

Карабаза П.В. «Компьютерные методы моделирования сложных систем» – ПГГПУ, Пермь, 2017. – 52 с. + Прил.

В ходе исследования выявлена сущность и содержание кибернетического подхода к пониманию сложных систем.

Проведен анализ основ моделирования кибернетических систем в робототехнике, и выявлены основные признаки сложных роботизированных систем, такие как автономность, универсальность, автоматичность, антропоморфизм и адаптивность.

Выделены основные составляющие робота как системы, активно взаимодействующей с окружающей средой.

Разработана модель роботизированной адаптивной системы, которая самостоятельно решает задачу нахождения траектории движения с обходом препятствий от места старта до определенной цели. Разработанная модель обладает всеми признаками роботизированной системы. Данная разработка может быть представлена как один из проектов по образовательной или соревновательной робототехнике.

Содержание

Введение	3
ГЛАВА 1. Общетеоретические аспекты моделирования сложных систем ..	6
1.1. Кибернетический подход к пониманию сложных систем	6
1.2. Основы моделирования кибернетических систем в робототехнике ..	17
ГЛАВА 2. Разработка роботизированной адаптивной системы	27
2.1. Постановка задачи	27
2.2. Разработка адаптивного алгоритма	34
2.3. Программная реализация адаптивного алгоритма	43
Заключение	49
Список использованной литературы	50
Приложение 1	53
Приложение 2	54
Приложение 3	55

Введение

Стремительное развитие научно-технического прогресса в течение последних десятилетий сыграло роль предпосылки перехода от постиндустриального к информационному обществу. Ежегодно увеличивается количество людей, задействованных в различных информационных процессах, суть которых состоит в создании, обработке, передаче и хранении информации.

Вышеперечисленные факторы обуславливают популяризацию и повсеместное внедрение информационных дисциплин в образовательные программы средних, средне-специальных и высших учебных заведений, что имеет своей главной целью привить навыки использования компьютерных технологий для решения различного рода задач.

Аналогичная ситуация складывается в сфере компьютерного моделирования, являющегося ведущим направлением при исследовании и проектировании систем разной природы. В сущности, компьютерное моделирование представляет собой создание определенного программного обеспечения с целью имитации поведения реально существующей системы, проведения экспериментов, в ходе которых возможно установить реакцию системы на изменения внешней среды. Результаты такого исследования позволяют установить наличие каких-либо закономерностей, а также оценить имеющиеся стратегии управления системой, направленные на обеспечение ее корректного и стабильного функционирования, параллельно осуществляя информационную поддержку и корректировку стратегий.

Таким образом, изучение методов компьютерного моделирования играет важную роль в рамках изучения компьютерных технологий и робототехники.

Современная робототехника является синтезом механики и кибернетики, а также принципиально новым витком в истории их развития, за счет того, что механика отвечает за разработку технической составляющей роботов

(манипуляторы), а кибернетика, в данном случае, за интеллектуальное управление ими.

Следовательно, главной задачей робототехники на современном этапе выступает активное применение с последующим совершенствованием инструментария механики и кибернетики с целью создания и применения в реальных условиях автоматизированных систем различного назначения. В рамках данной концепции роботы могут осуществлять разного рода деятельность – от их применения в качестве самостоятельной (адаптивной) или программируемой системы, до вспомогательной, когда робот может обслуживать отдельные механизмы, проводить диагностику оборудования или собирать, обрабатывать, передавать и хранить информацию.

Вместе с тем развитие научных теорий предполагает выдвижение основных принципов, построение математической модели объекта познания, получение из нее следствий, которые могут быть сопоставлены с результатами эксперимента. Использование компьютерной техники позволяет, исходя из математических уравнений, рассчитать поведение исследуемой системы в тех или иных условиях.

Актуальность избранной темы исследования состоит в том, что на данный момент робототехника представляет собой стремительно развивающееся научное направление ввиду наличия большого количества перспектив применения роботов для выполнения простых ежедневных или сложных и опасных для человека задач. При этом робот является сложной системой, состоящей из множества разнообразных компонентов взаимодействующих между собой и объединенных с целью воздействовать на окружающую среду в соответствии с поставленной задачей.

Объектом исследования выступают адаптивные роботизированные системы.

Предметом исследования является процесс моделирования сложных роботизированных систем.

Цель исследования состоит в выявление основных признаков сложных роботизированных систем, а также в разработке модели роботизированной адаптивной системы.

Для достижения поставленной цели необходимо решить ряд *исследовательских задач*:

1. Выявить сущность и содержание кибернетического подхода к пониманию сложных систем.
2. Проанализировать основы моделирования кибернетических систем в робототехнике.
3. Выявить основные признаки сложных роботизированных систем.
4. Сформулировать задачу для построения модели адаптивной роботизированной системы.
5. Разработать адаптивный алгоритм решения поставленной задачи.
6. Осуществить программную реализацию адаптивного алгоритма.

ГЛАВА 1. Общетеоретические аспекты моделирования сложных систем

1.1. Кибернетический подход к пониманию сложных систем

Стремительное развитие научно-технического прогресса в конце XX века способствовало появлению электронно-вычислительной техники, персональных компьютеров, смартфонов и всевозможных гаджетов, получивших широкое распространение и внедрившихся во все сферы жизни общества. На современном этапе происходит автоматизация трудовых, исследовательских, производственных процессов посредством внедрения «умных» технологий – программируемых автоматизированных систем. В этой связи, особую актуальность приобретает изучение кибернетики как науки, изучающей взаимодействие компонентов сложной системы между собой и с внешней средой посредством обмена информацией.

На наш взгляд, наиболее целесообразно приступить к исследованию сущности, содержания и основных подходов кибернетики к моделированию сложных систем с анализа основных этапов развития кибернетики, как науки и сопутствующей терминологии.

По замечанию С.И. Ожегова, термин «кибернетика» происходит от греческого «*kybernetike*» и дословно переводится как «искусство управления» [21].

В свою очередь Р.В. Маслов, С.П. Позднева считают, что кибернетика – это: «наука о самоуправляющихся машинах, в частности о машинах с электронным управлением («электронный мозг»))» [18].

Л.И. Лопатников под кибернетикой понимает науку об: «общих принципах управления, понимаемого как организация целенаправленных действий путем переработки информации» [15].

Под кибернетикой также понимают дисциплину, посвященную «изучению систем управления и коммуникации у животных, в организациях и механизмах» [13].

Очевидно, что приведенные выше определения разнятся и не дают четкого однозначного представления о данном термине, что предположительно обусловлено новизной кибернетики, как науки и сферы деятельности, а также определенными историческими предпосылками.

В этой связи представляется целесообразным предоставить краткий исторический очерк отражающий процесс становления и изменения кибернетики, ее целей и сфер применения.

Для решения поставленной задачи, мы проанализировали научные труды авторов: Д.М. Гориневского, А.М., Формальского, А.Ю. Шнейдера («Управление манипуляционными системами на основе информации об усилиях») [7], Л.В. Розановой («Основы кибернетики») [24], В.М. Бондаренко, С.Ю. Румянцевой, В.Е. Дементьева и др. («Кризисы и прогнозы в свете теории длинных волн») [3] и др. и выделили семь основных этапов формирования кибернетики как науки, а также составили сводную таблицу, отражающую их сущность и содержание (см. табл. 1).

Таблица 1

Этапы становления кибернетики

№ п/п	Временной промежуток	Автор	Сущность кибернетики	Примечание
1	428-347 гг. до н.э.	Платон	Под «кибернетикой» понимается «искусство управления»	Данный термин применялся по отношению к социуму и предполагал управление людьми
2	1830 г.	Андре Ампер	Понимал под «кибернетикой» – «искусство управления вообще»	А. Ампер применял термин исключительно к системе классификации человеческих знаний

Продолжение таблицы 1

№ п/п	Временной промежуток	Автор	Сущность кибернетики	Примечание
3	1948 г.	Норберт Винер	Понимал под «кибернетикой» – «управление процессами, как в живом организме, так и в машине»	Данной концепции предшествовало изобретение дифференциальных анализаторов, исследования в нейрофизиологии, исследование динамики живых организмов, создание дискретных преобразователей информации
4	40-е гг. XX вв.	Павлов И.П., Дж. Фон Нейман и др.	Кибернетика представляет собой науку об обмене информацией и ее переработке в различных системах	Данный подход сформировался благодаря возникновению первых электронно- вычислительных машин и соотнесению их функционирования с ЦНС человека
5	50-е гг. XX в.	Лебедев С.А., Глушников В.М. и др.	На данном этапе кибернетика получила более широкое определение и включала в себя множество процессов от автоматизации научных данных до создания сложных систем управления	Кибернетика приобрела характер принципиально новой области знаний об информации, информатике и вычислительной технике ввиду распространения электронно- вычислительных машин
6	60-е гг. XX в.	Институт кибернетики АН СССР	Кибернетика подразумевает разработку теории управления экономическими объектами (предприятия, отрасли промышленности) а также автоматические системы для управления	Разработка ЭВМ «Луч» и «МИР-1» на базе ступенчатого микропрограммирования. Внедрение кибернетики в экономические процессы посредством создания проекта общегосударственной сети вычислительных центров и

			различными техническими средствами	автоматизированных систем управления производством
--	--	--	--	--

Продолжение таблицы 1

№ п/п	Временной промежуток	Автор	Сущность кибернетики	Примечание
7	с 70-ых гг. XX в.	Бир С.	Кибернетика представляет собой совокупность методов исследования операций и научного управления системами с учетом необходимой аналитической информации и целостного описания системы	По мнению С. Бира, целостное описание можно получить исключительно посредством моделирования с помощью кибернетической логики на основании точных наук (математика, физика, информатика)

Из сводной таблицы следует, что кибернетика, как наука и обособленная сфера деятельности получила наиболее широкое распространение в последней трети XX в. и на данный момент нашла применение не только в физико-математическом сегменте научного знания, но и в биологии и социологии.

Обобщив вышеизложенное, мы заключили, что в рамках настоящего исследования под «кибернетикой» следует понимать элемент процесса создания, моделирования, а также управления адаптивными системами (сложными системами).

На современном этапе развития научного знания различают множество подвидов кибернетики. На наш взгляд представляется целесообразным кратко их описать. Для этого мы проанализировали описания, предложенные авторами Л.В. Розановой («Основы кибернетики: конспект лекций» [24]), В.К. Кабулова («Алгоритмическое направление в кибернетике. Некоторые аспекты» [12]) и др., синтезировали их и составили сводную таблицу,

содержащую описание сущностной составляющей каждого из видов кибернетики (см. табл. 2).

Таблица 2

Основные направления (виды) кибернетики

№ п/п	Направление кибернетики	Сфера применения
1	Теоретическая информатика	Математическая дисциплина, применяющая кибернетические методы для построения и изучения моделей, используемых для обработки и передачи информации
2	Экономическая кибернетика	Занимается применением классических кибернетических методов управления к экономическим системам
3	Медицинская кибернетика	Изучает процессы управления и переработки информации в живых организмах, в том числе людях
4	Бионика	Занимается моделированием различных биологических организмов
5	Нейрокибернетика	Занимается изучением «искусственного интеллекта»
6	Социальная кибернетика	Занимается изучением вопросов перехода от автоматизации общества к его кибернетизации и информатизации
7	Техническая кибернетика	Изучает технические системы управления, а также разработкой автоматизированных систем управления и автоматических устройств для хранения, переработки и передачи информации

Отметим, что в рамках настоящего исследования особый интерес представляет собой «техническая кибернетика», в связи с чем представляется целесообразным для упрощения восприятия в дальнейшем под термином «кибернетика» понимать именно «техническую кибернетику».

Ученые Ю.А. Зубарев, А.И. Шамардин отмечают: «современная кибернетика включает в себя ряд разделов, составляющих в совокупности ее теоретическое ядро» [10], графическое отражение которого представлено на рис. 1.



Рис 1. Теоретическое ядро кибернетики по Ю.А. Зубареву, А.И. Шамардину

По замечанию Е.В. Брызгалиной, кибернетика как наука об управлении объектом своего изучения имеет управляющие системы. Для того чтобы в системе могли протекать процессы управления, она должна обладать определенной степенью сложности. С другой стороны, осуществление процессов управления в системе имеет смысл только в том случае, если эта система изменяется, движется, т.е. если речь идет о динамической системе. Поэтому можно уточнить, что объектом изучения кибернетики являются сложные динамические системы [4].

