



ИПМ им.М.В.Келдыша РАН

Онлайновая библиотека



А.А.Богуславский, Г.К.Боровин,  
В.А.Карташев, В.Е.Павловский,  
С.М.Соколов

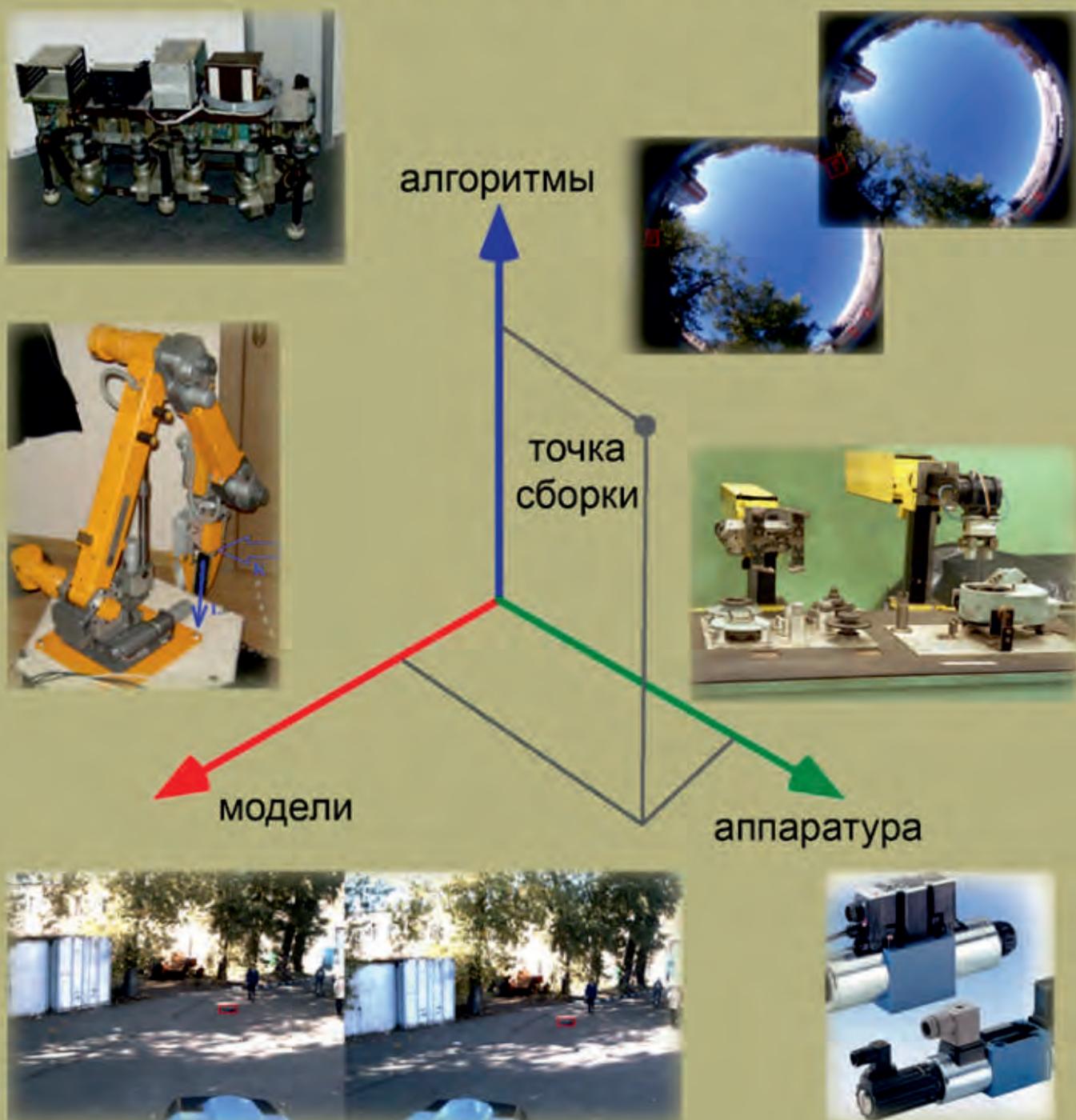
Модели и алгоритмы  
для интеллектуальных  
систем управления

***Рекомендуемая форма библиографической ссылки***

Богуславский А.А., Боровин Г.К., Карташев В.А., Павловский В.Е., Соколов С.М. Модели и алгоритмы для интеллектуальных систем управления. М.: ИПМ им.М.В.Келдыша, 2019. 228 с.  
<https://doi.org/10.20948/mono-2019-boguslav>  
<https://keldysh.ru/e-biblio/boguslav/index.pdf>

А.А. Богуславский, Г.К. Боровин, В.А. Карташев  
В.Е. Павловский, С.М. Соколов

# МОДЕЛИ И АЛГОРИТМЫ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ



*А.К. Платонову посвящается*



**Александр Константинович Платонов**

03.01.1931 – 17.09.2017

Доктор физико-математических наук, профессор, крупнейший специалист в областях баллистики и процессов управления сложными механическими системами, лауреат Ленинской и Государственной премий, заслуженный деятель науки РФ, награжден дважды Орденом Трудового Красного Знамени и другими государственными наградами.

Александр Константинович был вдохновителем написания данной книги и принимал активное участие в обсуждении её содержания.

**А.А. Богуславский, Г.К. Боровин, В.А. Карташев  
В.Е. Павловский, С.М. Соколов**

**Модели и алгоритмы  
для интеллектуальных систем управления**

*Рекомендовано к изданию  
редакционно-издательским советом  
ИПМ им. М.В. Келдыша РАН*

ИПМ им. М.В. Келдыша  
Москва – 2019

УДК 681.51:623(063)

ББК 32.965я43

М 744

Модели и алгоритмы для интеллектуальных систем управления / А.А. Богуславский, Г.К. Боровин, В.А. Карташев и др. — М.: ИПМ им. М.В. Келдыша, 2019. — 228 с.

ISBN 978-5-98354-050-7

В монографии собраны результаты исследований по разработке математических моделей и алгоритмов интеллектуальных систем управления для подвижных робототехнических систем различного целевого назначения, отличающихся конструкцией движителей, элементами мехатроники и платформами реализации программного обеспечения. Описаны методы и средства развития интеллектуальной составляющей во всех уровнях систем управления роботами. Исследование выполнено методами концептуального и логического анализа с построением моделей предметной области, требуемых для функционирования каждой из рассматриваемых технических систем, с построением математических моделей механики их силовых элементов и ограничений наблюдаемости их операционных сред. Алгоритмическое обеспечение исследуется на предмет его реализуемости в масштабе реального времени с учётом доступных вычислительных ресурсов и возможности унифицированного использования. Акцентируется внимание на программной составляющей интеллектуальных систем управления. Предлагается общая схема компоновки таких систем на основе программного обеспечения наиболее ресурсоёмкой и унифицированной части систем информационного обеспечения – системы технического зрения.

Рецензент: доктор технических наук, заведующий кафедрой  
автоматизированных систем управления НИТУ «МИСиС» И.О. Темкин

Работа выполнена при финансовой поддержке Российского научного фонда  
(проект №16-19-10705)

ISBN 978-5-98354-050-7

© ИПМ им.М.В.Келдыша, 2019

# Оглавление

<b>ПРЕДИСЛОВИЕ .....</b>	<b>7</b>
<b>ВВЕДЕНИЕ .....</b>	<b>10</b>
<b>1. СИЛОВЫЕ СИСТЕМЫ РОБОТОВ И ИХ МАТЕМАТИЧЕСКИЕ МОДЕЛИ.....</b>	<b>23</b>
1.1. УПРАВЛЕНИЕ С ВВЕДЕНИЕМ ПРОГНОЗИРУЮЩЕЙ МОДЕЛИ .....	27
1.2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ .....	29
1.2.1. <i>Электромеханические системы управления.....</i>	<i>30</i>
1.2.2. <i>Гидравлические системы управления.....</i>	<i>43</i>
1.3. ГИДРАВЛИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ЭКЗОСКЕЛЕТОНА И РЕЗУЛЬТАТЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ .....	79
1.4. НЕКОТОРЫЕ ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ЭЛЕКТРОМЕХАНИЧЕСКИХ И ГИДРАВЛИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ .....	102
ЛИТЕРАТУРА.....	114
<b>2. МАНИПУЛЯЦИОННЫЕ РОБОТЫ И ИХ ЗАДАЧИ.....</b>	<b>118</b>
2.1. КЛАССИФИКАЦИЯ МАНИПУЛЯЦИОННЫХ РОБОТОВ.....	121
2.2. КИНЕМАТИЧЕСКИЕ СХЕМЫ МАНИПУЛЯТОРОВ.....	128
2.2.1. <i>Ортгогональная кинематическая схема.....</i>	<i>128</i>
2.2.2. <i>Цилиндрическая кинематическая схема .....</i>	<i>129</i>
2.2.3. <i>Сферическая кинематическая схема.....</i>	<i>130</i>
2.2.4. <i>Антропоморфная кинематическая схема .....</i>	<i>131</i>
2.2.5. <i>Робот типа SCARA.....</i>	<i>133</i>
2.2.6. <i>Манипуляторы с параллельной кинематической схемой .....</i>	<i>134</i>
2.3. ОПЕРАЦИОННАЯ СРЕДА ФУНКЦИОНАЛЬНЫХ ЗАДАЧ ПРИМЕНЕНИЯ МАНИПУЛЯТОРОВ .....	135
2.3.1. <i>Организованное и неорганизованное рабочее пространство .....</i>	<i>135</i>
2.4. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ МАНИПУЛЯЦИОННЫМИ РОБОТАМИ С ЭЛЕМЕНТАМИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА .....	137
2.4.1. <i>Унифицированные алгоритмы функций нижнего уровня управления и их развитие.....</i>	<i>140</i>
2.4.2. <i>Унифицированные алгоритмы функций верхнего уровня управления и их развитие.....</i>	<i>143</i>
ЛИТЕРАТУРА.....	151
<b>3. БИОЛОГИЧЕСКИ ИНСПИРИРОВАННЫЕ НЕЙРО- НАВИГАЦИОННЫЕ СИСТЕМЫ ДЛЯ МОБИЛЬНЫХ ШАГАЮЩИХ РОБОТОВ.....</b>	<b>153</b>
3.1. НЕЙРОСЕТИ. КРАТКОЕ ОПИСАНИЕ .....	153
3.2. ШЕСТИНОГИЙ РОБОТ: ЛОКАЛИЗАЦИЯ .....	157
3.2.1. <i>Описание модели.....</i>	<i>157</i>

3.2.2. Методика расчетов.....	159
3.2.3. Определение архитектуры нейросети.....	161
3.2.4. Результаты моделирования.....	161
3.3. МЕТОД ЛОКАЛИЗАЦИИ ДВУНОГОГО АППАРАТА В ПРОСТРАНСТВЕ .....	163
3.3.1. Постановка задачи.....	163
3.3.2. Описание модели.....	163
3.3.3. Методика расчетов.....	164
3.3.4. Определение архитектуры сети.....	166
3.3.5. Результаты численного моделирования и анализ чувствительности.....	168
3.4. МЕТОД ЛОКАЛИЗАЦИИ ГОЛОВЫ ОТНОСИТЕЛЬНО КОРПУСА.....	170
3.4.1. Постановка задачи.....	171
3.4.2. Модель движения головы, модель измерений.....	172
3.4.3. Методика расчетов.....	173
3.4.4. Результаты численного моделирования и анализ чувствительности.....	175
ЛИТЕРАТУРА .....	177
<b>4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИСУ НА БАЗЕ СТЗ.....</b>	<b>178</b>
4.1. ФУНКЦИИ СТЗ В СОСТАВЕ ИСУ.....	178
4.1.1. Особенности реализации ПО ИСУ с использованием СТЗ.....	182
4.1.2. Примеры СТЗ реального времени.....	184
4.2. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К РАЗРАБОТКЕ ПО.....	186
4.2.1. Итеративные схемы применения объектно-ориентированного подхода.....	191
4.2.2. Выделение классов в ПО СТЗ.....	195
4.3. ШАБЛОНЫ ПРОЕКТИРОВАНИЯ ПО.....	200
4.3.1. Шаблоны проектирования архитектурного уровня.....	205
4.3.2. Шаблоны проектирования для обработки данных в реальном времени.....	206
4.4. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА .....	209
4.4.1. Базовый язык программирования.....	209
4.4.2. Базовая операционная система.....	213
4.4.3. Использование открытого ПО Robot Operating System.....	218
4.5. ВЫВОДЫ.....	219
ЛИТЕРАТУРА .....	221
<b>ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ.....</b>	<b>227</b>

## Предисловие

В последние годы в различных технических сферах активно используется прилагательное «интеллектуальный» и интеллектуальные системы управления в частности. У этих терминов довольно большая история и широкие дискуссии по уточнению соответствующих определений. Основным приложением, на которое ориентированы подобные системы, является робототехника.

Как справедливо отмечают авторы книги, одной из современных тенденций робототехники является переход от экспериментальных к широкому, практическому использованию не только промышленных роботов, но и сервисных, предназначенных для работы в заранее неопределённых или недоопределённых условиях. На передний план выступают задачи практической реализации различных подходов, методов унификации разнообразных алгоритмов управления, позволяющих экономически целесообразно создавать желаемые устройства. Вместе с тем, не снимается с повестки дня и формирование перспективного задела во всех составляющих мехатронных систем: механической, сенсорной, вычислительной. Повышение степени автономности робототехнических комплексов требует снабжения систем управления информацией о среде функционирования и условиях в ней. Эту задачу призваны решать системы информационного обеспечения и системы технического зрения, как один из центральных элементов осязательного робототехнических комплексов. Все составляющие систем управления, начиная от двигателей и приводов и заканчивая целеполаганием в действиях во внешнем мире требуют построения моделей объектов и процессов управления. Сбор и обработка сенсорных данных о среде функционирования и состоянии самой технической системы – одна из основных сфер приложения методов искусственного интеллекта в робототехнике и автоматизации производства.

Основным средством реализации алгоритмического обеспечения является программное обеспечение. Описанию результатов исследований перечисленных аспектов создания интеллектуальных систем управления и посвящена настоящая работа.

В книге А.А. Богуславского, Г.К. Боровина, В.А. Карташева, В.Е. Павловского, С.М. Соколова сконцентрирован многолетний опыт авторов известной школы робототехники Института прикладной математики имени М.В. Келдыша РАН по исследованию систем управления сложными механическими и робототехническими системами средствами математического, компьютерного, натурального и полунатурального моделирования.

Книга интересна широким охватом составляющих робототехнических комплексов (РТК) на всех уровнях систем управления и методов, используемых для выработки управляющих воздействий. Здесь и регуляторы электромеханических и гидравлических приводов, и глубокий анализ типов

манипуляторов и систем их управления, и оригинальное предложение нейросетевых подходов к управлению шагающими машинами.

Интересно для отечественной практики предложение по объединяющему программному инструментарию при реализации алгоритмического обеспечения подвижных РТК. Заслуживающим внимания является довольно объёмный перечень используемых в книге понятий и терминов. В отечественной практике только принимаются и осваиваются стандарты терминологии в рассматриваемых областях. Приведённые определения позволяют не только яснее понять мысли авторов, но и сопоставить их с другими известными толкованиями и будут способствовать улучшению профессиональной терминологии.

Для повышения эффективности процесса проектирования сложных РТК полезно предложение о рассмотрении сборки таких комплексов в пространстве конфигураций, группирующем большое количество технологий, используемых при создании РТК, в три основных направления: модели, аппарата, алгоритмическое обеспечение.

В книге рассмотрены схемы электромеханических и гидравлических систем управления робототехническими комплексами, представлены их математические модели и структурные схемы. Созданный программный комплекс для математического моделирования асинхронных электроприводов с векторным управлением использовался при проектировании микропроцессорного управления роботов. В качестве примеров практического использования гидроприводов рассмотрены шагающий аппарат и экзоскелет. Обосновано использование LS регуляторов для системы управления шагающей машины. Приведены результаты математического моделирования динамики гидропривода ноги шагающей машины в системе “Matlab+Simulink”.

Для верхних уровней управления шагающими машинами исследованы биоинспирированные нейросетевые методы. Построено решение задач локализации для шагающих роботов (шестиногого и андроидного двуногого) в нейросетевом базисе. Проведен анализ применимости метода и его чувствительности к изменениям параметров задачи (характеристикам ошибок измерений, геометрическим параметрам). Построены сети, определяющие навигационные параметры для систем с разным уровнем погрешностей датчиков.

Разработку программного обеспечения интеллектуальных систем управления (ПО ИСУ) предлагается проводить на базе компонентного программного каркаса. Это крупномасштабный архитектурный шаблон, предполагающий расширение для разработки приложений в конкретной прикладной области, в данном случае, в области разработки ПО для управления РТК. В этом каркасе предусмотрены подсистемы ввода зрительных данных, интерфейс с оператором, обработка зрительных данных. Эти подсистемы функционируют параллельно и оформляются в виде отдельных потоков. Реализуется компонентный каркас в виде классов на языке Си++. Использование компонентного каркаса позволяет выполнить

интеграцию с внешними программными системами. В частности, возможна реализация связи с компонентами в рамках распространенной в области робототехники программной системой ROS. Для этой цели возможна интеграция в каркасе программных компонентов, реализующих интерфейсы узлов системы ROS.

