данных и алгоритмы.: Пер. с англ.: Уч. пос. / А.В. Ахо, Д. Хопкрофт, Д.Д. Ульман – М.: Издательский дом «Вильямс», 2010. – 400с.
112. Управление манипуляционными роботами: теория и приложения. / Вукобратович М. Стокич Д. – М.:Наука, 1985. – 384с.
113. *Ф.А. Новиков.* Дискретная математика: Учебник для вузов. 2-е изд. Стандарт третьего поколения. – СПб.: Питер, 213 – 432с.
114. *Хиросэ Ш.* Бионические роботы: змееподобные мобильные роботы и манипуляторы. – М.–Ижевск: Институт компьютерных исследований, 2014. – 272 с.
115. *Хокинг Дж.* Unity в действии. Мультиплатформенная разработка на C# / Пер. с англ. И. Рузмайкиной. — СПб.: Питер, 2016. — 336 с
116. *Цетлин М.Л.* Исследования по теории автоматов и моделированию биологических систем. М.: Наука, 1969. 316 с.
117. *Цыпкин Я. З.* Адаптация и обучение в автоматических системах, «Наука», М., 1968

118. *Чечкин А.В.* Математическая информатика. – М.: Наука. Гл. ред. физ.-мат. лит., 1991.
119. *Шестаков Е.И.* Построение распределенной системы управления мехатронно-модульным манипуляционным роботом // Современные технологии в задачах управления, автоматизации, и обработки информации: Труды XXIV Международной научно-технической конференции, 14-20 сентября 2015 г., Алушта. – М.: Издательский дом МЭИ, 2015. – с. 115-116
120. *Шестаков Е.И., Жданов А.А.* Адаптивное управление модульным реконфигурируемым манипуляционным роботом // XXII Международная научно-техническая конференция "Нейроинформатика-2020": Сборник научных трудов. М.: НИЯУ МИФИ, 2020 – с.18-26.
121. *Шестаков Е.И., Жданов А.А.* Управление манипуляционным роботом на основе метода автономного адаптивного управления // Нейронаука для медицины и психологии: XVI Международный междисциплинарный конгресс. Судак, Крым, Россия; 6–16 октября 2020 г.: Труды Конгресса / Под ред. Е.В. Лосевой, А.В. Крючковой, Н.А. Логиновой. – Москва: МАКС Пресс, 2020. – с. 527-529. ISBN 978-5-317-06406-8
122. *Шестаков Е.И., Манько С.В., Лохин В.М.* Автоматическое формирование моделей управляемого движения автономных реконфигурируемых роботов // В книге: Десятая Всероссийская мультikonференция по проблемам управления (МКПУ-2017) Материалы 10-й Всероссийской мультikonференции в 3-х томах. Ответственный редактор: И.А. Каляев. 2017. С. 128-131.