При этом, всю совокупность сложных динамических систем можно отразить в виде трех групп, как показано на рис. 2.

Таким образом, объектом кибернетики являются все управляемые системы; системы, не поддающиеся управлению, в принципе, не являются объектами изучения кибернетики.

Вместе с тем, целью кибернетики является: «достижение желаемого состояния кибернетической системы в управляемом процессе развития системы» [26].



Рис 2. Виды кибернетических сложных систем

При этом, состояние системы, как и траектория ее развития в пространстве состояний, «оценивается с точки зрения их соответствия или несоответствия цели, математически выражением такой оценки является целевая функция, целевой функционал или критерий качества системы, критерий оптимизации» [Там же].

Итак, кибернетика занимается изучением процессов управления, связанных с обменом информацией между системой и окружающей средой, а также выявлением, существующей между информацией и другими характеристиками системы.

Следовательно, информация представляет собой центральное понятие кибернетики. Н. Винер в работе «Человеческое использование человеческих существ: Кибернетика и общество» писал: «в то время как энтропия является мерой дезорганизованности, информация, переносимая некоторым потоком посланий, определяет меру организованности. Фактически мы можем определить информацию как отрицательную энтропию» [9].

Понятию информации кибернетика придает очень широкий смысл, включая в него как всевозможные внешние данные, которые могут восприниматься или передаваться какой-либо определенной системой, так и данные, которые могут вырабатываться внутри системы. В последнем случае система будет служить источником сообщений [27].

Центральное положение понятия информации в кибернетике объясняется тем, что: «кибернетика, ограничивая и уточняя интуитивное представление об информации, изучает машины с точки зрения их способности воспринимать определенную информацию, сохранять ее в «памяти», передавать по «каналам связи» и перерабатывать ее в «сигналы», направляющие их деятельность в соответствующую сторону» [14].

В рамках данной концепции процесс передачи информации можно отразить в виде замкнутой цепи (см. рис. 3).



Рис 3. Замкнутая цепь передачи информации

Кибернетика вводит такие понятия, как «кибернетический подход», «кибернетическая система», «обратная связь», «управление» и т.п., которые играют важную роль в настоящем исследовании, в связи с чем следует определить их сущность и содержание.

Так, кибернетический подход – это «исследование системы на основе принципов кибернетики, в частности с помощью выявления прямых

и обратных связей, изучения процессов управления, рассмотрения элементов системы как неких «черных ящиков» (систем, в которых исследователю доступна лишь их входная и выходная информация, а внутреннее устройство может быть и неизвестно)» [1].

В свою очередь, кибернетическая система – это «система, состоящая из множества взаимосвязанных объектов, способных воспринимать, запоминать и перерабатывать информацию, а также обмениваться ею» [2].

В.М. Пономарева, А.П. Литвинова подчеркивают: «кибернетические системы предназначены для решения задач, существенно более сложных, чем задача автоматического регулирования» [22].

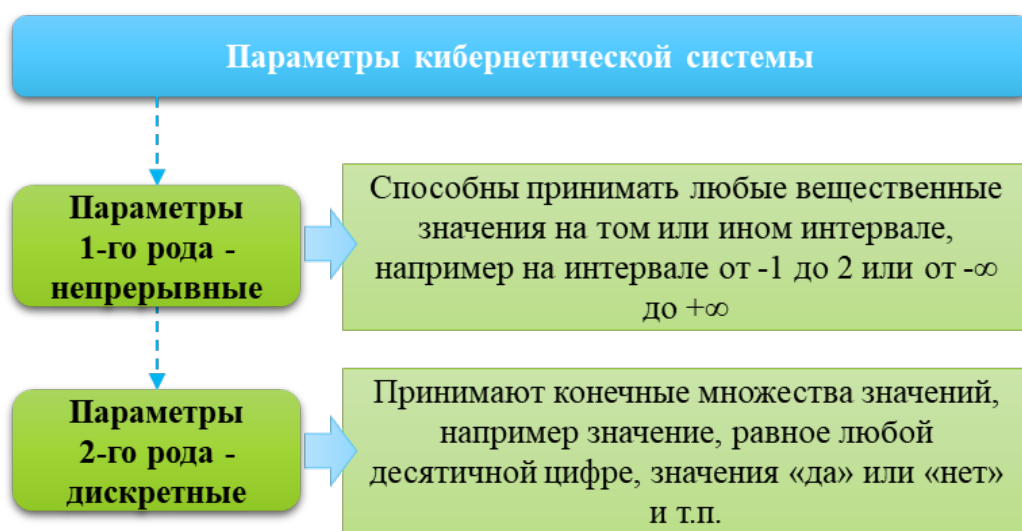


Рис 4. Параметры кибернетической системы

Современная кибернетика, в отличие от классической, занимается изучением сложных систем с большим количеством элементов и связей между ними. При этом в классе сложных систем можно выделить подкласс систем с так называемой «обратной связью». Обратная связь в сложных кибернетических системах рассматривают как: «процесс передачи информации о протекании процесса, на основе которой вырабатывается то или иное управляющее воздействие, в этом случае обратная связь является информационной» [20].

Следовательно, можно сделать вывод о том, что обратная связь представляет собой особую форму взаимодействия между управляемым объектом и внешней средой, она также играет важную роль в анализе функционирования и развития сложных систем управления.

На наш взгляд, графически отразить принцип возникновения и функционирования обратной связи можно в виде схемы на рис. 5.

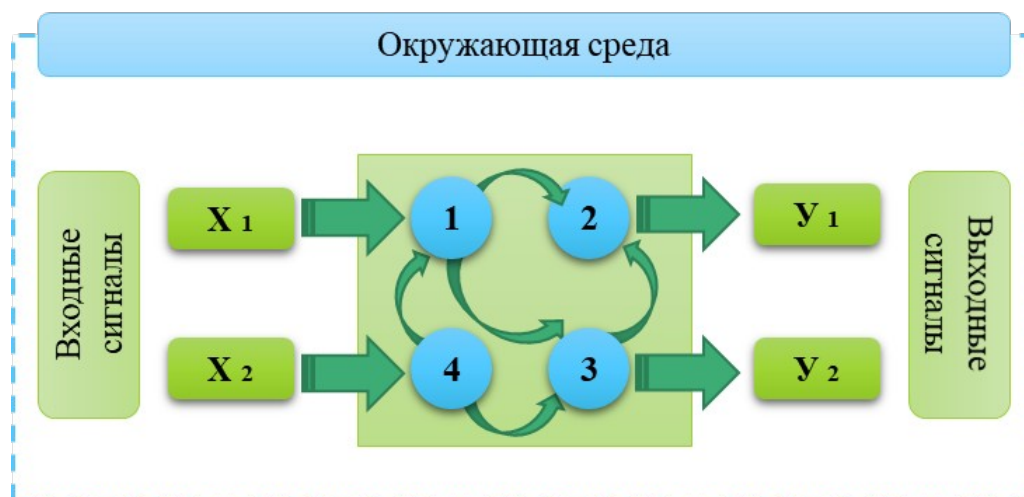


Рис 5. Взаимодействие изучаемой системы и окружающей среды

Схема наглядно демонстрирует, каждый из элементов изучаемой системы выступает в роли отдельной подсистемы, в то время как система, включающая анализируемый объект в качестве подсистемы, является надсистемой. Система имеет входы и выходы, обеспечивающие движение информации от внешней среды в управляемую систему и наоборот.

Очевидно, что рассмотрение управляемой системы в качестве целостного объекта познания (включая внешнюю среду), состоящего из многих взаимосвязанных между собой частей, позволяет увидеть за огромным количеством несущественных деталей и особенностей нечто главное и сформулировать максимально корректные выводы.

По замечанию А.В. Ушакова, необходимой начальной или априорной информацией называется – «совокупность сведений об управляемом процессе и управляемой системе, необходимых для ее функционирования и имеющихся в распоряжении до начала работы системы» [30, 34].

Кибернетика дает новые методы исследования, в частности, на закономерностях, открытых кибернетикой, основан метод моделирования, широко используемый как в естественных, так и в гуманитарных науках [9].

В свою очередь моделирование – это «исследование каких-либо явлений, процессов или систем объектов путем построения и изучения их моделей» [29, 101].

Таким образом, кибернетика – это новое научное направление, охватывающее большое количество проблем и взаимодействующее со многими разделами математики, теории автоматического регулирования, теории информации, математическими машинами, биологией, социальными науками.

Кибернетика, в нашем понимании, это научное направление и сфера деятельности, направленная на построение процессов управления сложными системами, координацию их взаимодействия друг с другом и с внешней средой.

1.2. Основы моделирования кибернетических систем в робототехнике

В данном пункте проанализируем робототехнику с точки зрения кибернетики, так как робот является сложной системой с обратной связью.

В этой связи, следует изучить сопутствующую терминологию, раскрыть понятие «робот», исследовать этапы его создания.

К.Д. Никитин считает, что для определения основного понятия «робот», «следует уяснить главные критерии оценки его возможностей, так как робот по своей концепции возник как устройство, призванное заменить труд человека в самых разнообразных видах и сферах его деятельности. Оценка возможностей робота должна исходить из трех категорий способностей, присущих живому существу, в частности человеку, – физических, функциональных и интеллектуальных» [19, 118].

Профессор Токийского университета, доктор С. Ватаата полагает, что под роботом стоит понимать: «устройство, способное самостоятельно перемещаться в пространстве, справляться с задачами анализа сцен и распознавания образов, обладающее числом степеней подвижности, умеющее анализировать обстановку с помощью обратной связи, а также прогнозировать ситуации, опираясь на собственный опыт и доступную информацию» [8, 278].

По мнению известного французского специалиста Ф. Куафе, робот – это управляемая машина, обладающая двумя отличительными особенностями, сущность которых раскрыта на рис. 6.

В большей степени конкретизировал понятие робота профессор Лондонского университета королевы Марии М. Щпрингер. В своем определении он описал уже шесть различных характерных особенностей, присущих роботам (Приложение 1). Отметим, что М. Щпрингер понимает под роботом не просто сложную систему, облаченную в корпус, а именно антропоморфную машину.



Рис 6. Свойства, отличающие роботов от иных систем по Ф. Куафе

Логично будет привести еще одно определение понятия «робот», полученное К. Д. Никитиным путем статистического анализа ответа респондентов на вопрос: «Что такое робот, чем он отличается от машин и автоматических систем?», заданный 156 экспертам, специализирующимся в различных областях науки и техники.

Результатом проведенного опроса стала следующая формулировка: «робот представляет собой подвижную компактную систему, отличительными признаками которой являются чувствительные элементы, манипуляторы и, самое главное, некоторая степень искусственного интеллекта, при этом искусственный интеллект отождествляется главным образом со способностью к обучению и соответственно к изменению поведения» [19, 164].

На основании изученных определений, мы предприняли попытку выделить характерные черты роботов, заложенные в них. Так, на наш взгляд, к числу характерных особенностей, отличающих роботов от остальных программируемых машин следует отнести:

1. *Автономность*, под которой понимается способность самостоятельного выполнения действий или производственных операций, сообразуясь лишь с программным алгоритмом либо с целеуказательной командой и изменяющимися условиями внешней среды.

2. *Универсальность*, понимаемая как способность выполнять самые различные действия или производственные операции и легко переходить с одного вида действий на другой.
3. *Автоматичность*, т.е. способность выполнять достаточно сложные и завершённые действия или производственные циклы без непосредственного вмешательства человека-оператора.
4. *Антропоморфизм*, понимаемый в широком смысле как наделение робота способностями, присущими человеку: физическими (силовыми), функциональными (двигательными) и интеллектуальными (умственными). В узком смысле под антропоморфизмом понимается внешнее сходство робота с человеком, что совершенно необязательно и может использоваться лишь в специальных целях.
5. *Адаптивность*, т.е. способность к целенаправленному изменению своего поведения под влиянием изменений внешних условий и к обучению в процессе взаимодействия с внешней средой (гибкость). Способность к адаптации и обучению реализуется путем наделения робота теми или иными средствами обратной связи: осязанием, зрением, слухом, обонянием, запоминанием и т.п.