Признанным средством обоснования эффективности решений в области робототехники является экспериментальная проверка. Большинство результатов, описанных в книге такую проверку прошли, как минимум в компьютерных моделях, а, зачастую, и в действующих образцах РТК.

Авторы формулируют ряд общих рекомендаций, могущих оказаться полезными для разработчиков мобильных роботов различного назначения. Некоторая часть материала книги разрозненно опубликована в виде журнальных статей и статей в материалах конференций, как на русском языке, так и на английском. Собранный воедино, этот материал, несомненно, будет интересен специалистам по робототехнике, автоматизации и управлению, как систематическое изложение новых разработанных авторами методов управления на всех уровнях перспективных робототехнических систем и их элементов, а также подходов к проектированию таких систем.

Авторы монографии – известные в нашей стране и за рубежом специалисты в области робототехники, теории управления, систем искусственного интеллекта, компьютерного видения. Книга А.А. Богуславского, Г.К. Боровина, В.А. Карташева, В.Е. Павловского, С.М. Соколова «Модели и алгоритмы для интеллектуальных систем управления» будет полезна широкому кругу специалистов по робототехническим и сложным техническим системам, студентам и аспирантам, занимающимся построением эффективных интеллектуальных систем управления.

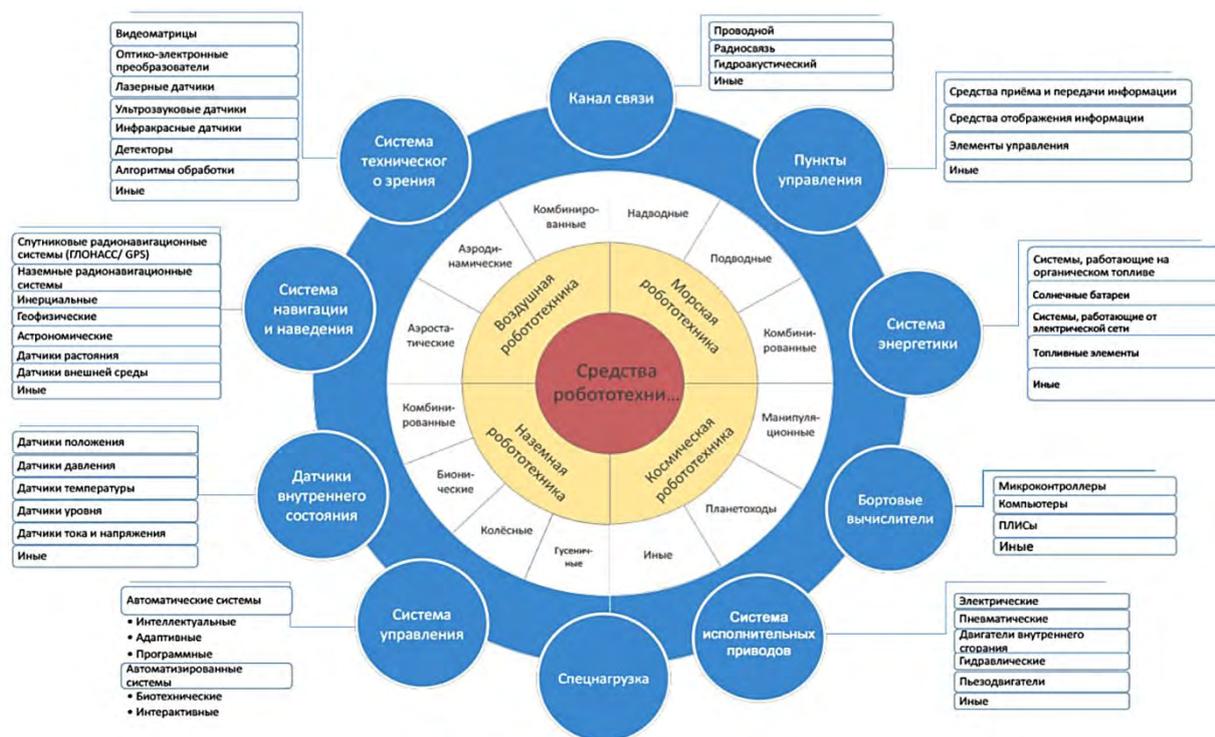
Заведующий кафедрой автоматизированных систем управления ИТАСУ  
НИТУ «МИСиС», доктор технических наук, профессор

И.О. Тёмкин

*Расчлените каждую изучаемую вами задачу на столько частей, сколько потребуется, чтобы их было легко решить*  
*Р. Декарт*

## Введение

В настоящее время всё в более широком круге областей используются приёмы повышения эффективности технических систем, которые часто называются интеллектуализацией. Отправной точкой и основной сферой приложения такого подхода является робототехника – научно-техническое направление, объединяющее самые передовые достижения и разработки в области механики, сенсорики, вычислительной техники, искусственного интеллекта и алгоритмических решений. В робототехнических разработках задействованы порядка 10 основных технологий. Известна так называемая схема «ромашка» – схема основных компонент робототехнических систем (РТС). По этой схеме предлагалось распределить разработчиков и исполнителей отечественной программы робототехники (рис. 1).



*Рис. 1. Основные направления развития технологий робототехники (предложены и одобрены девятой Всероссийской научно-практической конференции «Перспективные системы и задачи управления»).*

Одной из характеристик современного состояния мировой робототехники является переход к практическому использованию робототехнических комплексов (РТК) с повышенной степенью автономности

(ПСА). Под повышенной степенью автономности подвижных средств понимаем такие свойства бортовых систем управления как:

- Реализация операций согласованного управления нижним уровнем мехатронных систем (информационно-двигательными действиями) (приводами, сбором данных)
- Постановка и реализация подцелей в процессе решения основной задачи (безопасность; выбор пути; задача возврата; выбор места взятия известного объекта; взлёт, посадка ЛА; обеспечение «плавной деградации» и «мягкого отказа»)
- «Естественность» языка общения – интерфейса подвижного средства с верхним уровнем целеполагания (стратегическим).

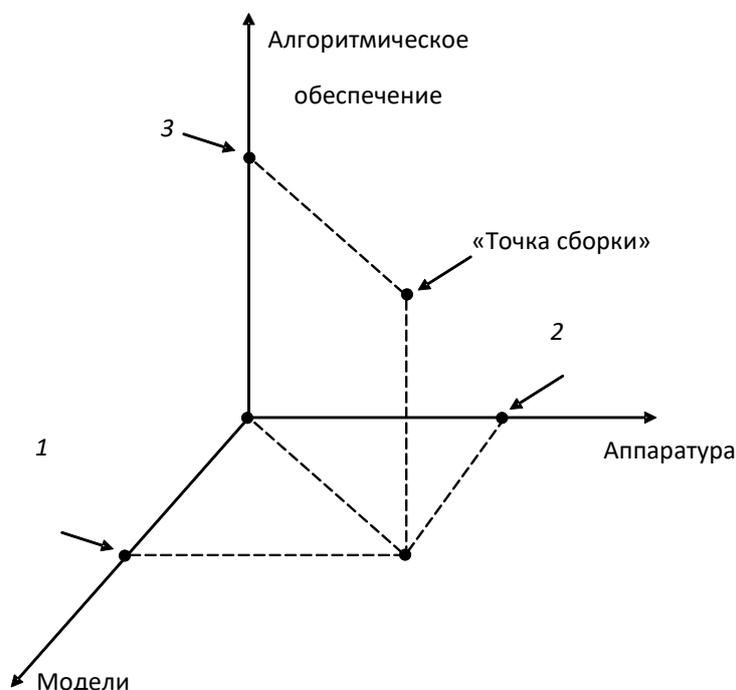
Наибольшее развитие среди роботов с ПСА в настоящее время получили беспилотные устройства (одиночных и групповых) с повышенной степенью автономности и полностью автономных. После проведения соревнований по целенаправленным перемещениям автономных наземных транспортных средств под эгидой американского агентства передовых оборонных исследовательских проектов Пентагона DARPA (Defense Advanced Research Projects Agency) [1, 2] активизировались работы по созданию коммерческих версий подобных систем. [3-8].

В качестве примеров одних из наиболее интересных, некоммерческих проектов автономных мобильных средств можно указать на два американских проекта, имеющих более чем десятилетнюю историю. Программа американского космического агентства NASA создания и использования марсоходов (1997 – по наст. вр.) [9]. Программа американского агентства DARPA создания беспилотных летательных аппаратов, в частности, проект X47B [10]. В научных публикациях много работ с описанием лабораторных макетов мобильных роботов с повышенной степенью автономности.

В других областях робототехники успехи внедрения РТК с ПСА не столь значительны. В частности, не индустриальное использование манипуляционных роботов весьма ограничено, что в значительной степени объясняется высокой стоимостью механической части, выводящей практическое использование из области экономической целесообразности. Подобные и другие причины, такие как автономная энергетика, недостаточная степень миниатюризации, сдерживают внедрение разнообразных РТК. Тем не менее, очевидно, что с развитием механической и элементной базы проблема аппаратной реализуемости будет терять свою остроту. И тогда на первый план будут выходить вопросы эффективности алгоритмического и компактности программного обеспечения. Именно поэтому, исследования в области интеллектуальных систем управления в теоретическом плане целесообразно проводить по всем возможным направлениям

Отдельно взятые, пусть и лучшие в своей области результаты, как составная часть робототехнической системы не могут автоматически обеспечить успех всего проекта. Необходимы дополнительные, комплексные исследования и разработки для эффективного интегрирования этих

результатов в робототехнический комплекс. Для конструктивного обсуждения методов и технологий разработки эффективных систем управления робототехническими комплексами (СУ РТК) предлагаем уменьшить размерность исходного пространства технологий и рассматривать точку сборки роботов с повышенной степенью автономности в пространстве конфигураций РТК (рис. 2). Для краткости будем использовать термин подвижный робототехнический комплекс (ПРТК). В существующей практике слово «мобильный» чаще связывают с наземными средствами, манипуляционные роботы также имеют своё закрепившееся имя, воздушные средства подразделяются на БЛА самолётного, вертолётного типа и мультикоптеры, термином космические роботы обобщённо называют все роботоподобные устройства, используемые в космосе. Результаты наших исследований применимы ко всем перечисленным категориям устройств, которые обобщённо можно назвать сложными механическими системами или подвижными робототехническими комплексами.



*Рис. 2.* Пространство конфигураций/архитектур подвижных РТК с ПСА. 1 – модели объекта управления (РТК с ПСА) и среды функционирования, постановка задачи управления в этих модельных представлениях; 2 – набор аппаратных средств с фиксированными техническими характеристиками (механические, сенсорные и вычислительные средства); 3 – набор алгоритмов решения целевой задачи.

Точки по осям предложенной системы координат конфигурационного пространства можно рассматривать как «точки со структурой». Эта структура масштабируема и подобна—общей структуре системы управления робототехническими комплексами с повышенной степенью автономности. СУ РТК с ПСА.

Точка по оси моделей – это элемент одного из уровней составляющих общую структуру моделей РТК и среды функционирования: модель привода; модель/представление особенностей (ориентиров/подцелей) окружающего пространства, получаемых от сенсоров; модель/представление рабочей области объекта управления; модель условий функционирования; модель решаемой задачи, возможные сценарии действий.

Точка по оси аппаратуры – отдельные приводы; сенсоры; совокупности приводов – технические устройства (подвижные платформы, манипуляторы, СТЗ, и т.п.), подвижные платформы с полезной бортовой нагрузкой.

Точка по оси алгоритмического обеспечения: нижний уровень – алгоритм управления отдельным приводом, алгоритм сбора данных, алгоритм выделения первичных признаков объектов; алгоритм реализации информационно-двигательного действия (ИДД); алгоритм реализации совокупности ИДД (реализация сценариев выполнения прецедентов использования ПРТК с ПСА);

Процесс компоновки/анализа ИСУ подвижного РТК для решения заданного класса задач с использованием предложенного конфигурационного пространства итерационный. Последовательность шагов компоновки/анализа ИСУ подвижного РТК с ПСА следующая.

1. Формулировка формализованного описания задачи (модели) на языке описания объектов управления/интереса, внешней среды и необходимых действий (преобразований) с объектами интереса в пространстве его состояний. (Плодотворным подходом этого этапа компоновки зарекомендовал себя метод построения онтологий предметной области.)  
Результат – модель задачи.
2. Формируется модель внешней среды и работа в ней.
3. Выбираются аппаратные средства, (механика, энергетика, сенсорика, бортовая вычислительная среда) обеспечивающих получение необходимых исходных данных (для формирования модели внешней среды).
4. Формирование комплекса алгоритмов для информационного обеспечения решения целевой задачи.
5. Проверка возможности нахождения/достижимости всех вышеуказанных компонент и подсчёт стоимости владения.
6. Определение: компоновка приемлема? Если да, то процесс закончен, если нет, то переход к следующей итерации (пп. 1-5).

Состав исходных данных к началу создания робота с ПСА:

- Прецеденты использования МР с ПСА с количественными характеристиками требований к целенаправленным перемещениям
- Условия функционирования МР с ПСА – модель среды, фоноцелевая обстановка
- Кинематическая и динамическая модели подвижной мехатронной платформы

- Условия согласования (УС) информационно-двигательных действий для решения задачи целенаправленных перемещений
- Экономически целесообразная стоимость МР с ПСА

Представленные в настоящей работе исследования выполнены с учётом указанной и ряда других современных тенденций развития робототехники и мехатроники. Среди этих тенденций выделены.

**Переход от уникальных, экспериментальных образцов, к широкому внедрению в практику.** Эта тенденция определяет требование технологичности реализации конкретных решений, т.е. сочетания реализуемости, надёжности, комплексности и эффективности на всех уровнях систем управления (СУ) (от управления приводами и первичной обработки сенсорных данных до принятия решений о выборе целей и траекторий движения) с экономически целесообразными ограничениями совокупной стоимости владения.

Как уже было отмечено, наиболее показательны в этой области примеры из автомобилестроения. На рис. 3 и 4 представлены диаграммы, иллюстрирующие основные вехи в развитии систем автономного управления движением.



Рис. 3. История технологического развития САУД.

## ИСТОРИЯ ЭКСПЛУАТАЦИИ АВТОМОБИЛЕЙ, ОСНАЩЕННЫХ СИСТЕМАМИ АВТОНОМНОГО УПРАВЛЕНИЯ ДВИЖЕНИЕМ



Рис. 4. История эксплуатации ТС, оснащенных САУД.

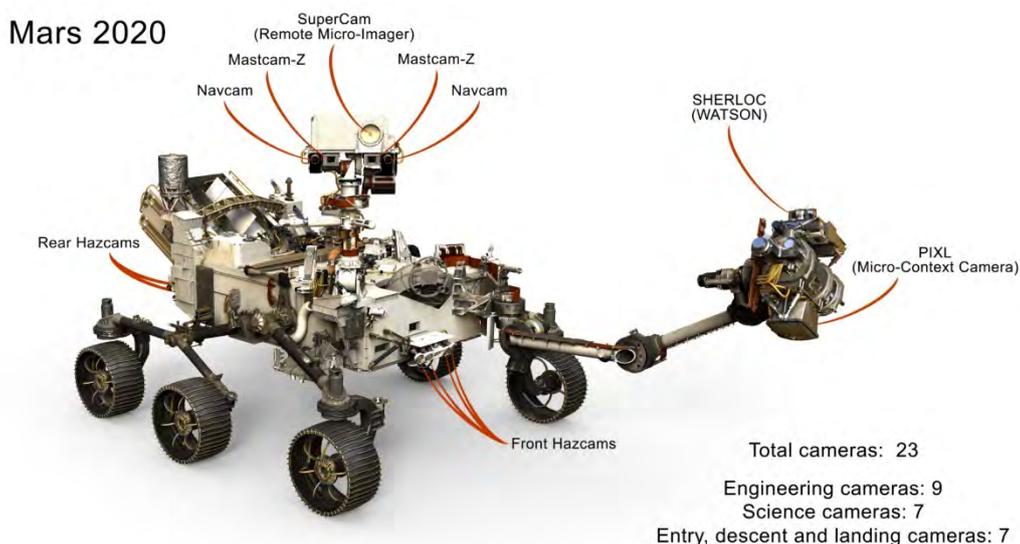
**Возрастание роли программного обеспечения в составе робототехнических комплексов.** Одним из наиболее показательных примеров названной тенденции в области широко тиражируемой мобильной техники является всё возрастающая доля микропроцессорных систем и их программного обеспечения в автомобильной промышленности. В наиболее продвинутой высокотехнологичной области – космических исследованиях примером возрастания роли программного обеспечения в составе робототехнических комплексов является структура цены американских марсоходов (разработка: NASA и Jet Propulsion Lab). Стоимость разработки СТЗ: 600000\$ аппаратура + 1000000\$ на разработку ПО и отладку. [] (Источник: Univ. of Regina, Champion's Brag, Vol. 9, 2003)

**Возрастание роли СТЗ в составе СИО мобильных РТК.** Среди средств информационного обеспечения подвижных роботов с повышенной степенью автономности одну из главных позиций занимают системы технического зрения (СТЗ). Привлекательными свойствами систем технического зрения являются:

- дистанционный анализ окружающего пространства и объектов интереса в нем
- одновременный анализ больших областей пространства;
- богатство признаков, которыми могут описываться и по которым идентифицируются объекты интереса;

- быстродействие: скорость выполнения инспекции/измерений (при условии «on line» обработки зрительных данных), и, как следствие, практическая непрерывность измерений в ходе движения;
- доступность разнообразных коммерческих аппаратных средств для сбора и обработки зрительных данных (экономическая целесообразность);
- скрытность/пассивность (без энергетического воздействия на окружающий мир);
- скрытность/пассивность (без энергетического воздействия на окружающий мир);
- хорошие перспективы унификации средств аппаратной поддержки (основная изменчивость СТЗ заключается в программно-алгоритмическом обеспечении).