Таким образом, робот представляет собой автономно функционирующую универсальную автоматическую машину, предназначенную для воспроизведения физических, двигательных и умственных функций человека, а также обладает способностью к адаптации и обучению в процессе активного взаимодействия с окружающей средой.

По замечанию Н. В. Василенко, К. Д. Никитина В. П. Пономарева, А. Ю. Смолина, робот – это: «хороший пример того, как сумма ранее известных составных частей (манипуляторов, ЭВМ, сенсорики) дает новое качество – принципиально новый тип технического устройства, обладающего в достаточно развитом варианте искусственным интеллектом, искусственными органами чувств (сенсорика), способностью воспринимать

окружающую среду и активно воздействовать на нее, обучаясь и совершенствуясь в ходе этого процесс» [5, 91-93].

Однако стоит иметь ввиду, что на данный момент не существует роботов, полностью соответствующих вышеприведенному определению, что связано со сравнительной новизной робототехники, как науки и дороговизной создания и обслуживания подобных роботов. Широко применяемые роботы не обладают устройствами типа органов чувств человека и способностью мыслить, а подобные разработки пока не вышли из стен научных лабораторий и не получили широкого применения. Но, несмотря на это, наука и техника делают уверенные шаги по пути решения одной из фундаментальных проблем робототехники – создания чувствующих и мыслящих роботов.

Ведущие отечественные и зарубежные ученые [6, 8], [16, 16], [23, 76] описывают различные этапы создания роботов; обобщив мнение авторов, мы составили схему, отражающую этапы процесса создания робота (Приложение 2).

Наиболее актуальным в рамках настоящего исследования является анализ процесса разработки прототипов – моделирование, а именно компьютерное моделирование.

По мнению Р.В. Майера, сущность компьютерного моделирования системы заключается в: «создании компьютерной программы (пакета программ), описывающей поведение элементов исследуемой системы в процессе ее функционирования, учитывающей их взаимодействие между собой и внешней средой, и проведении на ЭВМ серии вычислительных экспериментов» [17, 152].

Целью компьютерного моделирование в этом случае выступает изучение природы и поведения объекта, его оптимизации и структурного развития, прогнозирования новых явлений.

Советов Б. Я., Яковлев С. А. [28, 85-86] приводят перечень требований, которым должна удовлетворять компьютерная модель исследуемой системы,

основываясь на авторитетном мнении авторов, мы составили краткую схему, отражающую сущность этих требований (рис. 7).



Рис 7. Критерии эффективно компьютерной модели сложной системы

Так, полнота модели предполагает наличие возможности проведения вычисления всех характеристик системы с требуемой точностью и достоверностью.

Гибкость модели подразумевает, что модель позволяет воспроизводить и проигрывать различные ситуации и процессы, изменять структуру, алгоритмы и параметры изучаемой системы.

Длительность разработки и реализации, характеризующая временные затраты на создание модели, в то время как блочность структуры, допускающая добавление, исключение и замену некоторых частей (блоков) модели. Кроме того, информационное обеспечение, программные и технические средства должны позволять модели обмениваться информацией с соответствующей базой данных и обеспечивать эффективную машинную реализацию и удобную работу пользователя.

Следует отметить, что процесс компьютерного моделирования является технически сложным и включает в себя ряд циклических этапов (рис. 8).

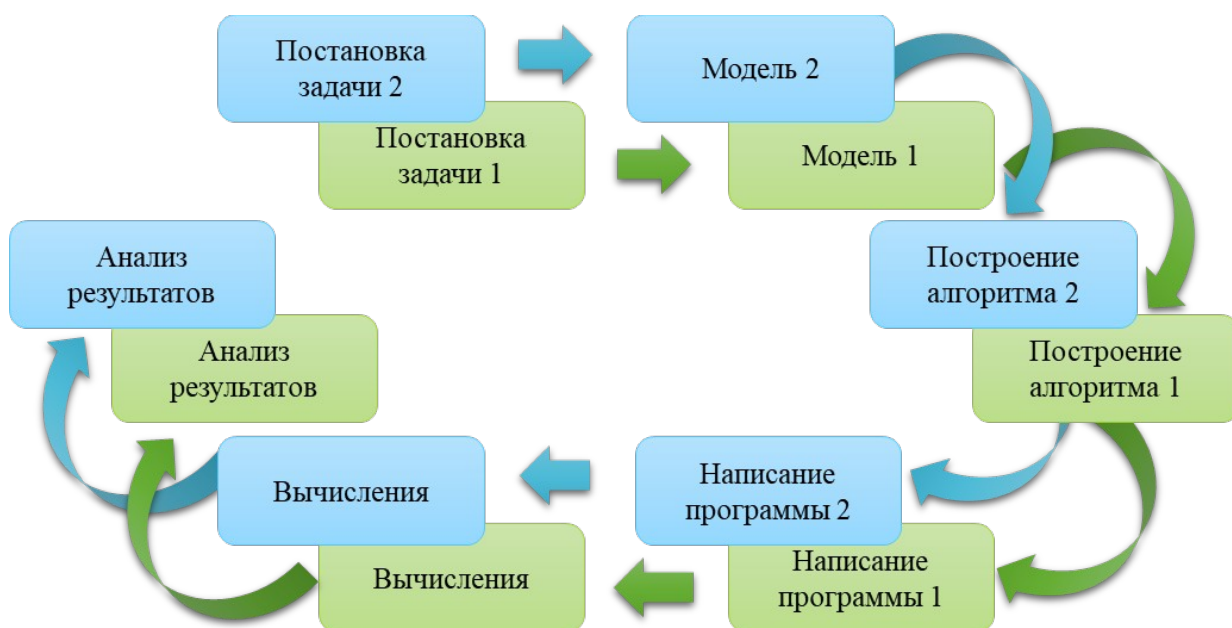


Рис 8. Основные этапы компьютерного моделирования сложной системы

Рассмотрим данный процесс более подробно. Так, постановка задачи предполагает описание исследуемой системы и выявление ее компонентов и элементарных актов взаимодействия.

Создание модели, представляющей собой систему, отражающую сущность исследуемого объекта.

Разработка алгоритма, направлена на реализацию поставленной задачи, а написание программы предусматривает использование конкретного языка программирования.

На определенном этапе происходит планирование и выполнение вычислений на ЭВМ, доработка программы и получение результатов с последующим плавным переходом к анализу и интерпретации результатов, их сопоставлению с эмпирическими данными, что может быть в дальнейшем проведено повторно в виде нового цикла.

Н. В. Сафронова отмечает: «робототехника – прикладная наука, занимающаяся разработкой автоматизированных технических систем. Робототехника опирается на такие дисциплины как электроника, механика, программирование, искусственный интеллект» [25].

Робот, активно взаимодействующий с окружающей средой, в общем виде должен содержать следующие подсистемы: управляющую, информационно-

измерительную (сенсорную), подсистему связи, исполнительную (моторную) [11, 34].

Согласно мнению авторов, управляющая, или интеллектуальная система – это «мозг» робота, который служит для выработки законов управления механизмами исполнительной системы на основе заложенной программы с учетом сигналов обратной связи от сенсорной системы.

На наш взгляд, данную взаимосвязь можно кратко представить в виде схемы на рис. 9, объединив моторную систему, систему связи и сенсорную систему в два блока: информационную и двигательную системы.

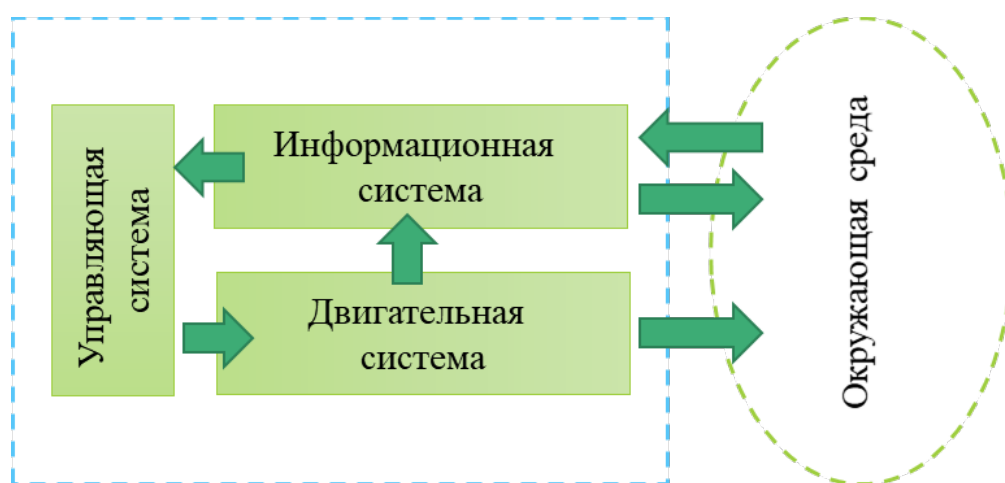


Рис 9. Взаимодействие робота и внешней среды

Важной функцией информационной системы, на наш взгляд, является распознавание ситуаций и моделирование среды функционирования робота, планирование действий и принятие целенаправленных решений, программирование и оптимизация движений, а также организация общения робота с человеком и взаимодействующими устройствами.

Таким образом, схему взаимодействия человека и робота можно графически отразить в виде взаимосвязи на рис. 10.

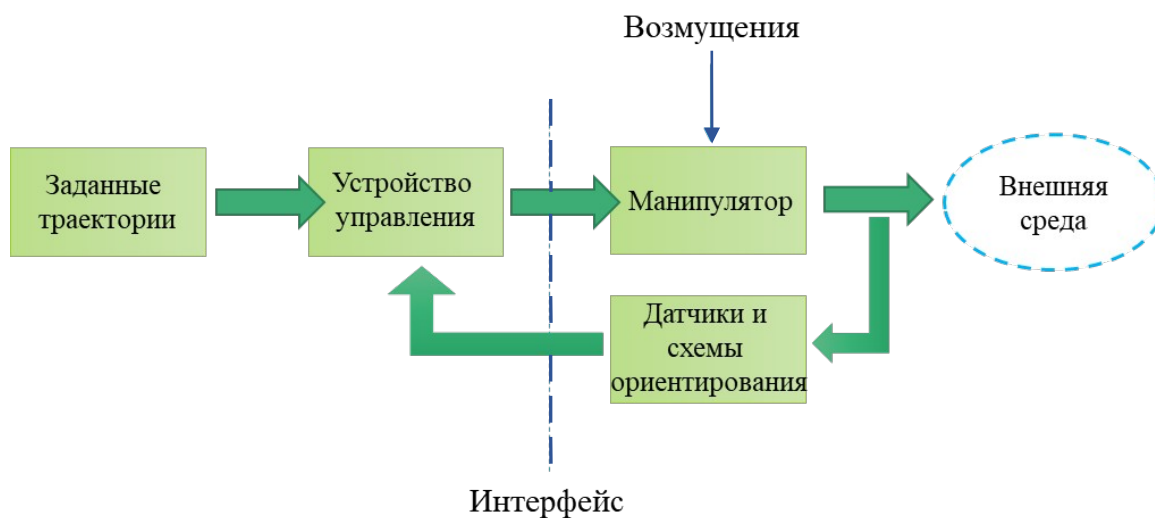


Рис. 10. Взаимосвязь человека, робота и окружающей среды

Управляющие системы роботов реализуются на базе пневматических или электрических логических элементов, аналоговых и цифровых систем, а в наиболее совершенном варианте – на основе ЭВМ или микропроцессоров, содержащих широкий набор входных (аналого-цифровых) и выходных (цифро-аналоговых) преобразователей и интерфейсных каналов связи, число которых может колебаться от нескольких десятков до нескольких тысяч. По этим каналам, как по нервным волокнам, передаются непрерывные (аналоговые) и дискретные (цифровые) сигналы. Интеллектуальные и адаптивные возможности робота определяются главным образом алгоритмическим и программным обеспечением управляющей системы.

Информационно-измерительная, или сенсорная система – это искусственные органы чувств робота, предназначенные для восприятия и преобразования информации о состоянии внешней среды и самого робота. В качестве элементов сенсорной системы используются телевизионные и оптико-электронные устройства, инфракрасные и ультразвуковые дальномеры, акустические датчики и гидролокаторы, тактильные, контактные и индукционные датчики, а также датчики положения, скорости, сил и моментов, потенциометры, тахометры, акселерометры и т. п.

Система связи – это «язык» робота, который служит для передачи сигналов информации между системами робота, а также для организации

обмена информацией между роботом и человеком или другими роботами и устройствами с целью осуществления диалога, формулирования заданий роботу, контроля над функционированием его систем, диагностики неисправностей, регламентной проверки и т.п.

Информация от человека поступает обычно через устройство ввода или пульт управления, путем программного или физического воздействия (нажатие кнопки или клавиши, ключа телеграфного аппарата и т.п.). В последнее время все шире начинает применяться речевое общение, а также ввод информации с помощью биопотенциалов (биоуправление).

Информация от робота к человеку поступает, как правило, в форме световых и звуковых сигналов, а носителями этой информации являются разного рода табло, цифровые индикаторы, дисплеи, телекамеры и т.п. Результаты лабораторных исследований дают основания полагать, что в ближайшем будущем станет возможным речевое общение с роботом на естественном человеческом языке.

Совокупность управляющей, информационно-измерительной и системы связи образует информационно-управляющую систему робота, обеспечивающую обработку и передачу информации, и непосредственное управление приводами и механизмами исполнительной системы с целью организации активного взаимодействия робота с окружающей средой и выполнения задач, сформулированных человеком.

Исполнительная, или моторная система – это устройства, предназначенные для непосредственного воздействия на объекты окружающей среды или взаимодействия с ними в соответствии с управляющими сигналами, формулируемыми информационно-измерительной системой или непосредственно оператором. В качестве элементов моторной системы используются приводы (двигатели), передаточные устройства (передачи), связанные с ними механические руки (манипуляторы), механические ноги (педикюляторы), различные

технологические инструменты, графопостроители, тележки с колесным, гусеничным и иными шасси и др.

Итак, исследовав основы моделирования сложных систем в робототехнике, мы заключили, что данный процесс является цикличным, что направлено на выявление и своевременное устранение ошибок, а также сложными многоуровневым и представляет собой один из важнейших этапов проектирования робота в целом.

ГЛАВА 2. Разработка роботизированной адаптивной системы

2.1. Постановка задачи

Робототехнические системы, которые способны выполнить поставленную перед ними задачу в изменяющихся условиях внешней среды, называются адаптивными. Это означает, что в процессе выполнения задания могут возникнуть ситуации, которые нельзя описать на этапе формирования задания, но можно предусмотреть ориентировочно. Определяющей характеристикой адаптивных систем является их способность получать информацию о внешней среде и использовать ее.

В рамках данного проекта внешней средой для робота является плоское поле размером 113 на 113 см с однородной по фактуре поверхностью серого цвета (рис. 10). По периметру поле огорожено бортами высотой 30 см. На поле случайным образом расположены препятствия и цель, которая представляет собой белый круг диаметром 9 см. Место старта обозначено темным прямоугольником.



Рис 10. Поле

Задача формулируется следующим образом: робот, способный самостоятельно передвигаться, должен объезжая препятствия найти цель, а затем осуществлять движение от цели до места старта и обратно.

Решение этой задачи может быть представлено в качестве проекта по образовательной робототехнике, в рамках которого будет показан процесс конструирования и программирования машины отвечающей всем признакам роботизированной системы способной самостоятельно взаимодействовать с внешней средой и подстраиваться под нее, т.е. являющейся адаптивной.

Создание самообучающейся системы не требует наличия каких-либо специальных технических средств, поскольку адаптивные возможности робота в большей степени обусловлены особенностями алгоритма его функционирования.

В качестве технической базы в данной работе выступает набор для сборки мобильного робота POP-BOT тайландской фирмы Innovative Experiment (рис. 11). Этот набор при относительно низкой стоимости включает в себя все необходимые компоненты для проведения практических занятий по робототехнике.



1. Управляющая система:

- ## 2. Сенсорная система:

- ### 3. Исполнительная система:

4. Связь между компонентами робота осуществляется подачей электрических сигналов на определенные порты платы управления.

Техническая база накладывает ограничения на функциональные возможности робота. Поэтому перед сборкой робототехнической системы рассмотрим особенности входящих в нее компонентов.

Модуль Arduino POP-168 является платой с гибкими функциональными возможностями. В нем используется микроконтроллер ATmega 168 компании Atmel с 10-разрядным аналого-цифровой преобразователем (АЦП) и тактовой частотой в 16 МГц. Модуль имеет 1 кБайт памяти ОЗУ и 16 кБайт флэш-памяти. Аппаратная часть Arduino POP-168 полностью совместима с Arduino-mini, что позволяет использовать среду Arduino IDE для разработки программного обеспечения робота.

Плата RBX-168, на которую установлен модуль Arduino POP-168, осуществляет управление роботом (рис. 12). На ней расположены порты для подключения внешних устройств, переключатель питания, кнопки сброса и выбора режима программирования, а также светодиодные индикаторы состояния устройств.

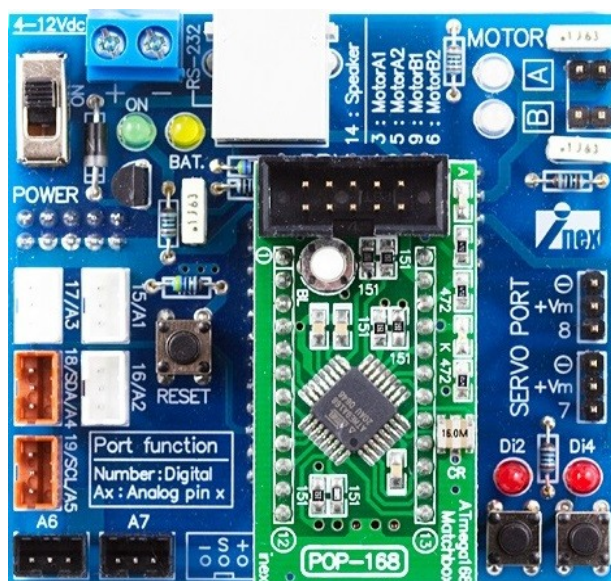


Рис 12. Плата управления роботом

Также плата управления соединена с аккумуляторным блоком для четырех батарей типоразмера AA.

Инфракрасный отражательный датчик ZX-03 может определять оттенки серого в непосредственной близости (рис. 13). Диапазон допустимого расстояния от датчика до объекта лежит в пределах от 3 до 8 мм.



Рис 13. Инфракрасные отражательные датчики ZX-03

Датчик GP2D120 выполняет функцию инфракрасного измерителя расстояния (рис. 14). Диапазон измеряемых расстояний составляет от 4 до 30 см.



Рис 14. Инфракрасный датчик расстояния GP2D120

Набор для сборки мобильного робота POP-BOT укомплектован двумя электродвигателями постоянного тока (рис. 15). Средняя частота вращения двигателей составляет 210 оборотов в минуту при отсутствии нагрузки. Минимальный крутящий момент равен 0.5 кг*см.

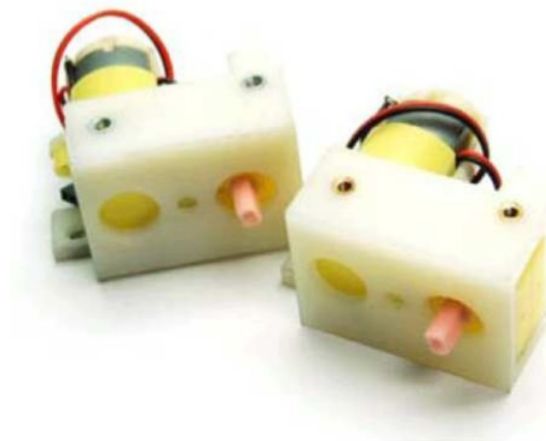


Рис 15. Электродвигатели постоянного тока

Стандартный сервомотор позволяет осуществлять поворот на заданный угол в диапазоне от 0 до 180 градусов (рис. 16). Максимальное угловое перемещение происходит за 1.5 секунды. Крутящий момент равен 3.40 кг*см.



Рис 16. Сервомотор

В набор входит пара колес предназначенных для использования совместно с модулями электродвигателей постоянного тока.

Для обмена данными между USB-портом компьютера и платой управления RBX-168 используется кабель UCON-4.

Описанные выше компоненты, объединенные в робототехническое устройство, связаны в соответствии со схемой, приведенной на рис. 17. Все взаимодействия между устройствами проходят через плату управления. Датчик расстояния закреплен на вале сервомотора, что позволяет

манипулировать им отдельно. Внешний вид собранного робота представлен на рис. 18.

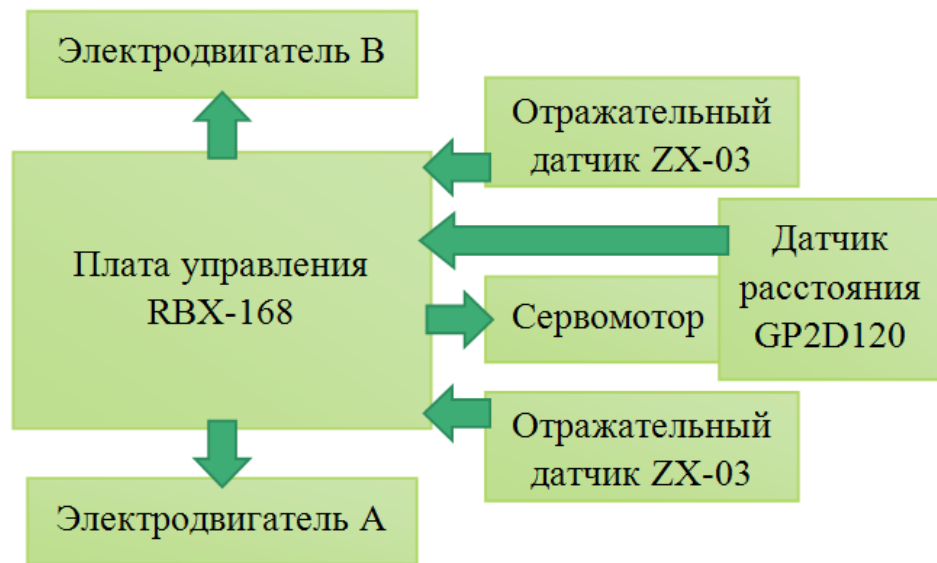


Рис 17. Схема взаимодействия компонентов робота

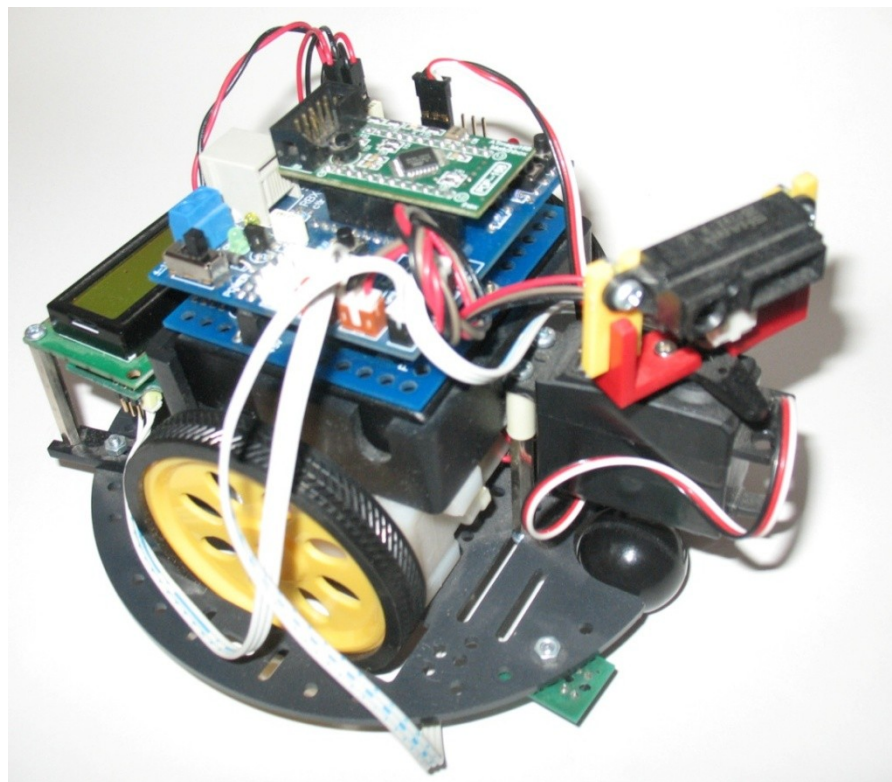


Рис 18. Внешний вид робота

Сконструированная робототехническая система способна изменять свое положение в окружающей среде, оказывая тем самым воздействие на нее.