Вместе с тем, эти положительные свойства выдвигают повышенные, относительно других сенсорных систем, требования к вычислительным ресурсам и математическому обеспечению. Современный уровень сенсоров и вычислительных средств позволяет находить компромисс между указанными преимуществами и требованиями и создавать эффективные СТЗ для решения практических задач. Приведём несколько известных, показательных примеров вооруженности СТЗ марсоходов, боевых роботов, беспилотных автомобилей.



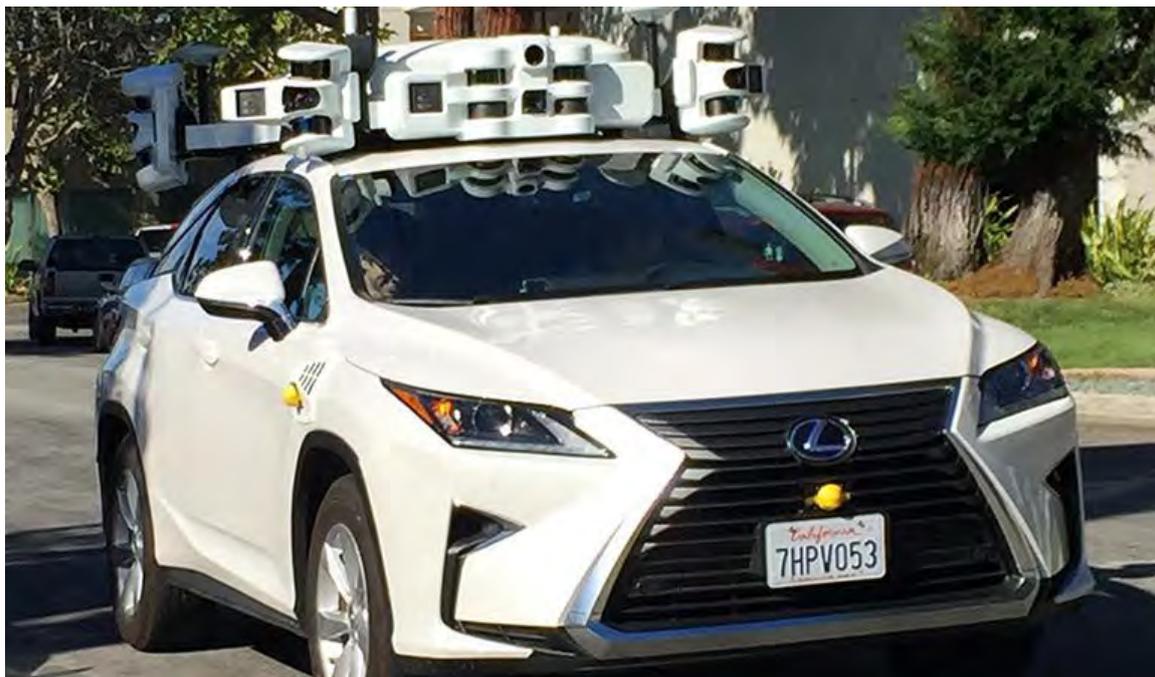
*Рис. 5.* Пример информационного оснащения мобильного робота. Обращает на себя внимание количество и разнообразие камер СТЗ в составе СИО марсохода 2020



*Рис. 6.* Прототип боевого беспилотного наземного робота Crasher. В составе СИО 18 видеокамер, включая 8 стереопар.



*Рис. 7.* Камеры в составе системы информационного обеспечения на беспилотном автомобиле Форд.



*Рис. 8. Камеры в составе системы информационного обеспечения Apple на беспилотном автомобиле.*

**Возрастание роли интеллектуальных методов обеспечения эффективности технических систем на всех уровнях (Проблема интеллектуализации в робототехнике).**

Среди методов обработки данных о среде функционирования РТК всё большее практическое применения находят интеллектуальные подходы, в частности, нейросетевые методы. По одной из используемых классификаций областей искусственного интеллекта [Юревич Е.И. Интеллектуальные роботы. / И.А. Каляев, В.М. Лохин и др. – М: Машиностроение, 13.2007. – 360 с.], рассматриваемые интеллектуальные системы управления (ИСУ) можно отнести к области так называемого «слабого интеллекта». Мы используем ещё один синоним этого термина – машинный интеллект. Машинный интеллект не претендует на замену человека в формулировках новых задач и создании новых объектов. Целью машинного интеллекта является повышение эффективности решения известных и формулируемых человеком задач для технических систем и освобождения человека от выполнения рутинных действий и работ в экстремальных условиях. Что касается базовых технологий обработки знаний, включая технологии экспертных систем, нейросетевых структур, нечеткой логики и ассоциативной памяти, то, как показывает анализ, их общим свойством является способность к решению задач классификации, которые возникают на каждом уровне иерархии интеллектуальных систем и лежат в основе ситуационного управления.

Указанные тенденции требуют решения проблемы унификации содержания и архитектуры алгоритмов управления действиями РТК. Коллеги

за рубежом ощутили эту необходимость ещё на рубеже 80-х – 90-х годов прошлого века, сформировав подход JTA и проект JAUS в его составе [16-19].

Большинством исследователей признаётся необходимость использования многоуровневой иерархической структуры построения систем управления робототехническими комплексами. Успешное решение задач управления РТК и другими сложными техническими комплексами возможно при построении эффективного управления на каждом из уровней и не менее эффективного их комплексирования.

Само деление на уровни и границы между ними не имеют чётко выраженных определений и в значительной степени зависит от конкретной реализации РТК. Приведём используемое нами выделение уровней в интеллектуальных системах управления. В качестве основного классифицирующего признака при выделении уровней управления используем свойства объектов, управление которыми реализуется на этом уровне: язык команд (управляющих воздействий).

Нижний уровень – уровень управления приводами исполнительных механизмов и выполнения базовых операций – элементарных движений, сбора информации о внешней среде.

Средний уровень – анализа информации о среде и условиях функционирования РТК, формирования последовательности и контроль исполнения базовых операций для решения поставленной верхним уровнем задачи (выбор пути движения, траектории движения исполнительных органов и тому подобные).

Верхний уровень – уровень принятия решения или помощи человеку в постановке задачи, целеполагании. В ближайшей перспективе наиболее важные функции этого уровня остаются за человеком – целеполагание, постановка новых задач. Потенциально, ряд исследователей полагает, что решение задач этого уровня может быть выполнено без участия человека с помощью «сильного» искусственного интеллекта.

Вне зависимости от границ уровней системы управления при построении РТК необходима интеграция всех уровней управления. Без единого механизма обмена данными и программирования всех уровней (кроссплатформенности) сложно ожидать создания эффективной робототехнической системы.

Зарубежный опыт показывает, что даже рыночная экономика приходит к необходимости вести разработки РТК по единому централизованному плану и унифицированным стандартам. В конце 80-х годов XX века заметно увеличились расходы МО США на проектирование новых систем контроля и управления. Они разрабатывались разными ведомствами, содержали много общих функциональных элементов, но реализовывались далеко не одинаково и, главное, не поддерживали единого стандарта взаимодействия. Для ликвидации этой проблемы с 1992 г. в научных армейских подразделениях Пентагона началось создание единой технической архитектуры Joint Technical Architecture (JTA) – стандарта на построение всех ИТ-решений. JTA состоит из нескольких архитектурных блоков. Функциональная архитектура включает

средства и методы описания рабочих элементов будущей системы и информационных потоков, направленных на поддержку действия боевых единиц. Она также задает тип хранимой или передаваемой информации, частоту обмена данными, описание реализуемых задач и т.д.

Техническая архитектура состоит из минимально необходимого набора правил, определяющих компоновку, взаимодействие и зависимости между всеми элементами системы. Она описывает доступные службы, интерфейсы и схемы взаимосвязей. В ЖТА также определяются различные физические параметры функционирования – виды и способы коммуникации, координатные системы, ключевые элементы (сети, узлы связи, платформы, настроечные параметры). Сначала ЖТА была успешно апробирована в системах электронной разведки, а затем также воплотилась в комплексах управления оружием, программах моделирования и другие. На основе ЖТА строится большинство прикладных инженерных стандартов и спецификаций Пентагона.

Одним из приложений ЖТА стал проект единой архитектуры робототехнических устройств Joint Architecture for Unmanned Systems (JAUS)\*. JAUS была задумана в 1994 г. в ходе экспериментов с различными способами дистанционного управления самоходными аппаратами. Уже тогда стала очевидной потребность в единых программных интерфейсах для подобных устройств – ведь главным определяющим фактором эффективности работа является в первую очередь программное обеспечение, однако на выработку программных спецификаций средств в то время не нашлось. Кроме того, скромные ресурсы бортовых систем накладывали дополнительные ограничения на поддержку сложной универсальной архитектуры.

В 1997 г. стартовала программа Пентагона по созданию универсальной автономной наземной машины Joint Robotics Program (JRP, [www.jointrobotics.com](http://www.jointrobotics.com)), а в начале 1998 г. в ее рамках была сформирована рабочая группа JAUS. Эффективность ее работы была подсчитана довольно быстро. Оказалось, что экономия на сопровождении жизненного цикла трех моделей роботов при внедрении JAUS составляет 200 млн. долл., прежде всего за счет повторного использования ПО.

Практически во всех автономных аппаратах применяются общие для них подсистемы, но до появления JAUS они были несовместимы друг с другом. А JAUS позволяет отказаться от привязки к конкретному поставщику решений, задействовать в каждой модели все ранее созданные средства как дистанционного управления, так и автономного поведения, ускорить и удешевить разработку роботов и быстро расширять их новыми функциональными возможностями на базе новых технических решений.

В России, в последние годы предпринимаются большие усилия для претворения в жизнь этих тенденций. В настоящее время формируется «нулевой» цикл, закладывается фундамент практической робототехники. Формируются стандарты, готовится правовое поле, испытываются отдельные образцы. Общие подходы к исследованию свойств и формированию требований к системам управления мобильных РТК с повышенной степенью

автономности и полностью автономных разрабатывались многими коллективами отечественных исследователей с 90-х годов прошлого века по настоящее время [20-24].

Мы постарались внести свой вклад в развитие перечисленных тенденций и исследовали/разработали составляющие во всех элементах интеллектуальных систем управления.

В нашем исследовании (с позиций «слабого» или машинного искусственного интеллекта) мы предполагаем, что исходная классификация выполняется людьми (указание свойств условий, объектов и процессов, по которым следует определять принадлежность объектов внешнего мира к тому или иному классу, сценарию действий). А в процессе функционирования системы это разбиение и принятие решений в соответствии с ним, принимается ИСУ.

Робототехника – относительно новое научно-техническое направление, терминология и понятия в этой области в отечественной практике только устанавливаются. Принимаются стандарты, в частности по терминологии, классификации, формируются справочники. Многие термины и понятия в русскоязычных публикациях используются без перевода или как прямой подстрочник. В этой связи мы считаем целесообразным привести используемые нами понятия, поэтому в книгу включен перечень используемых понятий, терминов и сокращений, а в начале каждого раздела есть составляющая, в которой указывается место описываемых исследований в соответствующей области робототехники.

Материалы книги по авторам распределены следующим образом: введение, частично раздел 4 и общая редакция – С.М. Соколов, раздел 1 – Г.К. Боровин, раздел 2 – В.А. Карташев, раздел 3 – В.Е. Павловский, раздел 4 – А.А. Богуславский и С.М. Соколов.

## **Литература**

1. Искусственный интеллект и интеллектуальные системы управления / И.М. Макаров, В.М. Лохин, С.В. Манько, М.П. Романов. – М.: Наука, 2006. – 333 с. – ISBN 5-02-033782-X.
2. Buehler M. et al. The 2005 DARPA Grand Challenge. Springer, 2007.
3. Buehler M. et al. The DARPA Urban Challenge. Springer, 2009.
4. Сайт проекта «самоуправляемый» автомобиль (self-driving car) режим доступа <https://www.google.com/selfdrivingcar>
5. Markoff J. (October 9, 2010). “Google Cars Drive Themselves in Traffic”. The New York Times. Retrieved October 11, 2010.
6. Сообщение агентства Defence Talk, режим доступа <https://lenta.ru/news/2006/01/26/oshkosh>
7. Ford первым тестирует автономные автомобили в условиях снега. Раздел «новости» официального сайта компании Ford Motor Company, режим доступа [http://www.ford.ru/AboutFord/News/Vehicle/2016/Vhcl\\_160113](http://www.ford.ru/AboutFord/News/Vehicle/2016/Vhcl_160113)

8. Кадаков М. Пересчитываем камеры самоуправляемого Nissan Leaf. За рулём.рф, режим доступа <http://www.zr.ru/content/articles/839794-derzhim-ruki-u-rulya-samoupravlyaemogo-nissan-leaf/>
9. Беспилотный КАМАЗ готов к испытаниям. Раздел «новости» официального сайта компании КАМАЗ, режим доступа [http://www.kamaz.ru/press/releases/bespilotnyu\\_kamaz\\_uzhe\\_na\\_doroge](http://www.kamaz.ru/press/releases/bespilotnyu_kamaz_uzhe_na_doroge)
10. Ализар А. Беспилотные «Камазы» выезжают на дороги общего назначения. Geektimes, режим доступа <https://geektimes.ru/post/264602/>
11. Ellery A. Rover vision – fundamentals / Planetary Rovers, Springer, 2016, pp.199-262.
12. США завершили палубные испытания беспилотника X-47В. Новостное агентство lenta.ru режим доступа <http://lenta.ru/news/2012/12/20/x47b/>
13. Юревич Е.И. Интеллектуальные роботы. / И.А. Каляев, В.М. Лохин и др. – М: Машиностроение, 13.2007. – 360 с.
14. Sokolov S.M., Boguslavsky A.A., Vasilyev A.I., Trifonov O. V. Methodical and practical aspects of system engineering of computer vision for navigation and measurements//News SFU. Engineering science, № 3, March 2012, ISSN 1999-9429, pp.76-85.
15. Sokolov S.M., Kirilchenko A.A. Information support of autonomous land vehicles on the basis of interpreting navigation, News SFU. Engineering science, №1, 2015, p. 120-133.

## 1. СИЛОВЫЕ СИСТЕМЫ РОБОТОВ И ИХ МАТЕМАТИЧЕСКИЕ МОДЕЛИ

Робототехническая система (РТС) состоит из системы управления (СУ), которая с помощью исполнительных устройств робота (двигателями) управляет объектом управления (ОУ). В качестве объектов управления может быть любой управляемый объект, технологический процесс, выделенный из окружающей среды по определенным признакам (конструктивным, функциональным и другим) и представляющий собой динамическую систему произвольной природы, изменяющую свое состояние под влиянием внешних воздействий. В зависимости от свойств и назначения объектов управления могут быть выделены различные технические, технологические и другие объекты и комплексы объектов. Объект управления выделяется из окружающей среды таким образом, чтобы на объект можно воздействовать и это воздействие должно изменять его состояние в заданном направлении.

Исполнительные устройства РТС состоят из механической системы и приводов. Механическая система РТС заканчивается рабочим органом в виде рабочего инструмента или захватного устройства, с помощью которого осуществляется целенаправленное воздействие робота на объекты внешней среды.

Обычно РТС включает в себя объект управления и исполнительные устройства, в качестве которых являются различные типы двигателей (электрические, гидравлические и др.), датчики обратной связи, корректирующие устройства. Интеллектуальная робототехническая система включает в себя объект управления совместно со средой, в которой она функционирует. Исполнительные двигатели, входящие в состав РТС, осуществляют перемещение ОУ по заданным законам. Информация о положении выходных звеньев РТС определяется датчиками, расположенными в шарнирах её звеньев, которые получают информации о выходных координатах механизмов перемещения, их скоростях, ускорениях и силах.

Задача управления заключается в том, чтобы ОУ в условиях реальных условиях эксплуатации обеспечивал бы выполнение требуемых функций. Фактическое состояние ОУ определяется одним или несколькими рабочими параметрами  $y(t)$ . Чаще всего рабочие параметры представляют собой физические величины: скорость (линейная и вращения), температура, напряжение электрического тока, линейные и угловые перемещения и т.д. В реальных условиях на ОУ оказывают влияние внешние воздействия, которые называются возмущающими  $z(t)$ . Эти воздействия вызывают изменение внутреннего состояния объекта и как следствие - рабочих параметров. В связи с этим, для выполнения требуемых рабочих функций по заданным алгоритмам необходимо на ОУ выдать управляющие воздействия  $U(t)$  (рис. 1.1).

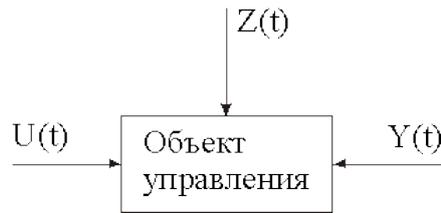


Рис. 1.1. Объект управления.

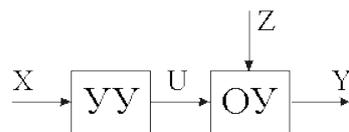
Основная функция системы управления РТС состоит в формировании законов перемещения её исполнительными механизмами в реальном времени. Обычно такие системы работают в следящем режиме, который должен обеспечивать выполнение каждой степенью подвижности РТС заданной траектории перемещения с требуемыми точностью, скоростью и усилием. Выходными координатами исполнительных механизмов РТС являются их положение и скорости в соответствующем пространстве. В результате взаимодействия исполнительного механизма с управляемым объектом создается усилие  $P(t)$ , которое воздействует на исполнительные органы РТС. В качестве объекта управления и внешней среды можно рассматривать перемещаемые изделия, какой-либо инструмент и технологический процесс.

Задача управления, по существу, заключается в формировании такого закона изменения управляющего воздействия, при котором обеспечивается заданный алгоритм при наличии возмущающих воздействий внешней среды.

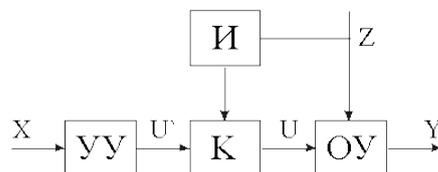
Для решения этой задачи используются три фундаментальных принципа управления: разомкнутое управление (а), управление по возмущению (б) (принцип компенсации) и замкнутое управление (в) (принцип обратной связи (ОС) или управление по отклонению).

Структурные схемы принципов управления представлены на рис. 1.2.

а)



б)



в)

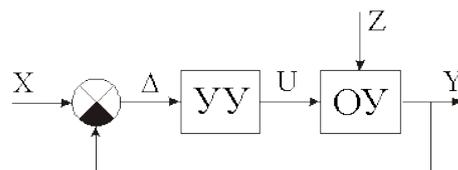


Рис. 1.2. Структурные схемы принципов управления (УУ – управляющее устройство; ОУ – объект управления; И – измеритель возмущения; К – корректирующее устройство)

Для решения задачи управления могут применяться все три принципа управления в зависимости от требуемой задачи. При управлении сложных задач с высокими требованиями по точности используется третий принцип с соответствующими надстройками. При этом для выработки управляющих воздействий на все степени подвижности решается краевая задача для обыкновенных дифференциальных уравнений (ОДУ), описывающих функционирование управляемых объектов.

Ниже будут рассмотрены математические модели различных электромеханических и электрогидравлических устройств, применяемых в системах управления робототехнических системах и особенности их моделирования в задачах проектирования и управления.

В настоящее время задачи управления многостепенными взаимосвязанными нелинейными электромеханическими и электрогидравлическими объектами, обеспечивающие функционирование сложных комплексных систем, занимают одно из первых мест по числу применений для высокотехнологичных установок в промышленности. К таким комплексам, как объектам управления, относятся робототехнические системы, конструкции экстремальных роботов-манипуляторов, наземные и морские подвижные объекты, беспилотные летательные аппараты и т.д. При этом достижение высокой точности и высокой производительности сложных электромеханических и электрогидравлических объектов может быть достигнуто методами и средствами более сложного управления, и в последнее время для решения научных и технических задач часто применяют аналитические и интеллектуальные адаптивные системы управления. Таким образом, задачи повышения динамической точности и быстродействия функционирования сложных механических объектов решаются созданием адекватных этим задачам эффективных систем управления их движением.

Для решения таких задач требуется развитие и совершенствование методов и средств автоматического управления многостепенными взаимосвязанными механическими объектами с априорно неопределенным и/или сложным нелинейным математическим описанием, с неполными измерениями, быстро и в широких пределах изменяющимися параметрами, свойствами и условиями функционирования, воздействиями внешней среды. В такой постановке одними из методов решения задач управления механическими объектами являются адаптивные методы.

Современный этап в проектировании сложных комплексов, управляющих высокотехнологичными объектами, связан с решением задач снижения влияния различных факторов, вызывающих нарушение рабочих режимов управляемых объектов. К таким факторам можно отнести отсутствие априорной информации о нестабильных значениях массо-инерционных и упругих параметров механических объектов, случайные изменения нагрузки; взаимовлияние степеней подвижности объектов; варьирование параметров объектов от образца к образцу и варьирование параметров стандартных систем регулирования при замене исполнительных приводов и отдельных блоков управления.

Построение большинства современных систем автоматического управления движением базируется на традиционной идеологии, так называемого подчиненного управления, основным вычислительным электронным модулем реализации которого является операционный усилитель. Однако такое построение систем с подчиненным управлением не учитывает проявление упомянутых не идеальностей динамики сложных многостепенных взаимосвязанных нелинейных упругих механических объектов с неопределенными параметрами и внешними возмущениями, которые могут иметь периодический и даже ударный характер.

Опыт подсказывает, что предельно возможное быстродействие широко распространенного в технике метода подчиненного управления при идеализированном представлении одной степени подвижности управляемого объекта в виде жестко присоединенной к исполнительному приводу нагрузки с неизменной инерционной характеристикой, соответствует полосе пропускания следящей системы до 15–40 Гц. Однако, при наличии упругих деформаций с частотами, лежащими в пределах 2–15 Гц реальное быстродействие систем может быть существенно снижено. Это может приводить к значительному недоиспользованию потенциальных возможностей современных исполнительных приводов. Очевидно при этом, что снижение быстродействия системы не решает проблему устойчивости к возникновению упругих колебаний, которые могут возбуждаться под действием ударных нагрузок. С другой стороны, вынужденное снижение быстродействия следящих систем с подчиненным управлением ухудшает реакцию систем на возмущения, которое может приводить к большим динамическим ошибкам этих систем в режимах стабилизированного наведения. Таким образом, современные приводы (электрические и гидравлические) создают необходимые предпосылки для совершенствования систем управления, исполнительными элементами которых они являются.

Другой необходимой предпосылкой создания более совершенных систем автоматического управления подвижными механическими объектами является идущий в настоящее время переход от аналоговой элементной базы электронных блоков бортовых систем управления к современной высокопроизводительной вычислительной микроконтроллерной технике бортового применения. А это уже вызывает необходимость корректировки традиционных методов построения систем управления подвижными объектами и перехода к более современным методам управления.

Мы считаем, что задача управления движением сложной механической системы (к ним можно отнести и робототехнические системы) формулируется как задача терминального управления с заданными начальными и конечными состояниями. По определению, терминальное управление – это управление, цель которого заключается в переводе объекта управления в заданное конечное состояние в заданный момент времени и при выполнении заданных критериев качества системы. Важной особенностью терминальных задач состоит в не единственности их решения. То есть жесткие требования по точности управления могут предъявляться только к конечной (терминальной)

точке, тем самым допуская для формирования промежуточной части траектории некоторый произвол. Поэтому в случаях, когда происходит отклонение объекта от номинальной траектории, нет необходимости в стабилизации объекта управления на исходной номинальной траектории, а в каждой текущей точке возможно построение траектории движения объекта управления, которая должна удовлетворять тем же терминальным условиям.

Наиболее адекватным управлением применительно к задачам терминального управления является управление с обратной связью. Под управлением с обратной связью понимается управление, при котором текущие управляющие воздействия вырабатываются с учетом состояния объекта управления. Однако, кроме учета текущего состояния объекта управления, закон управления должен обеспечивать выполнение терминальных условий и допускаяемое отличие новых траекторий к исходной траектории, соответствующей номинальным начальным условиям.

### **1.1. Управление с введением прогнозирующей модели**

Анализ существующих работ, связанных с алгоритмами управления, позволяет сделать вывод, что терминальные системы управления отличаются от других систем, в первую очередь, способом организации процесса управления. В этих случаях предусматривается прогнозирование будущего движения системы от текущего до терминального момента времени и формирование процесса изменения управляющего воздействия (программы управления), приводящего систему в заданное конечное состояние. Кроме того, задачи терминального управления часто являются многокритериальными, поскольку требование выполнения терминального состояния с заданной точностью в заданный момент времени может дополняться другими критериями.

Синтез систем терминального управления осуществляется в два этапа. На первом этапе из условия экстремума (наилучших значений) критерия (критериев) находится класс функций, определяющих алгоритм управления. На втором этапе решается задача синтеза системы управления с обратной связью, которое должно обеспечивать требуемое качество терминального управления при возмущенном движении системы, а также при возможных изменениях параметров объекта. Основой для нахождения терминального управления является система обыкновенных дифференциальных уравнений, описывающая объект управления, ограничения на фазовые координаты и функции от них, оптимизирующий функционал и граничные условия на левом (при  $t = 0$ ) и правом (при  $t = T$ ) концах фазовой траектории. Результатом решения является вектор-функция управления  $U(u_1, u_2, \dots, u_n)$ , которая должна минимизировать на отрезке времени  $[0, T]$  заданный функционал и обеспечивать выполнение наложенных ограничений и соблюдение граничных условий.

Однако, алгоритм и программа, осуществляющая поиск оптимальных управлений, требуют затрат машинного времени, которые затрудняют ее

использование в режиме реального времени. Поэтому при разработке алгоритмов и программ для синтеза законов управления, обеспечивающих оптимизацию заданных критериев качества следует иметь это в виду. Они должны быть максимально простыми, но в тоже время адекватными для управления с необходимой точностью.

Рассмотрим, с нашей точки зрения, одно из перспективных направлений повышения точности процесса управления в классе систем автоматического управления (САУ) введением параллельной прогнозирующей модели основному контуру. Рассматриваемые технологические объекты управления – промышленные роботы, у которых датчики обратных связей установлены в «суставах плеч», транспортные машины, на которых установлены системы слежения за подвижными объектами или иное специальное навесное оборудование, регулируемая координата: продольная ось антенны локатора, ось ствола орудия или направляющей пусковой установки или навесного специального оборудования, в то время как датчики обратной связи установлены на координатных приводах. Такое управление может быть использовано также для подвижных транспортных средств в автоматическом (дистанционном) режиме в условиях переменных параметров рельефа местности и трассы движения.

Однако, несмотря на широкое применение прогнозирующих систем в автоматических и полуавтоматических системах управления, в САУ с элементами неопределенности внешней среды и состояния объекта управления, это направление требует своего дальнейшего развития.

Алгоритм работы такой системы управления предлагается следующим:

- 1) Имеется некоторая математическая модель привода, управляющая каким-либо объектом, начальным условием для которой служит его текущее состояние. Математическая модель – система ОДУ, описывающая процессы, протекающие в приводе. При заданном программном управлении выполняется решение ОДУ этой модели, что дает прогноз движения объекта на некотором конечном отрезке времени.
- 2) Выполняется оптимизация программного управления, целью которого служит приближение регулируемых переменных прогнозирующей модели к соответствующим задающим сигналам прогноза.
- 3) На шаге вычислений, составляющем фиксированную малую часть прогноза, реализуется найденное оптимальное управление.
- 4) Прогноз сдвигается на шаг вперед и повторяются пп. 1–3 данной последовательности действий.

Принципиальными вопросами, подлежащими исследованию, являются: выбор прогнозирующих функций, или полиномов, и анализ точности их работы в реальном масштабе времени; алгоритмизация прогнозирующих функций и анализ особенностей их программной реализации; выбор и назначение значения времени прогноза; алгоритмы формирования дополнительной составляющей в управляющем воздействии; анализ устойчивости алгоритмов адаптации.

## **1.2. Математическое моделирование элементов системы управления робототехнических комплексов**

Дифференциальные уравнения движения сложного механического объекта, например, робота-манипулятора (вместе с уравнениями, описывающими его систему управления, являются, как правило, существенно нелинейными и часто имеют разрывные правые части. При необходимости получения адекватных, а не оценочных результатов, в математических моделях системы управления нужно учитывать многие из имеющихся в них нелинейностей. Это характерно при описании динамики многих механических систем.

При моделировании систем управления (СУ) сложных механических объектов, математические модели которых включают модели электромеханических или гидравлических приводов [21, 23], могут возникнуть трудности, связанные с особенностями таких систем. В таких случаях важными являются: выбор эффективных численных методов и вопросы, связанные с погрешностью вычислений.

Для исследования и проектирования СУ обычно проводятся расчеты анализа статического и динамического режимов, частотный анализ, вычисление чувствительности характеристик СУ и/или ее элементов к изменению значений конструктивных параметров, структурная и параметрическая оптимизация и т.д. Расчет динамических режимов СУ и/или ее элементов заключается в решении задач анализа переходных процессов и частотных характеристик, а также исследование и анализ процессов, протекающих в их элементах. Во всех этих случаях решение задачи сводится к решению, как правило, нелинейных систем обыкновенных дифференциальных уравнений (ОДУ), которые могут иметь высокий порядок, при различных типовых и не типовых входных управляющих воздействиях. Для решения этих задач используются численные методы решения систем ОДУ. Эти задачи обычно формулируются в виде задачи Коши для системы ОДУ в нормальной форме Коши:

$$\begin{aligned} \frac{dX}{dt} &= f(X, t) \\ X(t_0) &= X_0 \end{aligned}$$

где  $X$  – вектор зависимых переменных,  $t$  – время,  $X_0$  – начальные условия.

Ниже будут рассмотрены некоторые аспекты выбора и применения численных методов интегрирования ОДУ, описывающих электромеханические и гидравлические системы управления.

Существует большое многообразие схем следящих электро- и гидроприводов. Здесь мы ограничимся некоторыми из них, которые можно считать традиционными.

### 1.2.1. Электромеханические системы управления

В системах следящего электропривода с высокими требованиями к статическим и динамическим характеристикам широко применяется электропривод постоянного тока.

Под электроприводом понимают электромеханическую систему, состоящую из электродвигателя, преобразователя, передаточного и управляющего устройств и предназначенную для приведения в движение и управления движением исполнительных органов рабочей машины. Наиболее совершенным видом электропривода является следящий электропривод, отрабатывающий перемещения исполнительного органа рабочей машины в соответствии с произвольно изменяющимся входным сигналом. Развитие электропривода в настоящее время характеризуется широким внедрением силовых тиристорных и транзисторных преобразователей и унифицированных систем управления. При этом получили широкое распространение системы подчиненного регулирования, характеризующиеся типовой структурой, высоким качеством регулирования и удобством практического использования. Принцип работы этих схем заключается в подчинении работы каждого внутреннего контура многоконтурной системы внешнему контуру. Очевидно, что качество работы всей системы во многом определяется качеством внутренних контуров. Традиционные системы управления электродвигателями постоянного тока состоят из контуров регулирования положения, скорости и тока. На рис. 1.3 показана функциональная схема следящего электропривода, в которой имеются контур положения /РП/ с датчиком положения /ДП/, контур регулирования скорости с регулятором скорости /РС/ и с датчиком скорости /ДС/, контур регулирования тока /РТ/ и с датчиком тока /ДТ/. В качестве регуляторов тока и скорости может быть любой из П, ПИ, ПИД - регуляторов, в зависимости от требуемого качества регулирования. Выходной сигнал с РС поступает на вход системы управления тиристорного, либо транзисторного преобразователя /ТП/, сигнал с которого подается на обмотку якоря электродвигателя /ЭД/. На вход в РС поступает сигнал с выхода РП. Механическая часть электропривода включает в себя рабочий орган, редуктор и соединительные муфты и в общем случае может рассматриваться как многомассовая система с люфтом и упругими связями.

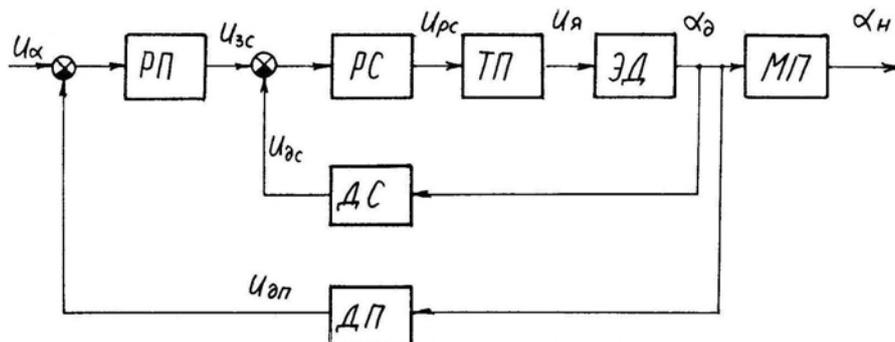


Рис. 1.3. Функциональная схема следящего привода.