Наличие датчиков дает системе потенциальную возможность взаимодействовать с внешней средой, подстраиваясь под ее состояние. В то же время, разнообразие получаемых данных сильно ограничено и сводится к двум типам:

- расстояние до предметов в поле зрения дальномера с учетом угла поворота сервомотора;
- яркость поверхности, по которой перемещается робот.

Для решения поставленной перед роботом задачи необходимо разработать алгоритм, с помощью которого он в заданных условиях сможет самостоятельно построить траекторию движения от места старта до искомой цели. В способности самостоятельного нахождения и запоминания траектории и состоит адаптивность данной робототехнической системы.

2.2. Разработка адаптивного алгоритма

Поскольку в рамках данного проекта задача, поставленная перед роботом, сводится к построению траектории, необходимо определить способ ее задания и хранения. В простейшем случае траектория движения представляет собой комбинацию прямолинейных участков различной длины и определенного количества поворотов на заданный угол. Таким образом, параметрами траектории является:

- количество прямолинейных отрезков;
- длины прямолинейных отрезков;
- количество поворотов между прямолинейными отрезками;
- направление поворотов.

Первоначально значения данных параметров не определено. Поэтому первым этапом алгоритма должен быть процесс обучения, который заключается в нахождении неизвестных параметров траектории опытным путем. Окончание процесса обучения совпадает с нахождением цели

Вторым этапом алгоритма должно быть движение назад от цели к месту старта по траектории, найденной на первом этапе.

Третий этап заключается в движении вперед от места старта к цели и является подтверждением работоспособности данного адаптивного алгоритма. При этом третий этап должен занимать меньше времени, чем первый, что демонстрирует эффективность запоминания траектории движения по сравнению с постоянным ее поиском. После завершения третьего этапа снова повторяется второй.

Общая схема этапов адаптивного алгоритма и переходов между ними представлена на рис. 19.

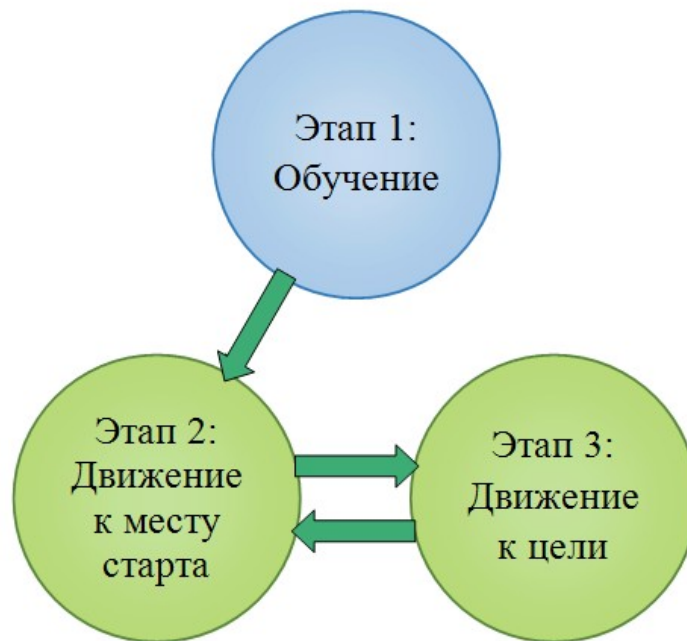


Рис 19. Общая схема этапов адаптивного алгоритма

Траектория, состоящая из конечного числа отрезков, может быть представлена в виде набора однотипных элементов. В компьютерном представлении такая траектория может иметь вид массива чисел. Для данного алгоритма выбрано представление траектории движения в виде двух массивов:

- TF – массив, хранящий длину прямолинейных отрезков;
- LR – массив, хранящий количества и направление поворотов между прямолинейными отрезками.

Для удобства работы с массивами, номер последнего элемента хранится в переменной N. При старте робота оба массива пусты. Они заполняются в процессе обучения и на момент перехода ко второму этапу, траектория движения от места старт к цели полностью определяется этими массивами.

Для идентификации этапа используется переменная ST. Таким образом, общая схема этапов адаптивного алгоритма примет вид блок-схемы представленной на рис. 20.

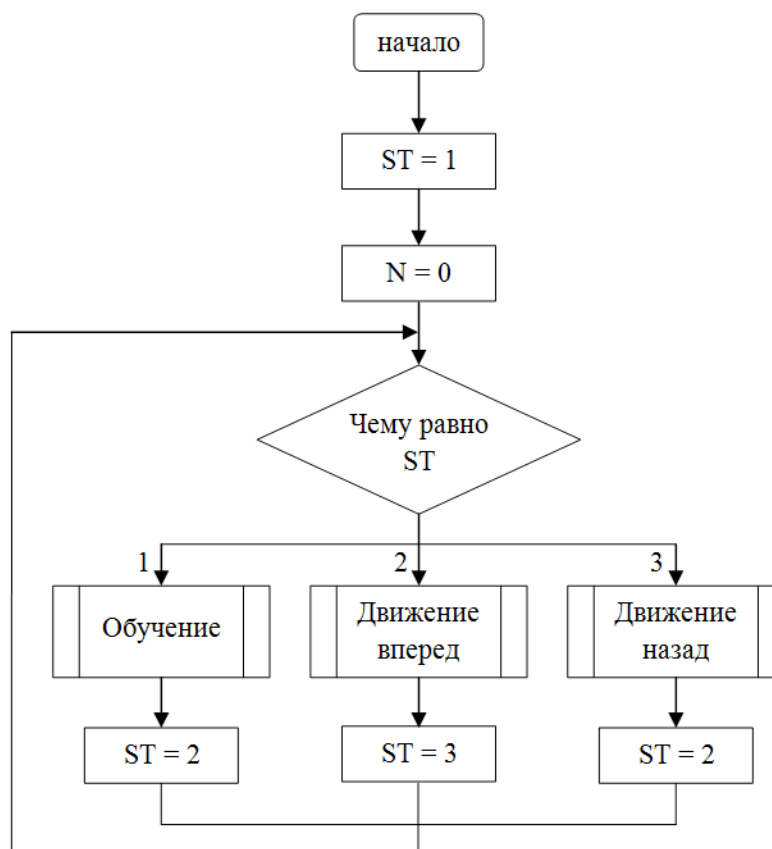


Рис 20. Общая блок-схема этапов адаптивного алгоритма

Бесконечный цикл обусловлен особенностями языка программирования Arduino, который используется для реализации этого алгоритма.

Далее подробно рассмотрим каждый этап алгоритма.

Процесс обучения удобно разделить на две части. Сначала определяется длительность прямолинейного отрезка. Поскольку по прямой робот движется с постоянной скоростью, удобно выражать длину таких отрезков через время, которое тратится на их преодоление. Затем, при достижении препятствия, роботу необходимо повернуть. Поворот осуществляется на небольшой фиксированный угол, поэтому достаточно запоминать количество необходимых поворотов, учитывая также их направление.

Следует также учесть, что прямолинейный отрезок может закончиться не только достижением препятствия, но также нахождением цели. В таком случае необходимо перейти от этапа обучения к следующему этапу. Алгоритм процесса обучения приведен на рис. 21.

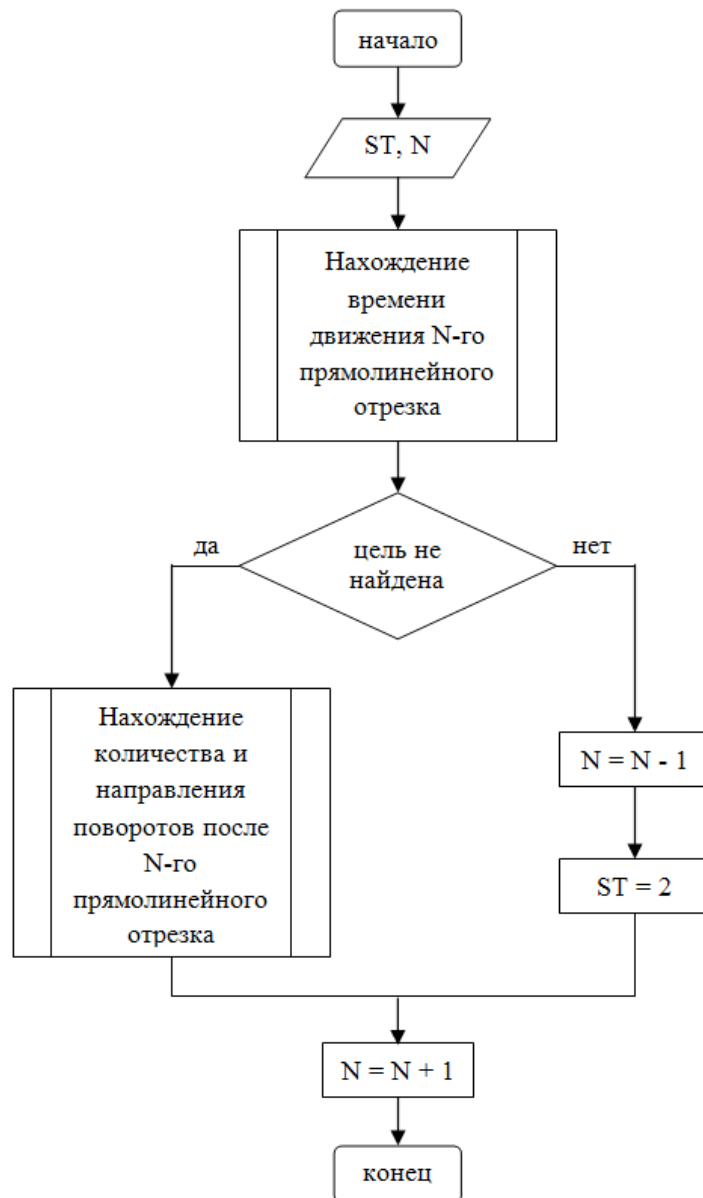


Рис 21. Алгоритм процесса обучения

Поскольку данный процесс находится внутри цикла, приведенного на рис. 20, на каждом шаге происходит заблаговременное увеличение номера последнего элемента в массиве отрезков траектории. Поэтому, при переходе к следующему этапу в момент нахождения цели необходимо компенсировать лишние в данном случае изменение N .

Рассмотрим подробнее процедуры, применяющиеся на этапе обучения.

Алгоритм процедуры нахождения времени движения на N -ом прямолинейном отрезке траектории представлен на рис. 22.

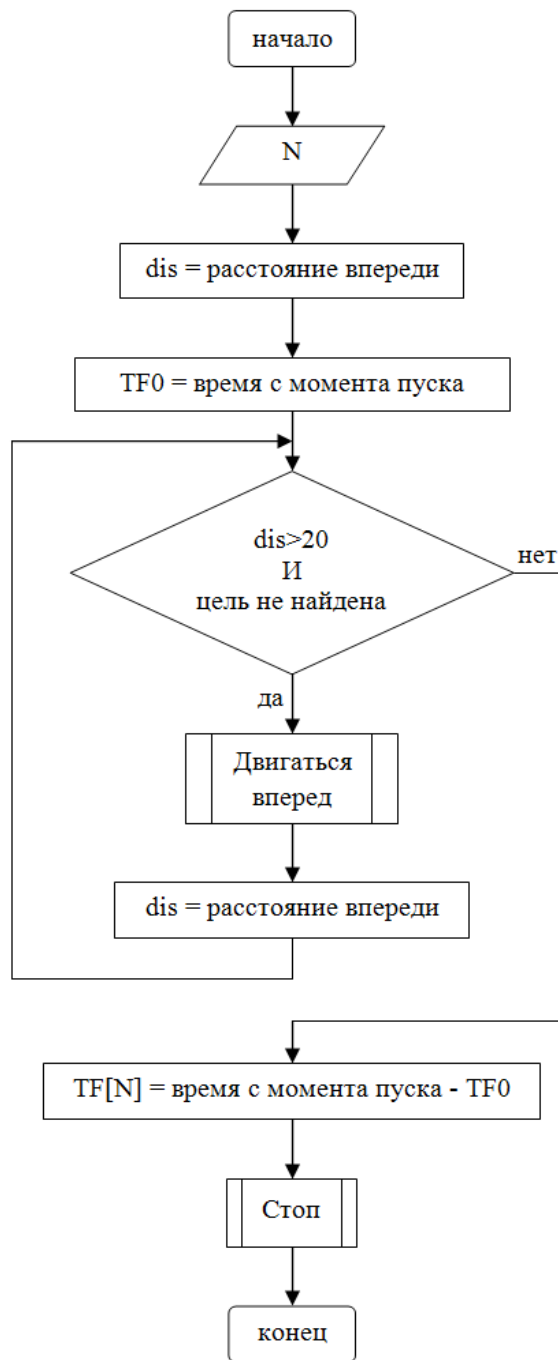


Рис 22. Процедура нахождения времени движения на прямолинейном отрезке

Расстояние до препятствия определяется с помощью датчика расстояния, сохраняющееся в локальную переменную *dis*. Оно должно быть не менее 20 см, чтобы робот имел возможность свободно осуществлять повороты. Время с момента пуска вычисляется с помощью встроенного счетчика. Достижение цели, представляющей собой белый круг на серой поверхности, определяется по отражательным датчикам, прикрепленным на небольшом расстоянии от поверхности.

При достижении препятствия, в алгоритме процесса обучения следует процедура нахождения количества и направления поворотов, необходимых после прохождения N-го прямолинейного отрезка траектории. Алгоритм этой процедуры представлен на рис. 23.

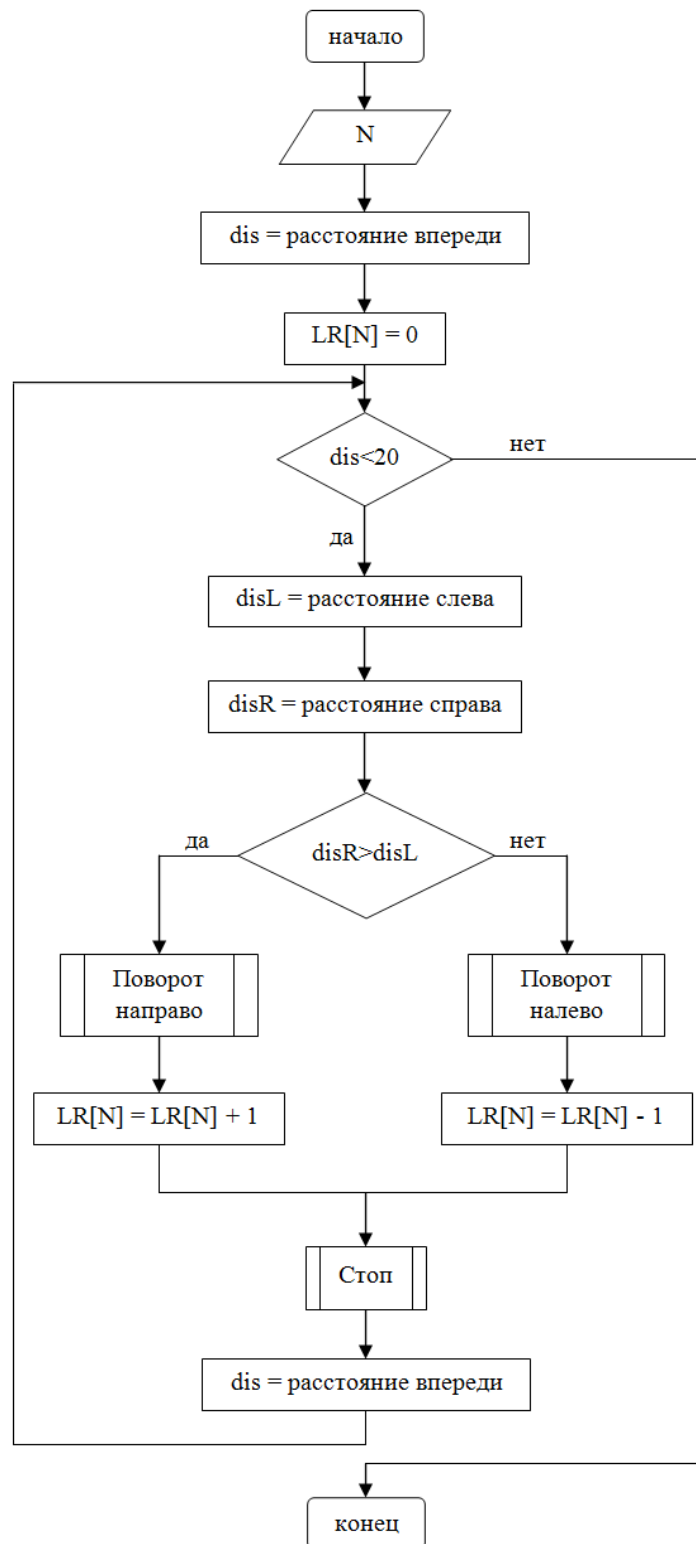


Рис 23. Процедура нахождения количества и направления поворотов

В начале процедуры робот стоит, а все манипуляции осуществляются сервомотором, на котором расположен датчик расстояния. Это позволяет определить расстояния до препятствий справа и слева и осуществить поворот в ту сторону, где расстояние до следующего препятствия наибольшее. В зависимости от направления поворота, число характеризующие количество поворотов имеет отрицательное (при повороте налево) или положительное (при повороте направо) значение. По окончании данной процедуры робот приступает к нахождению следующего прямолинейного отрезка в новом витке процесса обучения.

Этап обучения заканчивается, когда робот первый раз достигает цели. Затем следует этап движения назад. Робот разворачивается и движется к месту старта по найденной траектории. К этому моменту массивы TF и LR заполнены всеми необходимыми данными, и информация из них извлекается в цикле с фиксированным числом шагов. Извлечение данных при этом идет в обратном порядке. В конце второго этапа робот останавливается и меняет идентификатор этапа ST. Блок-схема второго этапа приведена на рис. 24.

На третьем этапе робот вновь осуществляет движение вперед от места старта до цели. По своему алгоритму третий этап аналогичен второму с некоторыми отличиями. Главное отличие состоит в направлении извлечения данных из массивов: от первого к последнему элементу или от последнего к первому. Соответственно, при движении робота от места старта к цели данные извлекаются в направлении от первого к последнему, а при движении от цели к месту старта – от последнего к первому.

В зависимости от направления поворота в переменную j из массива LR извлекаются либо положительное, либо отрицательное значение, которое всегда отлично от нуля. Также необходимо учесть, что при смене этапов алгоритма зеркально изменяются направления необходимых поворотов. Поворот налево при движении от места старта к цели соответствует повороту направо при движении обратно.

Блок-схема третьего этапа приведена на рис. 25.

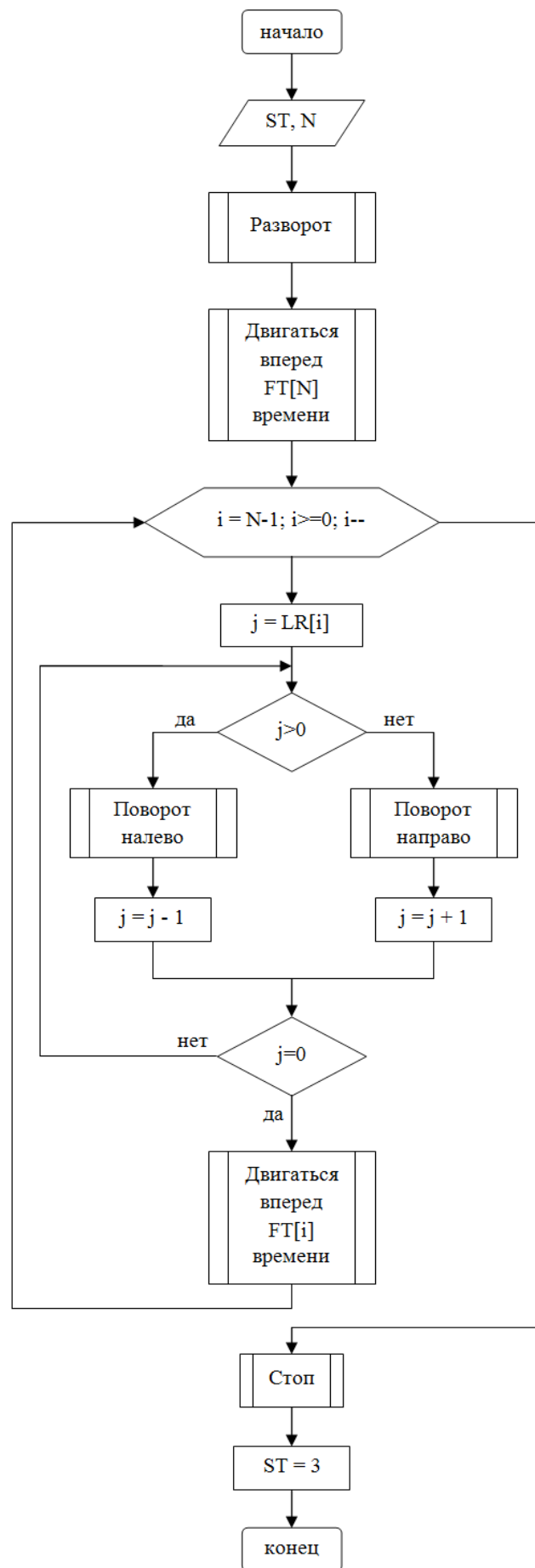


Рис 24. Блок-схема второго этапа

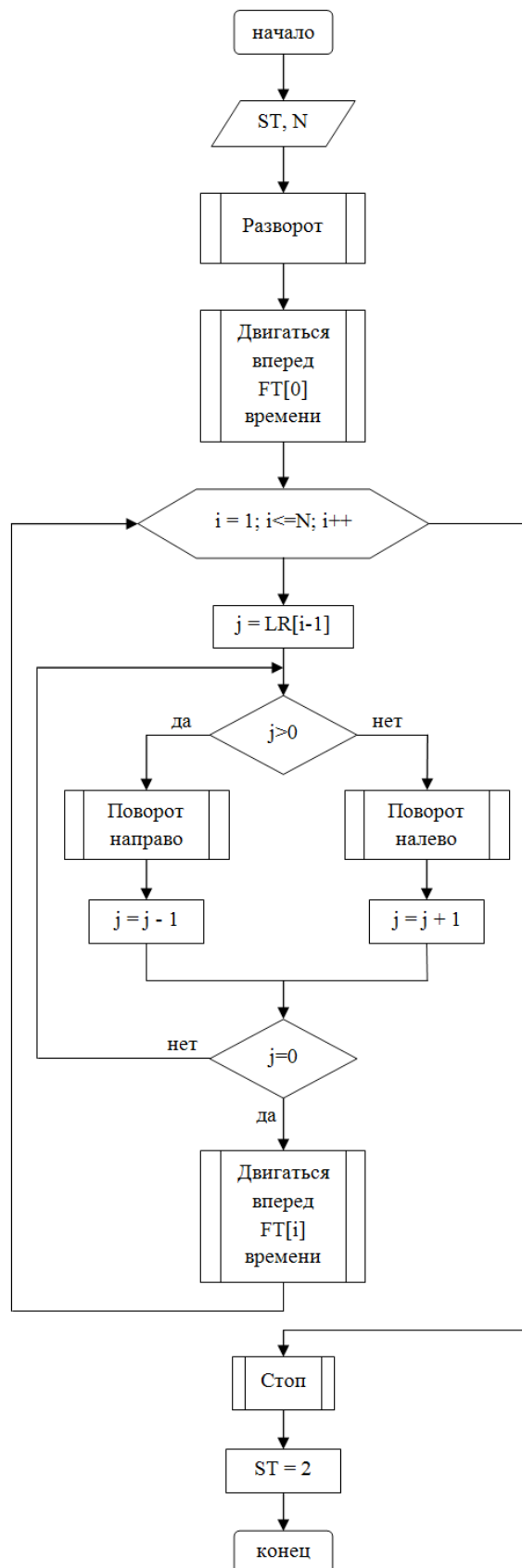


Рис 25. Блок-схема третьего этапа

Таким образом, адаптивный алгоритм нахождения траектории движения к определенной цели разработан и готов к программной реализации.