На рис. 1.4 приведена структурная схема электропривода постоянного тока и графический вид функций, используемых в схеме. Здесь приняты следующие обозначения:  $K_y$  – коэффициент усиления операционного усилителя;  $F_1$  – нелинейность, описывающая ограничение сигнала в операционном усилителе;  $K_{pc}$ ,  $T_{pc1}$ ,  $T_{pc2}$ ,  $T_{pc3}$  – коэффициент усиления и постоянные времени регулятора скорости;  $K_{pt1}$ ,  $K_{pt2}$ ,  $T_{pt1}$ ,  $T_{pt2}$ ,  $T_{pt3}$  – коэффициент передачи и постоянные времени регулятора тока;  $R$ ,  $L$  – активное сопротивление и индуктивность якорной цепи двигателя постоянного тока;  $K_m$  – коэффициент передачи по моменту двигателя;  $K$  – коэффициент противоэдс;  $J$  – момент инерции двигателя;  $K_{дт}$ ,  $T_{дт}$  – коэффициент передачи и постоянная времени датчика скорости;  $F_2$  – нелинейность, описывающая выходной сигнал регулятора скорости;  $F_3$  – нелинейность, описывающая зону нечувствительности в регуляторе тока;  $F_4$ ,  $F_5$  – нелинейности, описывающие устройство широтно-импульсной модуляции;  $F_6$  – нелинейность типа «сухого трения».

Математическая модель следящего электропривода описана дифференциальными и разностными уравнениями. Она представлена в виде отдельных моделей, каждая из которых описывает выделенный элемент привода. Такими элементами являются: РП, РС, РТ, механическая передача МП, датчики обратных связей, используемые в приводе, преобразователи и управляющие устройства. Дифференциальные уравнения, описывающие процессы в электродвигателе постоянного тока, учитывают особенности, как якорной цепи электродвигателя, так и механической системы. Они приведены ниже. Разностные уравнения описывают динамику цифрового РП.

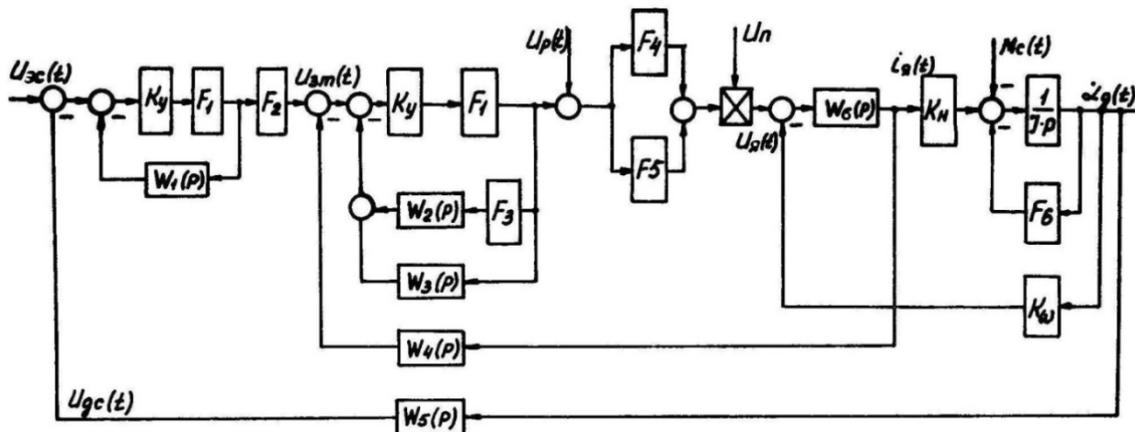


Рис. 1.4(а). Структурная схема электропривода постоянного тока.

Передаточные функции, указанные на схеме, имеют вид (графики показаны на рис. 1.4(б)):

$$W_1(p) = \frac{1 + T_{pc2} \cdot p}{K_{pc}(1 + T_{pc1} p)}$$

$$W_4 = \frac{K_{gm}}{1 + T_{gm} p}$$

$$W_2(p) = \frac{1 + T_{pT3} \cdot p}{K_{pT2}(1 + T_{pT2} p)}$$

$$W_5 = \frac{K_{gc}}{1 + T_{gc} \cdot p}$$

$$W_3(p) = \frac{1 + T_p T_1 p}{K_{pT1}}$$

$$W_6 = \frac{1}{R + L \cdot p}$$

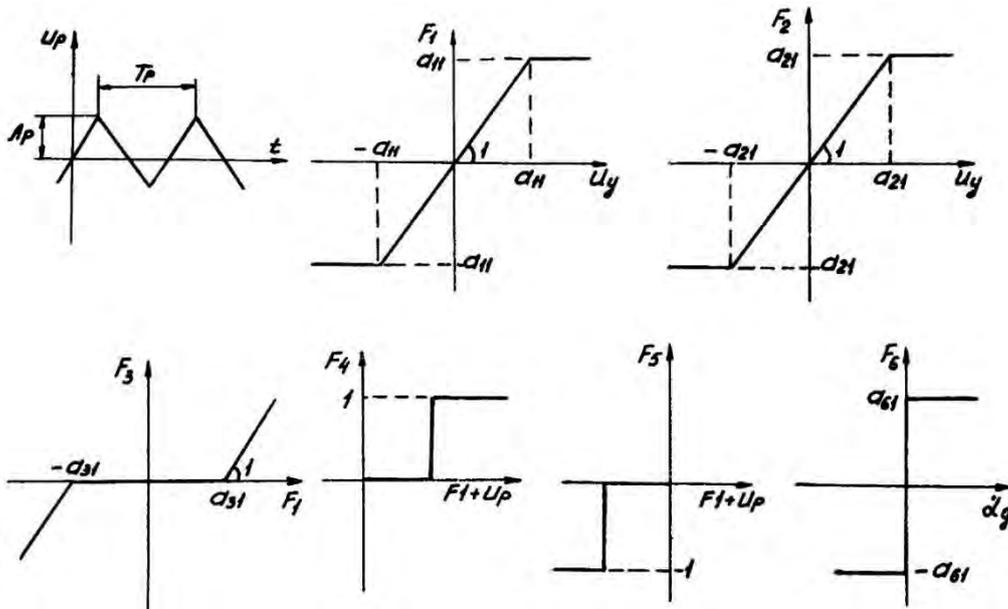


Рис. 1.4(б). Функции, используемые в схеме на рис. 1.4(а).

Асинхронный электропривод, является одним из основных компонентов современных станков с ЧПУ, промышленных роботов и гибких производственных модулей, во многом определяет их производительность, точность, надежность и экономические показатели. Наибольшая эффективность автоматических электроприводов в таких системах обеспечивается при применении электродвигателей, обладающих минимальной стоимостью, максимальной надежностью при малых массогабаритных показателях. При условии обеспечения требуемых показателей точности, быстродействия, качества регулирования параметров электропривода в наибольшей степени этому направлению отвечает электропривод с векторным управлением на базе асинхронного электродвигателя с короткозамкнутым ротором [4, 5, 35, 36]. При этом под векторным управлением понимается такое направленное изменение фазы, частоты и амплитуды переменного тока в статорных обмотках, при котором формируется требуемый закон управления фазового сдвига между векторами тока статор и магнитного потока, в результате чего обеспечивается воспроизведение требуемого момента электродвигателя. В отличие от других типов электродвигателей асинхронных электродвигатель может развивать при одной и той же величине тока статора различный момент двигателя в зависимости от того, как изменяется угол фазового сдвига. Этим можно объяснить большое количество вариантов структур электродвигателей и

реализуемых в них алгоритмов управления. Сравнение этих вариантов и анализ эффективности оптимальных законов векторного управления затруднены сложностью математической модели системы «Транзисторный преобразователь частоты-двигатель». Поэтому из основных задач математического моделирования системы векторного управления является анализ структуры и алгоритма векторного управления с точки зрения точности быстрогодействия воспроизведения момента и скорости асинхронного двигателя при достижении наилучших энергетических соотношений величин момента, тока, потерь и нагрева в асинхронном двигателе.

Другой группой задач является создание структуры, нахождение алгоритмов и анализ эффективности оптимального векторного управления по критерию максимума отношения момента к величине тока статора, преобразующих входные сигналы задания углового положения ротора, скорости или момента двигателя в сигналы управления транзисторным, инвертором, питающим асинхронный двигатель. Способы движения этого критерия оптимальности в совокупности как статистических, так и динамических режимов асинхронного двигателя, предложенные в работах [4, 5], существенно повышает эффективность исследования асинхронных электроприводов в станкостроении и робототехнике с точки зрения повышения удельных показателей по моменту и мощности, скорость, предельного ускорения, точности.

Комплексное моделирование на ЭВМ системы оптимального векторного управления, транзисторного инвертора и асинхронного электродвигателя обеспечивает решение задач анализа динамических и энергетических показателей электропривода, что позволяет создать САПР асинхронных электроприводов с векторным управлением и одновременно создавать программную основу для микропроцессорного управления.

### **Структурная схема следящего асинхронного электропривода с векторным управлением**

Структурная схема следящего асинхронного электропривода с векторным управлением приведена на рис. 1.5. Внешний контур регулирования положения с обратной связью по положению  $\varphi$  имеет регулятор положения с передаточной  $W_{pn}(p)$ . Ему подчинен контур регулирования скорости с регулятором  $W_{pc}(p)$  и контур регулирования момента  $W_{pm}(p)$ . Для асинхронного электропривода регулятор момента  $W_{pm}(p)$  выполняется во вращающейся системе координат и затем с введением положительной обратной связи по скорости  $\omega$ , сигналы заданного тока статора преобразуются преобразователем координат в выходные сигналы задания вектора тока в системе координат статора. Внутренний контур регулирования вектора тока статора осуществляется введением отрицательных обратных связей по мгновенным фазным токам, трехфазного регулятора тока  $W_{pt}(p)$ , устройства широтноимпульсной модуляции и быстрогодействующего силового импульсного инвертора  $W_{ин}(p)$  [4, 5, 35].

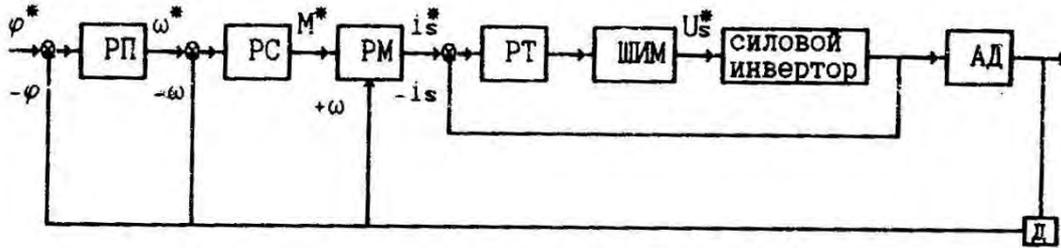


Рис. 1.5. Структурная схема асинхронного электропривода с векторным управлением (Обозначения на схеме: РП – регулятор положения; РС – регулятор скорости; РМ – регулятор момента; РТ – регулятор тока; АД – асинхронный двигатель; Д – датчик положения; ШИМ – устройство формирования ШИМа).

На рис. 1.6 приведена функциональная схема регулируемого по скорости электропривода с асинхронным электродвигателем.

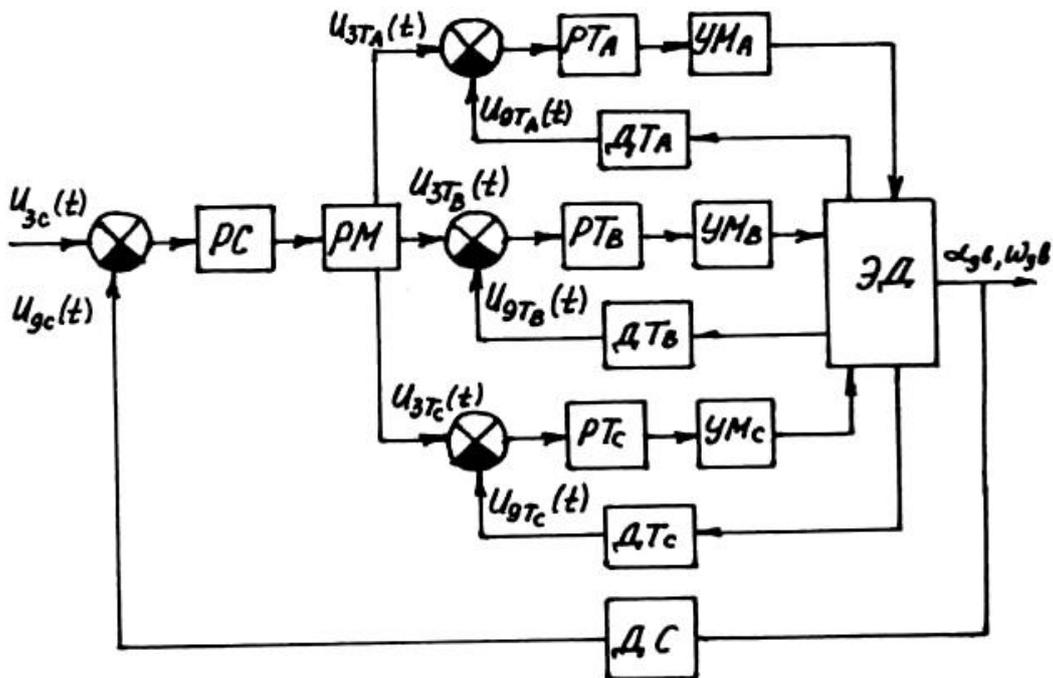


Рис. 1.6. Функциональная схема регулируемого по скорости электропривода с асинхронным электродвигателем.

Структурной и функциональной схемам следящего асинхронного электропривода с векторным управлением соответствует приведенная ниже математическая модель асинхронного электродвигателя.

Теоретической основой для построения математической модели асинхронного электродвигателя (АД), как и любого электромеханического преобразователя энергии, являются дифференциальные уравнения электрического и механического равновесия и уравнения преобразования электромагнитной энергии в механическую [30].

На основании общей теории электромеханических преобразователей, в которой любая многофазная электрическая машина с  $n$ -фазной обмоткой статора и  $m$ -фазной обмоткой ротора может быть представлена эквивалентной двухфазной машиной [30], получается достаточно известная математическая модель асинхронного электродвигателя для любой системы координат  $\alpha$ ,  $\beta$ , вращающейся с произвольной скоростью относительно оси статора «а» [30, 35, 36].

$$\begin{aligned}
 U_{s\alpha} &= \frac{d\psi_{s\alpha}}{dt} - \omega_k \cdot \psi_{s\beta} + R_s \cdot i_{s\alpha} \\
 U_{s\beta} &= \frac{d\psi_{s\beta}}{dt} - \omega_k \cdot \psi_{s\alpha} + R_s \cdot i_{s\beta} \\
 0 &= \frac{d\psi_{r\alpha}}{dt} - (\omega_k - \omega_3) \cdot \psi_{r\beta} + R_r \cdot i_{r\alpha} \\
 0 &= \frac{d\psi_{r\beta}}{dt} - (\omega_k - \omega_3) \cdot \psi_{r\alpha} + R_r \cdot i_{r\beta} = \\
 M &= \frac{3}{2} \cdot Z_p \cdot (\psi_{s\alpha} \cdot i_{s\beta} - \psi_{s\beta} \cdot i_{s\alpha}) = \\
 &= \frac{3}{2} \cdot Z_p \cdot (\psi_{r\alpha} \cdot i_{r\beta} - \psi_{r\beta} \cdot i_{r\alpha}) = \\
 &= \frac{3}{2} \cdot Z_p \cdot \frac{L_m}{L_r} (\psi_{r\alpha} \cdot i_{s\beta} - \psi_{r\beta} \cdot i_{s\alpha}) \\
 M - M_c &= \mathbf{I} \cdot \frac{d\omega_m}{dt}
 \end{aligned} \tag{1.1}$$

где  $U_{s\alpha}$ ,  $U_{s\beta}$  – составляющие напряжения обмотки статора по двум осям  $\alpha$ ,  $\beta$ ;  $\psi_{s\alpha}$ ,  $\psi_{s\beta}$  – составляющие потокосцепления статора по двум осям  $\alpha$ ,  $\beta$ ;  $\psi_{r\alpha}$ ,  $\psi_{r\beta}$  – составляющие потокосцепления ротора по двум осям  $\alpha$ ,  $\beta$ ;  $i_{s\alpha}$ ,  $i_{s\beta}$  – составляющая тока статора по двум осям  $\alpha$ ,  $\beta$ ;  $i_{r\alpha}$ ,  $i_{r\beta}$  – составляющая тока ротора по двум осям  $\alpha$ ,  $\beta$ ;  $R_s$ ,  $R_r$  – соответственно активные сопротивления статора и ротора;  $L_m$ ,  $L_r$  – соответственно взаимная и индуктивная ротора;  $\omega_3$  – электрическая частота вращения ротора;  $\omega_m$  – механическая частота вращения ротора;  $\omega_3 = Z_p \cdot \omega_m$ ;  $Z_p$  – число пар полюсов;  $\mathbf{I}$  – момент инерции ротора;  $\omega_k$  – скорость вращения системы координат  $\alpha$ ,  $\beta$ ;  $M$  – электромагнитный момент двигателя;  $M_c$  – момент сопротивления.