2.3. Программная реализация адаптивного алгоритма

Для программной реализации разработанного алгоритма используется среда разработки Arduino IDE (рис. 26). Язык программирования Arduino является стандартным C++ с некоторыми особенностями. Простейшая Arduino-программа состоит из двух процедур:

- `setup` – вызывается однократно при старте микроконтроллера;
- `loop` – вызывается после `setup` в бесконечном цикле все время работы микроконтроллера.

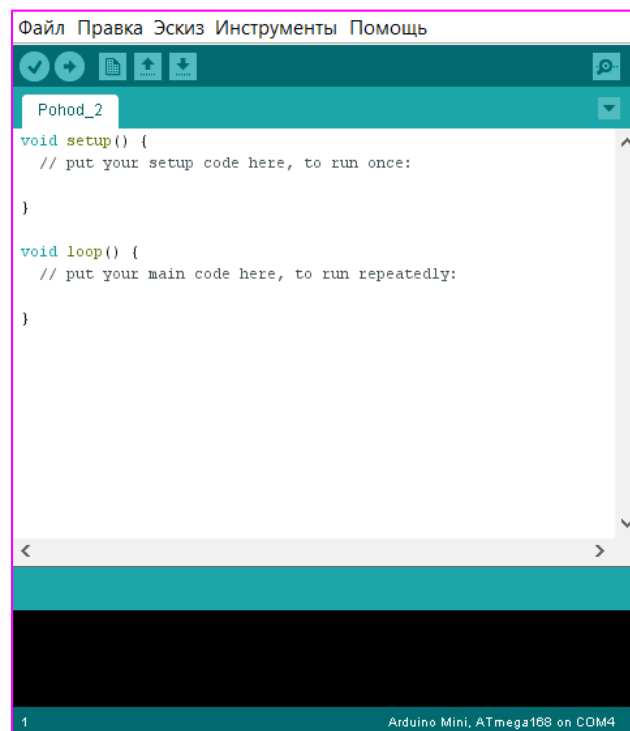


Рис 26. Главное окно приложения Arduino IDE

В процедуре `setup` осуществляется назначение портов платы управления и присвоение первичных значений переменным `ST` и `N`. Код процедуры `setup` представлен ниже:

```
void setup() { //Установки
  pinMode(3,OUTPUT); //Электродвигатель A1
  pinMode(5,OUTPUT); //Электродвигатель A2
  pinMode(6,OUTPUT); //Электродвигатель B1
```

```

pinMode(9,OUTPUT); //Электродвигатель B2
pinMode(8,OUTPUT); //Сервомотор
ST=1; //Идентификатор этапа
N=0; //Номер последнего элемента массива TF
}

```

Процедура `pinMode` является стандартной и конфигурирует режим работы указанного порта либо как вход, либо как выход. Например, `pinMode(3,OUTPUT)` устанавливает режим работы третьего порта как выходного или принимающего сигналы. Это позволяет микроконтроллеру управлять электродвигателями А и В и сервомотором, которые подключены к соответствующим портам. Все остальные порты по умолчанию будут работать в режиме INPUT как входные.

Вторая часть программы – бесконечный цикл, представленный на рис. 20, преобразуется в программный код внутри процедуры `loop`:

```

void loop() { //Основной цикл
    switch(ST){
        case 1: //Этап 1
            Learn(); //Обучение
            break;
        case 2: //Этап 2
            To_Start(); //Движение от цели к месту старта
            break;
        case 3: //Этап 3
            To_Target(); //Движение от места старта к цели
            break;
    }
}

```

Программный код полностью соответствует разработанному алгоритму. Достижение цели определяется с помощью функции `Target`, которая для

удобства возвращает значение «ложь», если цель достигнута. Код функции Target представлен ниже:

```
boolean Target(){ //Определение достижения цели
    int LS, RS;
    LS=analogRead(7); //Чтение значения с левого отражательного датчика
    RS=analogRead(6); //Чтение значения с правого отражательного датчика
    return (LS<500)&&(RS<500); //Определения достижения белой области
}
```

В данной функции происходит считывание аналогового сигнала с отражательных датчиков, подключенных к седьмому и восьмому портам. Поскольку поверхность поля однородного серого цвета, значение которое сообщают датчики меньше 500. Значение, посылаемое датчиком над белой областью цели, будет близко к 650. Это позволяет роботу соответствующе реагировать, когда искомая цель оказывается под одним из его отражательных датчиков.

Управление передвижением робота осуществляется отправлением сигналов в порты, к которым подключены электродвигатели постоянного тока. Электродвигатели управляются сигналами с широтно-импульсной модуляцией (ШИМ). Меняя ширину импульса, подаваемого на электродвигатель постоянного тока, можно менять количество энергии, поступающее на электродвигатель в единицу времени, изменяя таким образом частоту вращения (скорость) электродвигателя. При большей ширине импульса, электродвигатель будет вращаться с большей частотой, в то время как амплитуда подаваемого напряжения остается постоянной. Так, процедура Forward, обеспечивающая прямолинейное движение робота вперед, имеет вид:

```
void Forward() { //Двигаться вперед
    analogWrite(3,200); //Электродвигатель A1
    digitalWrite(5,LOW); //Электродвигатель A2
    analogWrite(6,200); //Электродвигатель B1
```

```
digitalWrite(9,LOW); //Электродвигатель B2  
}
```

Стандартная процедура `analogWrite(порт, значение)` отправляет значение ширины импульса на соответствующий порт. Максимальной ширине импульса, а следовательно, максимальной скорости, соответствует значение 255. При таком значении тормозной путь робота слишком велик, что мешает ему эффективно реагировать на препятствия. Снижение скорости уменьшает длину тормозного пути. Как наиболее оптимальная была выбрана скорость, которой соответствует значение ширины импульса равное 200.

Процедура `digitalWrite` также является стандартной и работает аналогичным образом, но оперирует только двумя значениями: минимальное (LOW) и максимальное (HIGH).

Каждый электродвигатель подключен к двум портам, что позволяет управлять направлением их вращения. Один порт отвечает за вращение по часовой стрелке, а другой – против. Таким образом, запустив одновременное вращение одного двигателей в направлении соответствующем движению вперед, а вращение другого – назад, можно организовать поворот робота вокруг своей оси. Регулируя при этом время такого вращения, можно регулировать угол, на который повернется робот. По такому принципу работает процедура `Left`, которая осуществляет поворот налево:

```
void Left() { //Поворот налево  
digitalWrite(3,LOW);  
analogWrite(5,200);  
analogWrite(6,200);  
digitalWrite(9,LOW);  
delay(100); //Задержка для осуществления поворота  
}
```

Процедура поворота направо `Right` работает аналогично, только значения отправляются на порты, отвечающие за противоположное направление вращения электродвигателей. Процедура разворота `Turn`,

осуществляющая разворот робота вокруг своей оси, представляет собой более длительный поворот налево. В процедуре остановки Stop на все порты посылается минимальное значение LOW.

Определение наличия препятствия происходит с помощью датчика расстояния. Для удобства приходящие с датчика данные переводятся в сантиметры функцией distance_sm:

```
int distance_sm() { //Расстояние до препятствия в см
    byte i;
    int gp2=0;
    for(i=0;i<5;i++){ //Повторить 5 раз для фильтрации шума
        gp2=(gp2+analogRead(5)); //Чтение значения с датчика расстояния
    }
    gp2=gp2/5; //Вычисление среднего
    return (2914/(gp2+5))-1; //Преобразование данных в сантиметры
}
```

Поскольку датчик расстояния закреплен на вале сервомотора, перед считыванием данных необходимо обеспечить его правильное расположение. Для того чтобы датчик расстояния смотрел вперед по ходу движения робота, вал сервомотора должен быть в положение поворота на 90°. Для нахождения расстояния до препятствия слева или справа датчик должен быть повернут на 135° и 45° соответственно. Все манипуляции сервомотором и, следовательно, расположенным на нем датчиком осуществляются с помощью процедуры Servo_Move:

```
void Servo_Move(int val) { //Управление положением сервомотора
    byte d;
    for(d=0;d<20;d++){
        digitalWrite(8,HIGH); //Послать положительный импульс в порт 8 для
        осуществления поворота сервомотора
        delayMicroseconds(val); //Задержка для формирования положительного
        импульса
    }
}
```

```
digitalWrite(8,LOW); //Прекращение поворота сервомотора  
delay(10); //Задержка для формирования отрицательного импульса  
}  
}
```

Программная реализация остальных частей алгоритма произведена с использованием стандартных конструкций языка Arduino. Полный листинг программы разработанной модели роботизированной адаптивной системы приведен в Приложении 3.

Заключение

В ходе проведенного исследования были выявлены сущность и содержание кибернетического подхода к пониманию сложных систем. В результате анализа научно-технической информации по тематике исследования были выявлены определения таких понятий, как «кибернетический подход», «кибернетическая система», «обратная связь», «управление» и т.п.

В рамках данной работы были проанализированы основы моделирования кибернетических систем в робототехнике и выявлены основные признаки сложных роботизированных систем, такие как автономность, универсальность, автоматичность, антропоморфизм и адаптивность.

Были выделены основные составляющие робота как системы, активно взаимодействующей с окружающей средой. Это следующие подсистемы: управляющая, информационно-измерительная (сенсорная), подсистема связи, исполнительная (моторная).

Была разработана модель роботизированной адаптивной системы, которая самостоятельно решает задачу нахождения траектории движения с обходом препятствий от места старта до определенной цели. Разработанная модель обладает всеми признаками роботизированной системы. При ее создании были использован набор для сборки мобильного робота POP-BOT, который включает в себя стандартные технические средства, поэтому данная разработка может быть представлена как один из проектов по образовательной или соревновательной робототехнике. Сформулированная задача может быть расширена до нескольких видов целей или типов препятствий.

Список использованной литературы

1. Алесинская Т.В. Основы логистики. Общие вопросы логистического управления: учебное пособие / Т.В. Алесинская. – Таганрог: Изд-во ТРТУ, 2015. – 116 с.
2. Большая энциклопедия нефти и газа [Электронный ресурс]. Режим доступа URL: <http://www.ngpedia.ru/id634255p2.html>
3. Бондаренко В.М., Румянцева С.Ю., Дементьев В.Е., Клинов В.Г., Гринин А.Л., Малков С.Ю., Исаев Д.М., Шишкина А.Р. Кризисы и прогнозы в свете теории длинных волн / Л.Е. Гринин, А.В. Коротаев, Р.С. Гринберг. – М.: Учитель, 2016. – 367 с.
4. Брызгалина Е.В. Концепции современного естествознания: учебник / Е.В. Брызгалина. – СПб.: Проспект, 2014. – 561 с.
5. Василенко Н.В., Никитин К.Д., Пономарев В.П., Смолин А.Ю. Основы робототехники: учебное пособие / К.Д. Никитин. – М.: Эксмо, 2014. – 364 с.
6. Воячек, А.И. Основы проектирования и конструирования машин: учебное пособие / А.И. Воячек, В.В. Сенькин. – Пенза: Изд-во Пенз. гос. ун-та, 2012. – 228 с.
7. Гориневский Д.М., Формальский А.М., Шнейдер А.Ю. Управление манипуляционными системами на основе информации об усилиях / В.С. Гурфинкель, Е.А. Девянин. – М.: Физико-математическая литература, 2004. – 368 с.
8. Гринин Л.Е., Гринин А.Л. От рубил до нанороботов. Мир на пути к эпохе самоуправляемых систем: История технологий и описание их будущего / Л.Е. Гринин. – СПб.: Питер, 2015. – 424 с.
9. Задачи кибернетики и основные направления исследования [Электронный ресурс]. Режим доступа URL: <http://tenpercent.ru/03/19/zadachi-kibernetiki-i-osnovnie-napravleniya-issledovaniya/>
10. Зубарев Ю.А., Шамардин А.И. Энциклопедический словарь терминов по менеджменту, маркетингу, экономике, предпринимательству: учебное пособие в двух томах. Том 1. – Волгоград: Волгоградское научное издательство, 2012. – 561 с.