Для потокосцепления и токов имеют место следующие соотношения:

$$\begin{aligned}
 \psi_{s\alpha} &= L'_s \cdot i_{s\alpha} + \frac{L_m}{L_r} \cdot \psi_{r\alpha} \\
 \psi_{s\beta} &= L'_s \cdot i_{s\beta} + \frac{L_m}{L_r} \cdot \psi_{r\beta} \\
 \psi_{r\alpha} &= \psi_{m\alpha} + L_{\sigma r} \cdot i_{r\alpha} = L_m \cdot (i_{s\alpha} + i_{r\alpha}) + L_{\sigma r} \cdot i_{r\alpha} \\
 \psi_{r\beta} &= \psi_{m\beta} + L_{\sigma r} \cdot i_{r\beta} = L_m \cdot (i_{s\beta} + i_{r\beta}) + L_{\sigma r} \cdot i_{r\beta}
 \end{aligned}$$

$$L'_s = L_s - \frac{L_m^2}{L_r} = L_{\sigma r} + \frac{L_m}{L_r}$$

$$L_s = L_{mH} + L_{\sigma s}.$$

Здесь  $L_{\sigma r}$  – индуктивность рассеяния ротора;  $L_{\sigma s}$  – индуктивность рассеяния статора;  $L_{mH}$  – номинальная взаимная индуктивность. Векторная диаграмма асинхронного электродвигателя приведена на рис. 1.7.

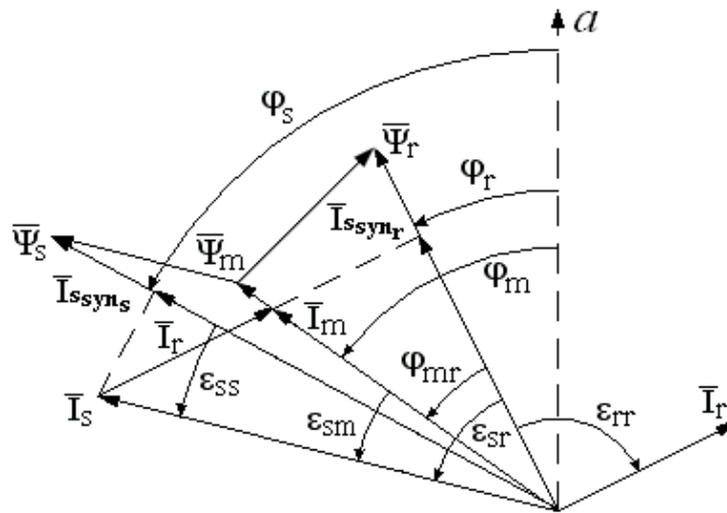


Рис. 1.7. Векторная диаграмма асинхронного двигателя.

При моделировании процессов управления АД важен выбор систем координат, который для конкретной записи системы уравнений может оказывать влияние на время, затраченное ЭВМ на решение, точность получаемых результатов и сложность алгоритмов и программ моделирования. Кроме того, одним из основных вопросов при анализе электромагнитных переходных процессов и синтезе систем управления асинхронным электродвигателем является обоснованный выбор и корректное математическое представление управляющих переменных асинхронного электродвигателя, образующих многомерное фазовое пространство области управления асинхронным электроприводом. Трехфазный асинхронный электродвигатель с короткозамкнутым ротором может управляться изменением параметров электрической энергии только со стороны статорной цепи. Параметры электрической энергии характеризуется мгновенными значениями напряжений и токов статорных обмоток электродвигателя и изменяются во времени с помощью преобразовательного устройства и процессов в системе «преобразователь – двигатель», где преобразователь может рассматриваться либо как источник напряжения, либо как источник тока. С учетом всего этого система уравнений (1.1), описывающая поведения АД, а также при совмещении оси  $\alpha$  системы координат  $\alpha, \beta$  с неподвижной осью фазы «а» трехфазной симметричной стартовой обмотки асинхронного двигателя ( $\omega_k = 0$ ) и выбрав качестве управляющего вектора – вектора тока статора, будет иметь следующий вид:

$$\begin{aligned}
U_{s\alpha} &= \frac{d\psi_{s\alpha}}{dt} - \omega_k \cdot \psi_{s\beta} + R_s \cdot i_{s\alpha} \\
U_{s\beta} &= \frac{d\psi_{s\beta}}{dt} - \omega_k \cdot \psi_{s\alpha} + R_s \cdot i_{s\beta} \\
0 &= \frac{d\psi_{r\alpha}}{dt} - (\omega_k - \omega_3) \cdot \psi_{r\beta} + R_r \cdot i_{r\alpha} \\
0 &= \frac{d\psi_{r\beta}}{dt} - (\omega_k - \omega_3) \cdot \psi_{r\alpha} + R_r \cdot i_{r\beta} \\
M &= \frac{3}{2} \cdot Z_p \cdot (\psi_{s\alpha} \cdot i_{s\beta} - \psi_{s\beta} \cdot i_{s\alpha}) = \\
&= \frac{3}{2} \cdot Z_p \cdot (\psi_{r\alpha} \cdot i_{r\beta} - \psi_{r\beta} \cdot i_{r\alpha}) = \\
&= \frac{3}{2} \cdot Z_p \cdot \frac{L_m}{L_r} (\psi_{r\alpha} \cdot i_{s\beta} - \psi_{r\beta} \cdot i_{s\alpha}) \\
M - M_c &= \mathbf{I} \cdot \frac{d\omega_m}{dt}
\end{aligned} \tag{1.2}$$

При этом входными параметрами для математической модели асинхронного двигателя (1.2) являются напряжения с выхода силового инвертора  $U_{sa}$ ,  $U_{sb}$ ,  $U_{sc}$ , преобразованные в напряжения обмотки статора двухфазной машины в неподвижной декартовой системе координат

$$\begin{aligned}
U_{s\alpha} &= \frac{2}{3} \left( U_{sa} - \frac{U_{sb} + U_{sc}}{2} \right) \\
U_{s\beta} &= \frac{U_{sb} + U_{sc}}{\sqrt{3}}
\end{aligned} \tag{1.3}$$

Для получения математической модели силового инвертора необходимо выразить фазные напряжения обмотки статора  $U_{sa}$ ,  $U_{sb}$ ,  $U_{sc}$ , асинхронного электродвигателя в зависимости от состояния ключей силового инвертора, т.е. необходимо матрицу состояний инвертора связать с матрицей фазных напряжений таким образом, чтобы каждому сочетанию ключей инвертора (любому значению матрицы состояния инвертора) соответствовало единственное значение фазных напряжений  $U_{sa}$ ,  $U_{sb}$ ,  $U_{sc}$  [30, 35, 36].

Рассмотрим возможные реализации данного способа формирования ШИМа. Они состоят в представлении опорного треугольного гармонического сигнала. Опорный сигнал можно представить двумя способами:

- как периодическую кусочно-линейную функцию с амплитудой  $A$  и периодом  $T = 1/f$ ;
- виде зависимости, полученной как результат разложения треугольного гармонического сигнала в ряд Фурье.

Такую зависимость можно записать в виде:

$$Y_{\text{оп}} = \frac{2T}{n^2} \left( -\text{Sin } \omega t + \frac{1}{9} \cdot \text{Sin} 3\omega t - \frac{1}{25} \cdot \text{Sin } 5\omega t - \dots \right), \quad (1.4)$$

где  $A$  – амплитуда опорного сигнала,  $\omega = 2\pi f$  – круговая частота. Алгоритм определения точек пересечения кривых, соответствующих опорному и управляющему сигналам для формирования ШИМа в зависимости от вида представления опорного сигнала могут быть следующими:

- 1) Если опорный сигнал представлен в виде аналитической зависимости, то точки пересечения могут быть определены, при решении нелинейного алгоритмического уравнения  $f = y - U = 0$ , в котором  $y$  – функция опорного сигнала, а  $U$  – функция, описывающая управляющий сигнал. Приведенные уравнения удобно решать с помощью метода Ньютона, для которого в качестве нулевого приближения брать пересечения опорного сигнала с осью абсцисс. В таком случае имеем сходимость за 2-3 итерации.
- 2) Если опорный сигнал представлен в виде периодической кусочно-линейной зависимости, то точки пересечения можно определить как пересечения сигнала  $U$  с отрезками прямой треугольного гармонического сигнала. В этом случае в вычислениях преобладают логические операции.

Контур регулирования вектора тока статора представляет собой многосвязную нелинейную систему автоматического векторного регулирования асинхронного электродвигателя, содержащую трехфазный транзисторный силовой инвертор, питаем постоянным напряжением, трехфазный асинхронный электродвигатель с короткозамкнутым ротором и трехфазную цепь обратной связи по мгновенным выходным токам инвертора.

Входными величинами контура тока для указанного выше случая являются величины ортогональных составляющих тока статора  $I_{syn}^*$ ,  $I_{ort}^*$ , (рис. 1.7) и величина скольжения  $\Delta\omega_s$ , поступающих из контура регулирования момента.

Формирование составляющих тока статора  $i_{s\alpha}^*$ ,  $i_{s\beta}^*$  в системе координат потокосцепления ротора  $I_{syn}^*$ ,  $I_{ort}^*$  проводится по следующим формулам:

$$\begin{aligned} i_{s\alpha}^* &= I_{syn}^* \cdot \text{Cos} \varphi_s - I_{ort}^* \text{Sin} \varphi_s \\ i_{s\beta}^* &= I_{syn}^* \cdot \text{Sin} \varphi_s - I_{ort}^* \text{Cos} \varphi_s, \end{aligned} \quad (1.5)$$

где  $\varphi_s$  – фаза синхронизации;  $\varphi_s = \int \omega_s dt$ ;  $\omega_s$  – частота синхронизации,  $\omega_s = \omega_3 + \Delta\omega_s$ ;  $\omega_3$  – реальная скорость ротора, полученная с помощью датчика обратной связи по скорости. Вычисление заданных мгновенных фазных токов двигателя производится по следующим формулам:

$$\begin{aligned}
 i_{sa}^* &= i \\
 i &= \frac{\sqrt{3}}{2} i_{s\beta}^* - \frac{1}{2} i_{s\alpha}^* \\
 i_{sc}^* &= -i_{sa}^* - i
 \end{aligned} \tag{1.6}$$

Величины мгновенных заданных значений токов в фазах двигателя  $i_{sa}^*, i_{sb}^*, i_{sc}^*$  поступают на входы пропорциональных регуляторов тока. На вторые входы поступают величины измеренных мгновенных фазных токов  $U_{sa}, U_{sb}, U_{sc}$ . Происходит сравнение заданных и реальных величин

$$\begin{aligned}
 \Delta i_{sa} &= i_{sa}^* - i_{sa} \\
 \Delta i_{sb} &= i_{sb}^* - i_{sb} \\
 \Delta i_{sc} &= i_{sc}^* - i_{sc}
 \end{aligned}$$

Величины задания фазных напряжений двигателя, подаваемых на устройства ШИМа для отработки, находятся по уравнениям:

$$\begin{aligned}
 U_{sa}^* &= K_{pT} \cdot \Delta i_{sa} \\
 U_{sb}^* &= K_{pT} \cdot \Delta i_{sb} \\
 U_{sc}^* &= K_{pT} \cdot \Delta i_{sc}
 \end{aligned}$$

Входным сигналом, поступающим на контур регулирования момента, является величина заданного момента  $M^*$  – сигнал с выхода контура регулирования скорости.

Рассмотрим два закона регулирования момента:

### 1) Управление с постоянством угла $\varepsilon_{\varepsilon} = \pm 45^\circ$ ( $I_{syn}^* = I_{ort}^*$ )

Математическое описание контура момента следующее [4, 35]:

$$\begin{aligned}
 I_{syn}^* &= I_{ort}^* \\
 I_{ort}^* &= \frac{2}{3 \cdot Z_p} \frac{L_r}{L_m} \frac{M^*}{\psi_r^*(t)} \\
 \Delta \omega_s &= \frac{R_r}{L_r} L_m \frac{I_{ort}^*}{\psi_r^*(t)}
 \end{aligned} \tag{1.7}$$

В начальный момент времени до подачи сигнала на двигатель, формируется сигнал задания начального момента  $\psi_{r0}^* = K \psi_{rH}$ , где  $K = 0.5, \dots, 1.35$ . При этом  $I_{syn0}^* = \frac{\psi_{r0}^*}{L_{m0}}$ , где  $L_m$  – индуктивность намагничивающего контура, соответствующая  $\psi_{r0}^*$ . При достижении потокосцепления ротора  $\psi_r(t)$  значения  $\psi_{r0}^*$  включается сигнал на работу двигателя в соответствии с алгоритмом управления (1.7). Для вычисления потокосцепления на текущем такте используется формула:

$$\psi_r^*(t) = \exp\left(-\Delta t/T_r\right) \cdot \psi_r^* \cdot (t - \Delta t) + (1 - \exp\left(-\Delta t/T_r\right))\psi_{r\infty}^*$$

Здесь  $T_r = \frac{R_r}{L_r}$  – постоянная ротора,  $\Delta t$  – такт управления.

Для  $t \neq 0$  и  $\psi_{r\infty} = const$  получается известная формула

$$\psi_r^*(t) = \psi_{r\infty}^* \cdot \left(1 - \exp\left(-t/T_r\right)\right).$$

Необходимо также учитывать ограничения по максимально допустимому току  $I_{ort}^* \leq i_{smax}^*$ .

## 2) Управление с постоянством потокосцепления ротора

Математическое описание контура момента следующее [4, 5, 30, 35, 36].

$$\psi_{r0}^* = K\psi_r^*, \text{ где } K = 0.5, \dots, 1.35.$$

$$I_{syn}^* = \frac{\psi_{r0}^*}{L_{m0}}$$

$$I_{ort}^* = \frac{2}{3 \cdot Z_p} \frac{L_r}{L_m} \frac{M^*}{\psi_{r0}^*(t)}$$

$$\Delta\omega_s = \frac{R_r}{L_r} L_m \frac{I_{ort}^*}{\psi_{r0}^*(t)}$$

Ограничение по току проводится аналогично п.1.

Входным сигналом контура регулирования скорости является сигнал задания скорости  $\omega^*$ , поступающий с выхода регулятора положения. При использовании ПИ-регулятора скорости, обеспечивающего астатическое регулирование скорости асинхронного двигателя формула для вычисления требуемого момента  $M^*$  может быть записана

$$M^* = K_{pc} \cdot (\omega^* - \omega) + \frac{1}{T_{и}} \cdot \int_0^t (\omega^* - \omega) dt ,$$

где  $K_{po}$  – коэффициент пропорциональной части регулятора скорости;  $T_{и}$  – постоянная времени регулятора;  $\omega^*$  – величина скорости ротора двигателя с датчика обратной связи по скорости. Величины  $K_{po}$ ,  $T_{и}$  вычисляются исходя из параметров двигателя и методов настройки регулятора на симметричный оптимум.

При использовании пропорционального регулятора положения, обеспечивающего статистическое регулирование углового положения асинхронного двигателя формула для вычисления требуемого задания скорости  $\omega^*$  может быть записана

$$\omega^* = K_{рп} \cdot (\varphi^* - \varphi) ,$$

где  $K_{рп}$  – коэффициент регулятора положения,  $\varphi^*$  – заданное угловое положение ротора двигателя,  $\varphi$  – действительное угловое положение ротора двигателя.

При использовании ПИ-регулятора положения, обеспечивающего астатическое регулирование углового положения ротора асинхронного двигателя формула для вычисления требуемого задания скорости  $\omega^*$  может быть записана как

$$\omega^* = K_{рп} \cdot (\varphi^* - \varphi) + \frac{1}{T_{и}} \cdot \int_0^t (\varphi^* - \varphi) dt,$$

где  $T_{и}$  – постоянная времени регулятора положения. Величины  $K_{рп}$ ,  $T_{и}$  вычисляются исходя из параметров двигателя и методов настройки регулятора положения на симметричный оптимум.

Описанные выше математические модели асинхронного электропривода реализованы в виде пакета прикладных программ (ППП) на языке фортран 77 на ПЭВМ совместимой с IBM PC. В основу построения ППП положен широко применяемый принцип модульности. Все модули ППП используют единые соглашения о связях по данным и управлению.

ППП состоит из следующих основных элементов:

- редактора подготовки исходных данных;
- вспомогательных программ,
- диалогового монитора;
- вычислительного модуля;
- модуля обработки результатов.