11. Информационные системы виртуальной реальности в мехатронике и робототехнике: учебное пособие / Алферов Г.В., Кулаков Ф.М., Нечаев А.И., Чернакова С.Э. – СПб.: «СОЛО», 2006. – 146 с.
12. Кабулов В.К. Алгоритмическое направление в кибернетике. Некоторые аспекты. – М.: Наука, 2014. – 79 с.
13. Кибернетика [Электронный ресурс]. Режим доступа URL: http://gufo.me/content_naukteh/kibernetika-64069.html
14. Крамер Г. Математические методы статистики в пер. с англ. А.С. Монин, А.А. Петров / А.Н. Колмогоров. – М.: Наука, 2004. – 648 с.
15. Лопатников Л.И. От плана к рынку: карманная энциклопедия / Л.И. Лопатников. – М.: Издательский дом «Дело» РАНХиГС, 2013. – 168 с.
16. Лоцманенко В.В., Кочегаров Б.Е. Проектирование и конструирование (основы): учебное пособие. – Владивосток: Изд-во ДВГТУ, 2012. – 96 с.
17. Майер Р.В. Компьютерное моделирование: учебно-методическое пособие для студентов педагогических вузов / Р.В. Майер. – Глазов: ГГПИ, 2015. – 620 с.
18. Маслов Р.В. Позднева С.П. Компьютеризация науки как путь к диалогу и интеграции / Р.В. Маслов, С.П. Позднева // Известия Саратовского университета. Новая серия. Серия Философия. Психология. Педагогика. - № 7. – 2017. – С.44-53.
19. Никитин К.Д. Основы робототехники: учебник / К.Д. Никитин. – М.: Наука, 2013. – 468 с.
20. Обратная связь [Электронный ресурс]. Режим доступа URL: <http://bse.sci-lib.com/article083233.html>
21. Ожегов С.И. Толковый словарь русского языка / С.И. Ожегов. – М.: Оникс, 2013. – 736 с.
22. Пономарева В.М., А.П. Литвинова. Основы автоматического регулирования и управления: учебное пособие для неэлектротехн. специальностей вузов / В.М. Пономарева, А.П. Литвинова. – М.: Высшая школа, 2015. – 439 с.
23. Пшихопов В.Х., Медведев М.Ю., Костюков В.А., Гайдук А.Р., Федоренко Р.В., Гуренко Б.В., Крухмалев В.А., Медведева Т.Н. Проектирование роботов и робототехнических систем: учебное пособие – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 196 с.

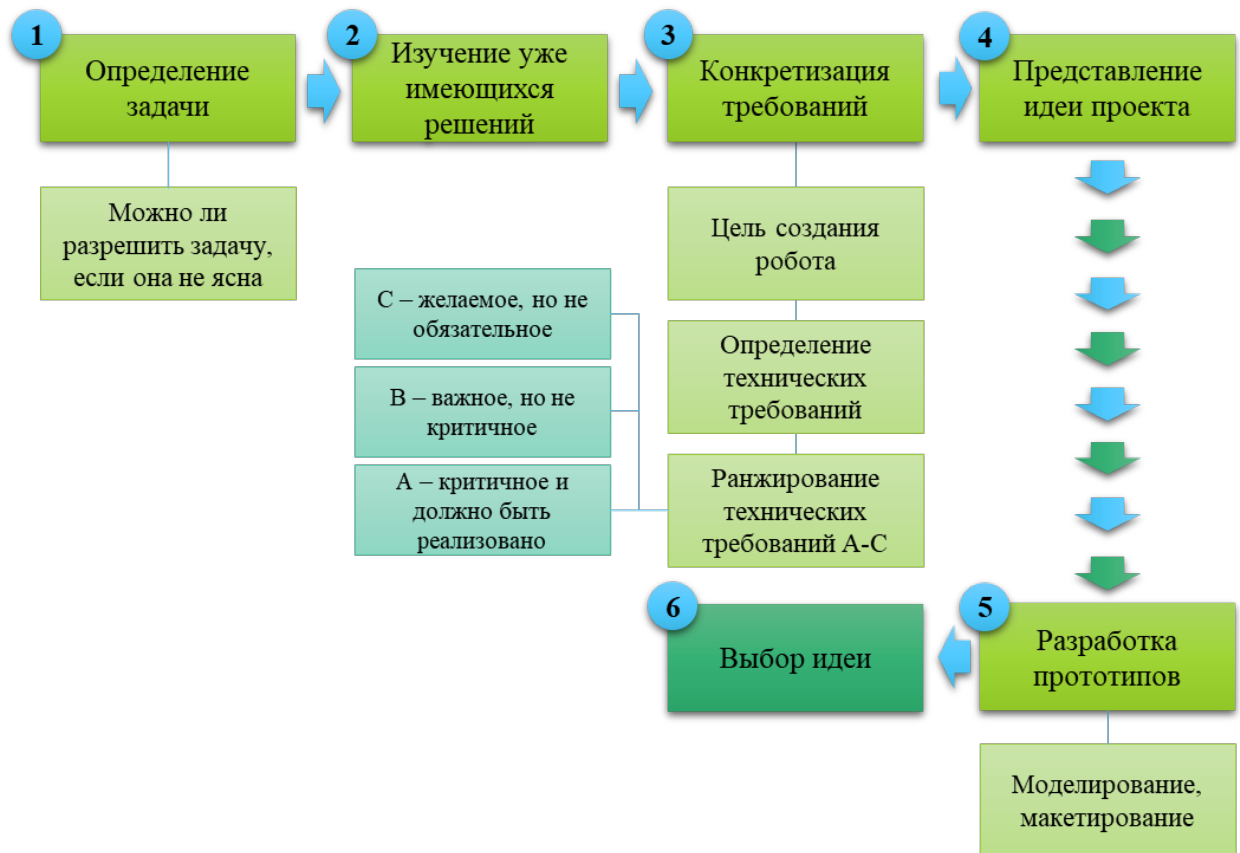
24. Розанова Л.В. Основы кибернетики: конспект лекций / Л.В. Розанова. – Омск: Издательство ОмГТУ, 2015. – 60 с.
25. Сафронова Н.В. Робототехника как инновационное направление обучения информатике в школе // Вестник ФГБОУ ВПО. – 2014. - № 6
26. Система управления, закономерности и принципы управления [Электронный ресурс]. Режим доступа URL: <http://center-yf.ru/data/Menedzheru/Sistema-upravleniya-zakonomernosti-i-principy-upravleniya.php>
27. Соболев С.Л., Китов А.И., Ляпунов А.А. Основные черты кибернетики [Электронный ресурс]. Режим доступа URL: <http://www.computer-museum.ru/books/cybernetics.htm>
28. Советов Б.Я., Яковлев С.А. Моделирование систем: учебник для вузов. – М.: Высшая школа, 2012. – 343 с.
29. Суворова С.С., Колосовская Т.А., Подгорбунских А.А. Проектная и исследовательская деятельность обучающихся: теория и практика: монография / С.С. Суворова, Т.А. Колосовская, А.А. Подгорбунских. – Шадринск: ШГПИ, 2015. – 175 с.
30. Ушаков А.В. Прикладная теория информации: элементы теории и практикум: учебное пособие для вузов. – СПб.: СПбНИУИТМО, 2012. – 325 с.

Приложение 1

Характерные особенности робота



Этапы создания робота



Листинг программы роботизированной адаптивной системы

```
int TF[30]; //время движения на прямолинейных отрезках траектории
int LR[30]; //количество и направление поворотов между прямолинейными
отрезками
byte N; //количество отрезков траектории
byte ST; //идентификатор этапа

void Forward() { //Двигаться вперед
    analogWrite(3,200);
    digitalWrite(5,LOW);
    analogWrite(6,200);
    digitalWrite(9,LOW);
}

void Stop() { //Стоп
    digitalWrite(3,LOW);
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(9,LOW);
    delay(10);
}

void Left() { //Поворот налево
    digitalWrite(3,LOW);
    analogWrite(5,200);
    analogWrite(6,200);
    digitalWrite(9,LOW);
    delay(100);
}

void Right() { //Поворот направо
    analogWrite(3,200);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    analogWrite(9,200);
    delay(100);
}
```

```

void Turn(){ //Разворот
  digitalWrite(3,LOW);
  analogWrite(5,200);
  analogWrite(6,200);
  digitalWrite(9,LOW);
  delay(340);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
}

```

```

void Servo_Move(int val){ //Управление положением сервомотора
  byte d;
  for(d=0;d<20;d++){
    digitalWrite(8,HIGH); //Осуществления поворота сервомотора
    delayMicroseconds(val); //Формирование положительного импульса
    digitalWrite(8,LOW); //Прекращение поворота сервомотора
    delay(10); //Формирование отрицательного импульса
  }
}

```

```

int distance_sm(){ //Вычисление расстояние до препятствия в см
  byte i;
  int gp2=0;
  for(i=0;i<5;i++){ //Повторить 5 раз для фильтрации шума
    gp2=(gp2+analogRead(5)); //Чтение значения с датчика расстояния
  }
  gp2=gp2/5; //Вычисление среднего
  return (2914/(gp2+5))-1; //Преобразование данных в сантиметры
}

```

```

boolean Target(){ //Определение достижения цели
  int LS, RS;
  LS=analogRead(7); //Чтение значения с левого отражательного датчика
  RS=analogRead(6); //Чтение значения с правого отражательного датчика
  return (LS<500)&&(RS<500); //Определения достижения белой области
}

```

```

}
void Learn_TF(byte n){ //Нахождение времени движения n-го прямолинейного
отрезка
    byte dis=distance_sm();
    int TF0;
    TF0=millis(); //Определение времени в начале прямолинейного участка
    while((dis>20)&&(Target())){
        Forward();
        Servo_Move(1340); //Поворот датчика расстояния в положение вперед
        dis=distance_sm(); //Определение расстояния впереди
    }
    TF[n]=millis()-TF0; //Вычисление времени, затраченного на прямолинейный
участок
    Stop();
}

```

```

void Learn_LR(byte n){ //Нахождение количества и направления поворотов
после n-го прямолинейного отрезка
    int dis, disL, disR;
    Servo_Move(1340);
    dis=distance_sm();
    LR[n]=0;
    while(dis<20){
        Servo_Move(1660); //Поворот датчика расстояния налево
        disL=distance_sm();
        Servo_Move(760); //Поворот датчика расстояния направо
        disR=distance_sm();
        if(disR>disL){
            Right();
            LR[n]=LR[n]+1;
        }
        else{
            Left();
            LR[n]=LR[n]-1;
        }
        Stop();
        Servo_Move(1340);
        dis=distance_sm();
    }
}

```

```
}
```

```
void Learn() { //Обучение: нахождение траектории движения от места старта к цели
```

```
    Learn_TF(N);
    if(Target()){
        Learn_LR(N);
    }
    else{
        N=N-1;
        ST=2;
    }
    N=N+1;
}
```

```
void To_Start() { //Движение от цели к месту старта по известной траектории
```

```
    int i, j;
    Turn();
    Forward();
    delay(TF[N]);
    for(i=N-1; i>=0; i--){
        j=LR[i];
        do{
            if(j>0){
                Right();
                j=j-1;
            }
            else{
                Left();
                j=j+1;
            }
        } while(j!=0);
        Forward();
        delay(TF[i]);
    }
    Stop();
    ST=3;
}
```

```

void To_Target() { //Движение от места старта к цели по известной траектории
  int i, j;
  Turn();
  Forward();
  delay(TF[0]);
  for(i=1;i<=N;i++){
    j=LR[i-1];
    do{
      if(j>0){
        Right();
        j=j-1;
      }
      else{
        Left();
        j=j+1;
      }
    } while(j!=0);
    Forward();
    delay(TF[i]);
  }
  Stop();
  ST=2;
}

```

```

void setup() { //Установки
  pinMode(3,OUTPUT); //Электродвигатель A1
  pinMode(5,OUTPUT); //Электродвигатель A2
  pinMode(6,OUTPUT); //Электродвигатель B1
  pinMode(9,OUTPUT); //Электродвигатель B2
  pinMode(8,OUTPUT); //Сервомотор
  ST=1; //Идентификатор этапа
  N=0; //Номер последнего элемента массива TF
}

```

```
void loop() { //Основной цикл
  switch(ST){
    case 1: //Этап 1
      Learn(); //Обучение
      break;
    case 2: //Этап 2
      To_Start(); //Движение от цели к месту старта
      break;
    case 3: //Этап 3
      To_Target(); //Движение от места старта к цели
      break;
  }
}
```