Редактор подготовки исходных данных служит для предварительного ввода в диалоговом режиме исходных параметров в заданном формате. Введенные данные записываются на магнитный диск в определенный пользователем файл. Вспомогательные программы позволяют выполнить вычисления некоторых коэффициентов и параметров, используемых в других подпрограммах и которые являются постоянными при моделировании. Работа с ППП ведется в диалоговом режиме. Для этого используются программы работы с «таблицами». С помощью этих «таблиц» организован ввод данных для моделирования асинхронного электропривода, подготовленных с помощью редактора подготовки исходных данных в процессе диалога. В вычислительный модуль входят программы, с помощью которых организуется и выполняется процесс интегрирования системы дифференциальных уравнений, описывающей асинхронный электропривод, вычисление управляющих сигналов в регуляторах тока, момента, скорости и положения, устройства ШИМа и других элементах привода. Результаты моделирования выводятся по желанию пользователя в процессе и/или после проведения расчета каждого варианта. Кроме того, предусмотрена возможность вывода результатов расчета после проведения расчета или расчетов. Результаты моделирования могут быть выведены как в цифровом виде, так и в графическом на экран дисплея и /или на принтер.

Процесс моделирования асинхронного электропривода разбит на следующие этапы:

- создание файла (файлов) исходных данных;

- изменение значений, если это необходимо, части исходных данных в диалоговом режиме;
- выполнение предварительных вычислений;
- основные вычисления и накопление результатов счета
- обработка результатов моделирования;
- получение «твердых» копий результатов моделирования.

ППП для моделирования асинхронного электропривода позволяет:

- исследовать динамические характеристики привода;
- характеристики переходного процесса;
- частотные характеристики;
- оценивать качество работы применяемых в пакете алгоритмов управления:
- управление с постоянством потокосцепления;
- управление с постоянством угла  $\varepsilon_{\varepsilon} = \pm 45^\circ$

Функциональная схема пакета прикладных программ приведена на рис. 1.8.

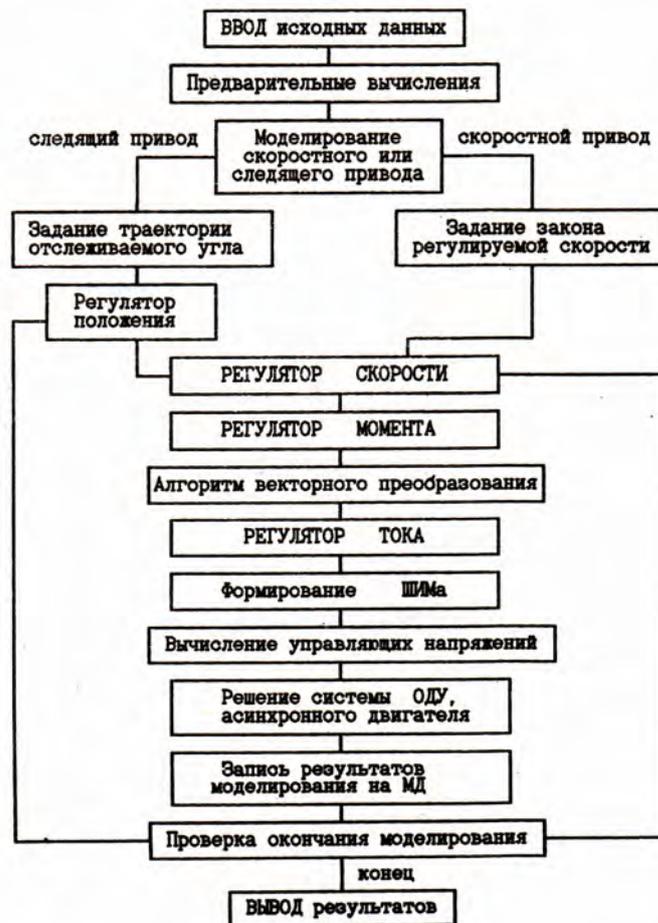


Рис. 1.8. Функциональная схема пакета прикладных программ.

### Результаты моделирования

При использовании разработанных математических моделей, алгоритмов управления и пакета прикладных программ проведено исследование характеристик асинхронного электропривода с векторным управлением на базе двигателя 4AA2M63B2Y3 ( $P_H=250$  Вт,  $\omega_H=314$  рад/сек) методом математического моделирования.

Проведен анализ динамических характеристик асинхронного электропривода при различных установившихся скоростях  $\omega_{уст}^* = (0.0314 \div 314)$  рад/сек, моменте  $M_c=0.01$  Мн, моменте инерции  $J = J_{дв}; 2J_{дв}$  и законах управления с постоянным потокосцеплением ротора  $\psi_r^* = \psi_{r0}^* = Const$  и оптимальным законом управления  $\varepsilon_\varepsilon = \pm 45^\circ$  ( $I_{syn}^* = I_{ort}^*$ )

Анализ характеристик показывает, что величины потокосцепления ротора  $\psi_r$  в двигателе отличаются от заданного в системе управления  $\psi_r^*$  в процессе пуска 16–25% при тенденции нарастания ошибки с увеличением значения установившейся скорости и появлением колебательного характера нарастания потокосцепления в двигателе  $\psi_r$ . Это вызывает необходимость учета явления насыщения в асинхронном двигателе при моделировании электромагнитных переходных процессов.

Наблюдалось полное совпадение заданного в системе управления и реального в двигателе синфазного токов  $I_{syn}^*$  и  $I_{syn}$ .

### 1.2.2. Гидравлические системы управления

В настоящее время для управления перемещением рабочих органов с помощью гидроприводов в гидроприводах находят применение гидрораспределители (ГР) с дискретными (включено-выключено) электромагнитами и дросселирующие гидрораспределители (ДГР). Гидрораспределители выполняют по существу функции движения, реверса и остановки движения гидродвигателей. Приводы с пропорциональным управлением часто называют «интеллектуальной гидравликой». Ниже представлен общий вид такого гидрораспределителя (рис. 1.9).



Рис. 1.9. Общий вид гидрораспределителя.

Гидропривода, в которых находят применение гидрораспределители дискретными (включено-выключено) электромагнитами, отличаются предельной простотой конструкции и, соответственно, низкой стоимостью, способны работать при повышенной степени загрязненности рабочей жидкости, однако для регулирования скорости необходима дополнительная установка дросселей или регуляторов расхода. Гидропривода с дросселирующими гидрораспределителями отличаются превосходными точностными и динамическими характеристиками, однако они значительно сложнее, требуют сверхпрецизионного изготовления основных деталей и наивысшего качества очистки рабочей жидкости, что многократно увеличивает их стоимость и снижает надежность при эксплуатации.

Далее простейшим вариантом является установка пропорционального электромагнита на обычный ГР. В результате мы получаем дополнительную возможность «приоткрывания» золотника, т. е. регулирования скорости движения гидродвигателя, однако практика показывает, что это регулирование - невысокого качества. Причины - недостаточная точность позиционирования, нестабильность расходной характеристики из-за неточности осевого расположения рабочих кромок золотника, повышенный гистерезис в результате сухого трения в подвижных элементах, зависимость расхода от перепада давлений, ограниченные динамические возможности аппарата. Поэтому вся последующая история пропорциональной электрогидравлики направлена на снижение влияния этих недостатков. Для повышения качества управления проектировщиками проводились работы, связанные с усовершенствованием пропорциональных электромагнитов и управляющей электроники, установки дополнительных датчиков обратной связи (ДОС), повышения точности изготовления золотниковой пары, использования ДГР в первом каскаде усиления, введения компенсаторов перепада давлений на рабочих кромках. При этом пределом целесообразности соответствующих усложнений конструкции пропорциональных электромагнитов является сохранение конкурентоспособности по сравнению с ДГР.

Аппараты с пропорциональным электроуправлением применяются в основном для дистанционного управления параметрами гидропривода. Возможно также их использование в качестве звеньев замкнутых систем автоматического регулирования. Объединение гидроприводов с электронными системами управления, применение «интеллектуальных» гидрокомпонентов со встроенной электроникой и специальных коммутационных средств (полевых шин) с открытой структурой позволяют успешно сочетать силовые и динамические качества гидравлики с быстроразвивающимися возможностями микроэлектроники и комплексных систем управления. В электрогидравлических приводах заметно растёт использование цифровых электронных устройств, обеспечивающих быструю обработку данных, простое программирование, повышенную надежность, числовое задание настроек и возможность длительного сохранения информации.

Достаточно хорошие перспективы применения имеется у ШИМ-усилителей, в которых изменение выходного сигнала достигается за счет широтно-импульсной модуляции. В результате амплитуда и частота генерируемых усилителем импульсов остаются постоянными, а ширина импульса определяется величиной входного сигнала управления. При этом пульсирующий характер сигнала, поступающего в обмотку пропорционального электромагнита, вызывает соответствующие колебания якоря и, возможно, золотника гидрораспределителя, что позволяет существенно снизить трение и улучшить характеристики аппарата в целом. Однако пульсирующий характер сигнала приводит к снижению ресурса изделия.

В распределителях с одним пропорциональным электромагнитом могут возникнуть опасные ситуации при отказе системы управления. Поэтому для повышения безопасности в ряде случаев применяют аппараты, в которых пружина в случае отказа устанавливает золотник в крайнюю позицию, обеспечивающую блокировку всех гидролиний.

Можно отметить, что одним из возможных направлений развития пропорциональной техники являются аппараты с управлением от шаговых электродвигателей.

Большое влияние развитие техники пропорционального управления оказывает и на современные регулируемые насосы. В частности, одной из новинок фирмы Parker является регулируемый насос с электрогидравлическим р/Q-управлением, в котором регулирование рабочего объема реализовано с помощью встроенного пропорционального распределителя с датчиком обратной связи по положению наклонного диска. В качестве компенсатора давления используется устанавливаемый отдельно предохранительный клапан с пропорциональным электроуправлением. Система управления имеет высокие надежность и быстродействие, она позволяет реализовать управление по давлению, а также электронное ограничение потребляемой мощности.

Фирма Bosch Rexroth использует систему управления, позволяющая бесступенчато регулировать давление и подачу насоса, ограничивать мощность, повысить точность регулирования благодаря использованию датчиков давления и угла поворота наклонного диска, а также повысить быстродействие до уровня, сравнимого с пропорциональными гидроаппаратами.

Возможность высокоточного регулирования основных параметров гидропривода позволяет обеспечить жесткие циклы управления. Вместе с тем для большинства мобильных машин необходимая величина давления в различных переходах рабочего цикла не может быть заранее установлена. Так, при перемещении шагающего робота работа гидропривода находится в тяжёлых условиях, т.к. один и тот же привод должен достаточно эффективно работать как в режиме передвижения ноги, так и в режиме её переноса. В этих случаях целесообразно использование регуляторов LS (Load Sensing – чувствительность к нагрузке), в которых перепад давлений на дросселе регулирования скорости (возможно пропорциональном) стабилизируется на

уровне  $\Delta p = 0,4 \dots 1,2$  МПа путем изменения фактической подачи насоса в режиме online, что обеспечивает минимизацию потерь мощности в гидроприводе.

Насосы с энергосберегающими LS-регуляторами широко используются всеми ведущими инофирмами в гидроприводах мобильных машин. При остановке технологических операций центральный контроллер подает команду на электронный блок двигателя и насосы. В результате снижаются обороты коленчатого вала до частоты холостого хода и устанавливаются подачи насосов, близкие к нулевым. Возобновление технологической операции сопровождается увеличением частоты вращения коленчатого вала и установкой дросселя системы LS в рабочее положение; при этом давление на выходе из насоса стабилизируется на уровне действующего сопротивления нагрузки плюс перепад давления  $\Delta p$ .

В современных гидроприводах техника пропорционального управления широко применяется и для решения простейших задач, например, плавного изменения законов разгона и торможения гидродвигателей, замены группы электрически переключаемых регуляторов расхода или давления одним пропорциональным аппаратом и т. п.

В заключение следует отметить, что гидроприводы с пропорциональным электроуправлением находятся в стадии интенсивного развития. Более подробные сведения об отечественной и импортной пропорциональной гидроаппаратуре и регулируемых насосах с полной расшифровкой кодовых обозначений и указанием габаритных и присоединительных размеров приведены в работе [43].

Ниже приведены структурные схемы гидропривода с LS-управлением.

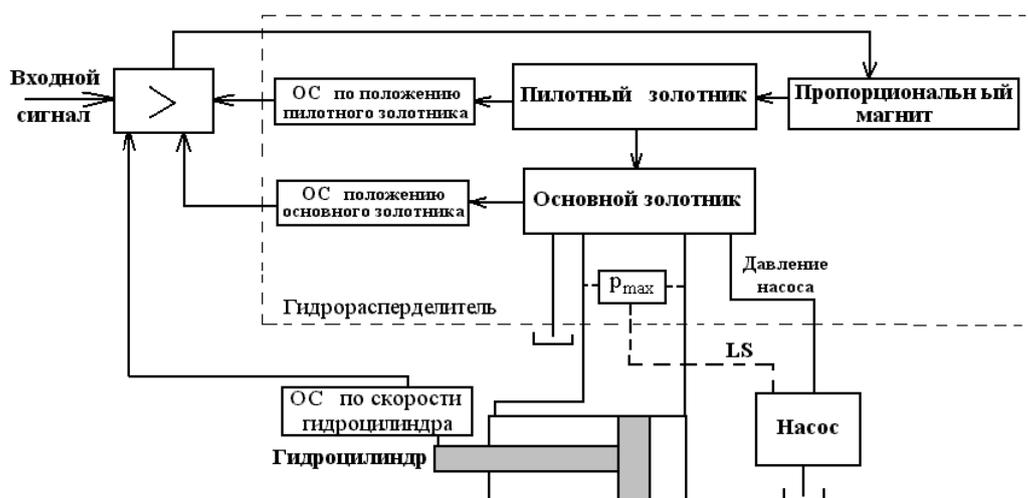


Рис. 1.10. Структурная схема гидропривода с LS-управлением.

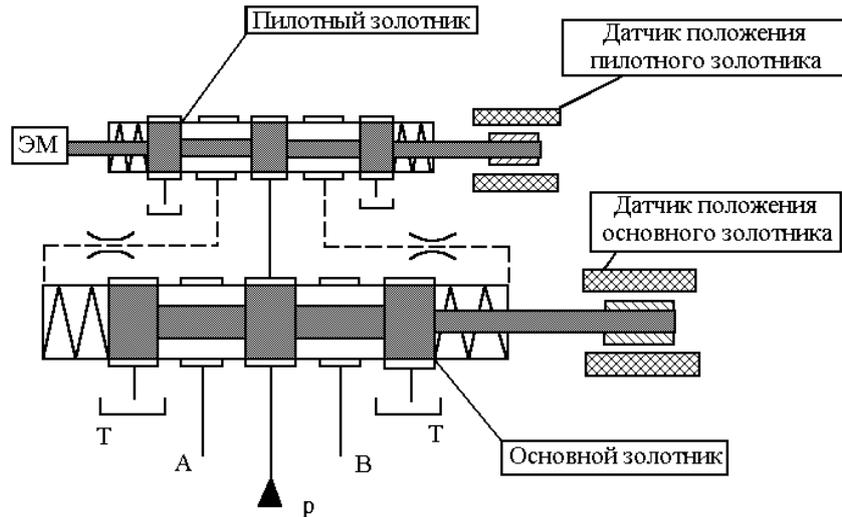


Рис. 1.11. Принципиальная схема пропорционального распределителя.

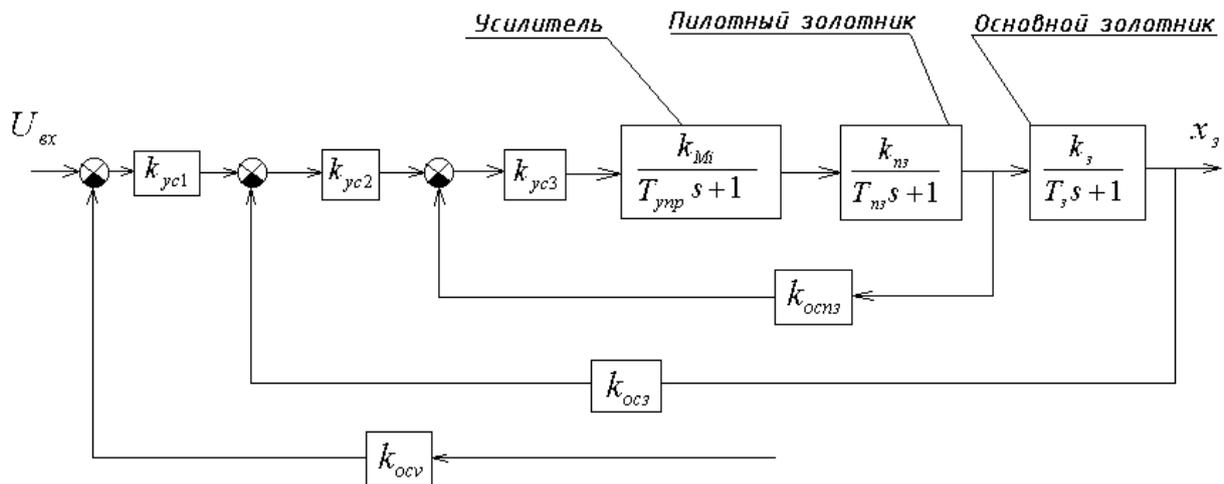


Рис. 1.12. Структурная схема пропорционального распределителя и электронного усилителя. (В этой схеме  $k_{осн3}$  – коэффициент обратной связи по положению пилотного золотника;  $k_{ос3}$  – коэффициент обратной связи по положению основного золотника;  $k_{осv}$  – коэффициент обратной связи по скорости основного золотника).

Термин LS (load sensing – чувствующий нагрузку) применяется для гидравлических систем, в которых мгновенное давление нагрузки служит сигналом обратной связи для управляющего устройства, которое в свою очередь, устанавливает необходимое давление насоса. Давление насоса поддерживается равным давлению нагрузки наиболее нагруженного потребителя плюс постоянное управляющее давление. С помощью компенсаторов давления поддерживается постоянный перепад давления на дросселях А1 и А2, что и определяет отсутствие зависимости

скорости потребителя от его нагрузки. Это и является основным принципом LS-системы.

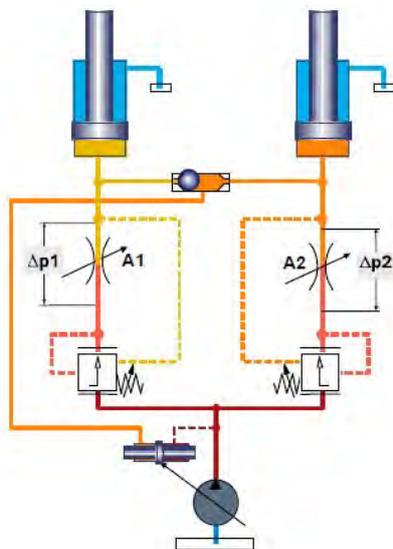


Рис. 1.13. Принципиальная схема гидросистемы с LS-управлением.

### Математические модели и моделирование гидравлического привода с LS-управлением шагающей машины

При создании гидравлической шагающей машины (ШМ) одной из главных задач, требующей своего решения, является выбор (проектирование) гидравлической системы. Это можно объяснить многими причинами и в частности энергетикой ШМ. Система управления ШМ должна обеспечивать согласованное движение каждой из степеней подвижности ноги путем одновременного управления сервоприводами в шарнирах ног в зависимости от положения машины на местности, показаний датчиков информации, управляющих сигналов от водителя или верхних уровней СУ автоматической ШМ, а также характеристик опорной поверхности. При этом должна учитываться специфика работы гидропривода при использовании его на автономной ШМ, а именно: жесткие массогабаритные ограничения; высокое быстродействие; ограниченная мощность приводного двигателя.

Здесь рассмотрены несколько схем гидроприводов с объёмно-дрессельным регулированием, предназначенных для применения в системе управления ШМ. Дано описание рассматриваемых схем гидроприводов, приведены математические модели, описывающие процессы, протекающие в них. Были рассмотрены варианты схем гидросистем, в которых используются насосы переменной производительности с LS-управлением (load sensing control). В качестве возможных вариантов были рассмотрены гидросхемы с одним насосом, работающем на все гидроцилиндры, с двумя насосами, каждый из которых работает на гидроцилиндры ног, составляющих одну «трепку» и схема с шестью насосами, по одному насосу на каждую ногу.

Математические модели гидравлического привода были составлены для шестиногой ШМ, имеющей следующие характеристики: полная масса 10000 кг, длина 7.5 м, ширина 3.6 м, высота 3.8 м., высота преодолеваемых препятствий до 1.5 м, максимальная скорость 0.6 м/с [9]. Расчетные нагрузки и скорости в степенях подвижности ноги при движении ШМ «трешками» приведены на рис. 1.14 [43]. Схема ноги ШМ приведена на рис. 1.15 [43].

### **Анализ основных типов гидросхем ШМ**

В качестве возможных вариантов гидросхем для применения на шестиногой ШМ рассмотрим основные типы гидроприводов, применяемых в машиностроении:

- Гидроприводы с дроссельным регулированием.
- Гидроприводы с объёмным регулированием.
- Гидроприводы с объёмно-дроссельным регулированием.

Гидроприводы с дроссельным регулированием здесь не будем рассматривать из-за их низкого к.п.д. Проект машины, для которой проводился анализ гидросхем, имеет 18 степеней свободы, каждая из которых управляется гидроцилиндром [43]. Так как скорости движения и нагрузки на каждую степень свободы в общем случае различны, то скорость каждого гидроцилиндра должна будет регулироваться при помощи электрогидравлического усилителя с дроссельным управлением. По оценкам [43] это приводит к большим потерям мощности, неудовлетворительному тепловому режиму, и, следовательно, к необходимости использовать приводной двигатель значительно большей мощности (500-600 кВт), чем мощность полезной нагрузки.

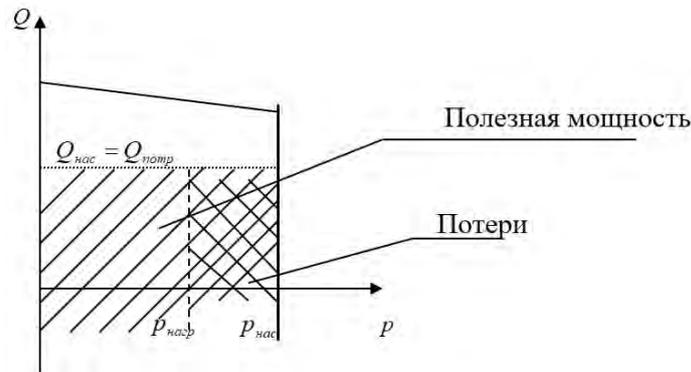
Гидроприводы с объёмным регулированием наряду с преимуществами, присущими этому типу, основным из которых является высокий к.п.д., при использовании их на шестиногой ШМ, имеют некоторые недостатки. Так как в приводе с объёмным регулированием изменение подачи производится путем изменения рабочего объёма насоса, то для управления скоростью одного гидроцилиндра необходимо использовать один регулируемый насос, т.е. для 18 независимых гидроцилиндров 18 регулируемых насосов. Это приводит к усложнению гидропривода, ухудшению массогабаритных характеристик конструкции ШМ. Быстродействие гидропривода с объёмным регулированием определяется временем, необходимым для изменения подачи насоса, которое может составить до нескольких десятых долей секунды. Кроме того, регулируемые машины более дороги и менее долговечны по сравнению с нерегулируемыми. Тем не менее схема гидропривода с объёмным регулированием была применена на шестиногой шагающей машине ASV [12].

Уменьшение потерь и одновременно высокое быстродействие можно получить в гидроприводах с объёмно-дроссельным регулированием. В них регулируемые гидромашины применяются вместе с аппаратами, регулирующими расход рабочей жидкости.

В настоящее время в связи с увеличением энергонасыщения современных машин, большим количеством исполнительных механизмов (сельскохозяйственные, дорожно-строительные машины и т.д.) существует необходимость в насосных установках, способных работать на несколько потребителей. В таких случаях используются следующие типы насосов:

а) Насосы, работающие при постоянном давлении.

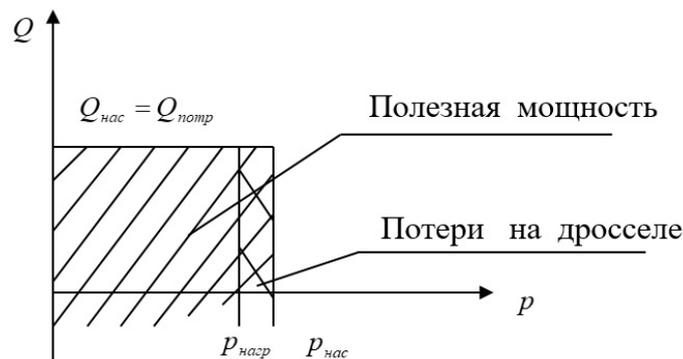
Их рабочая характеристика имеет вид:



Этот тип насосов обеспечивает подачу рабочей жидкости равную расходу потребителя. При этом давление нагнетания остаётся постоянным и определяется усилием пружины регулятора.

б) Насосы с LS-управлением (Load sensing – чувствительный к нагрузке).

Основной принцип LS-управления заключается в том, что угол наклона шайбы насоса и, следовательно, его производительность регулируются в зависимости от перепада давления на регулируемом дросселе (дросселирующем распределителе). Рабочая характеристика таких насосов, имеет вид:



### Устройство и принцип работы LS-регулятора

Регулятор подачи насоса состоит (рис. 1.16) из распределителя РЗ, двух гидроцилиндров управления ЦУ1, ЦУ2, пружины П, дросселя ДР. На левый торец золотника распределителя РЗ действует давление из линии нагнетания насоса, а на правый - давление в полости силового гидроцилиндра ЦС. Разность этих давлений равна перепаду давления на регулируемом дросселе

РДР  $\Delta p_{др}$ . На правый торец золотника, кроме того, действует усилие от пружины. Если сила, возникающая под действием перепада давления на регулируемом дросселе,  $\Delta p_{др} = p_n - p_{цил}$  меньше силы предварительного поджатия пружины золотника распределителя РЗ, то золотник находится в крайнем левом положении. При этом он соединяет полость ЦУ2 со сливом. Под действием пружины П и давления в линии нагнетания насоса, действующего на поршень ЦУ1, шайба насоса отклонена на максимальный угол, обеспечивая максимальную подачу насоса. Если необходимо уменьшить скорость исполнительного гидроцилиндра, то уменьшают площадь проходного сечения регулируемого дросселя. При этом начнет увеличиваться давление насоса, и следовательно, перепад давления на дросселе РДР  $\Delta p_{др} = p_n - p_{цил}$ . Как только сила, вызванная этим перепадом давления, превысит силу предварительного поджатия пружины, золотник распределителя РЗ начнет двигаться вправо, постепенно соединяя линию нагнетания насоса с полостью гидроцилиндра управления ЦУ2. Под действием давления, поршень гидроцилиндра ЦУ2 начнет двигаться, преодолевая усилие пружины П и силы давления в ЦУ1. По мере выдвижения штока гидроцилиндра ЦУ2 угол наклона шайбы начнет уменьшаться, снижая при этом подачу насоса, а, следовательно, и перепад давления на дросселе РДР. Из-за уменьшения величины  $\Delta p_{др}$  пружина золотника распределителя РЗ начнет возвращать его в исходное положение, перекрывая канал, соединяющий линию нагнетания насоса и гидроцилиндра ЦУ2. Когда силы, действующие на золотник, уравниваются друг друга, движение прекратится и он займет положение, при котором небольшое количество жидкости будет проходить через его щель из линии нагнетания насоса через дроссель ДР в полость гидроцилиндра ЦУ2 и далее через канал в поршне на слив, поддерживая тем самым давление, необходимое для удержания шайбы насоса в промежуточном равновесном состоянии. Подача насоса при этом будет соответствовать расходу, необходимому для обеспечения заданной скорости исполнительного гидроцилиндра и будет равна пропускной способности дросселя РДР. Если необходимо увеличить скорость движения гидроцилиндра, то увеличивают проходное сечение дросселя РДР. При этом перепад давления на нем уменьшается и золотник распределителя РЗ смещается вправо, открывая канал, соединяющий гидроцилиндр ЦУ2 со сливом. Под действием пружины и силы давления в гидроцилиндре ЦУ1 шайба насоса начнет отклоняться в сторону увеличения угла наклона, увеличивая при этом подачу. Перепад давления, при котором начинает двигаться золотник распределителя РЗ, выбирается обычно в пределах 1.4-2.5 МПа и регулируется путем настройки силы предварительного поджатия пружины. Максимальная подача выбирается таким образом, чтобы обеспечить наибольшую требуемую скорость движения выходного звена гидродвигателя при максимальном проходном сечении дросселя. Направляющий распределитель Р служит для реверса силового гидроцилиндра. Клапан

«ИЛИ» необходим для определения большего давления в силовом гидроцилиндре СЦ, которое служит для управления LS-золотника распределителя РЗ.

Таким образом, как видно из приведенных выше рассуждений, насосы с LS-регулируемым обеспечивают достаточно высокий к.п.д привода. Это можно объяснить тем, что они производят подачу рабочей жидкости равную потребностям гидродвигателя, а давление, развиваемое насосом, равно давлению нагрузки в силовом цилиндре за вычетом потерь на регулируемом дросселе, которые сравнительно невелики (1.4-2.5 МПа).

На рис. 1.17 показана принципиальная схема насоса, в котором кроме LS-регулируемого используется также ограничитель давления ОГД, выполняющий функции предохранительного клапана. Рабочая характеристика такого насоса при рабочем давлении меньшем чем давление срабатывания ОГД (35.0 МПа) будет соответствовать рабочей характеристике насоса с LS-управлением. Как только давление достигнет величины, равной ~35.0 МПа, золотник ОГД соединит линию нагнетания насоса с ЦУ2 и шайба насоса займет нулевое положение, при котором подача рабочей жидкости будет равняться величине внутренних утечек в насосе.

Одним из вариантов LS-управления являются насосы с ограничителем мощности (рис. 1.18). Они используются для избегания перегрузок и остановок приводного двигателя. Принцип действия ограничителя мощности заключается в том, чтобы поддерживать мощность, потребляемую насосом на постоянном уровне в независимости от изменения давления.

Таким образом, можно сделать вывод, что наиболее подходящим для применения на шестиногой шагающей машине является гидропривод с объемно-дроссельным регулированием с использованием насоса с LS-управлением и ограничителем давления.

На рис. 1.19 представлена схема использования насоса с LS-управлением и ограничителем давления для управления одноштоковым гидроцилиндром.

### **Регулирующая аппаратура**

Для регулирования скорости движения исполнительных гидроцилиндров можно применить:

- дросселирующие гидрораспределители с пропорциональным управлением и электрической обратной связью по положению золотника (рис. 1.20);
- двухкаскадные электрогидравлические усилители с электрической обратной связью по положению золотника.

В связи с тем, что в последнее время в мировой практике имеется тенденция к более широкому применению пропорциональной техники, поэтому мы остановим свой выбор на двухкаскадном распределителе с пропорциональным управлением.

### **Принцип работы пропорционального распределителя**

Пропорциональный распределитель (рис. 1.20) состоит из пилотного золотника ПЗ, основного золотника ОЗ, электромагнита ЭМ и датчиков положения этих золотников ДОС.

Функционирование распределителя с пропорциональным управлением можно описать следующим образом. Рабочая жидкость под давлением поступает по каналам к основному и пилотному золотникам. При отсутствии электрического сигнала на катушках пропорционального магнита пилотный золотник находится в нейтральном положении и расход жидкости в каналах управления отсутствует. При подаче напряжения на катушки электромагнита происходит смещение пилотного золотника. Одна из камер управления основного золотника соединяется с линией высокого давления, а другая со сливом. Под действием возникшего перепада давления происходит смещение золотника в сторону камеры с меньшим давлением. При этом сигнал с датчиков обратной связи вычитается из входного сигнала, тем самым уменьшая входное напряжение на обмотках электромагнита. При уменьшении напряжения на обмотках электромагнита, пружины пилотного золотника начнут возвращать его в исходное положение, перекрывая тем самым каналы управления основного золотника, что приведет к его остановке. Для возвращения основного золотника в нулевое положение необходимо изменить знак входного напряжения, при котором пилотный золотник сместится в противоположную сторону и в управляющих полостях возникнет перепад давлений, в результате которого основной золотник возвращается к исходному состоянию. Обратные связи по положению золотников служат для улучшения динамических характеристик распределителя и уменьшения петли гистерезиса.

### **Исполнительные гидроцилиндры**

В качестве исполнительных двигателей в ШМ можно использовать: двухштоковые гидроцилиндры, одноштоковые гидроцилиндры, гидроцилиндры с фальшштоком.

Применение двухштоковых гидроцилиндров усложняет компоновку ноги. Если исходить из соображений компактности ноги, то наиболее приемлемым является вариант с одноштоковым гидроцилиндром. Но этому типу двигателя присущ существенный недостаток, который заключается в необходимости подавать разные объёмы рабочей жидкости при движении гидроцилиндра в противоположных направлениях. Этого недостатка можно избежать, если применить дифференциальную схему подключения гидроцилиндра, которая заключается в том, что при движении в направлении штока обе полости гидроцилиндра соединяются с напорной линией, а при движении в противоположном направлении штоковая полость продолжает соединяться с напорной линией, а поршневая соединяется со сливом рис. 1.20.

Если выбрать соотношение площадей поршня и штока, как  $F_{п}=2F_{шт}$ , то в таком случае можно обеспечить равные расходы жидкости при прямом и обратном ходах поршня гидроцилиндра.

На рис. 1.19 представлена принципиальная схема гидропривода с использованием насоса с LS-регулированием, дифференциального гидроцилиндра и пропорционального распределителя.

При смещении распределителя ПР вправо линия нагнетания насоса соединяется с поршневой полостью силового гидроцилиндра ЦС. Под действием давления поршень цилиндра начинает двигаться вправо, при этом рабочая жидкость из штоковой полости не пойдет на слив, так как обратный клапан КО1 не пропустит её, а попадет снова в напорную линию насоса. Так как отношение площадей поршневой и штоковой полостей равно 2/1, то подача насоса будет такой же как и при соединении линии нагнетания со штоковой полостью.

### **Возможные варианты схем**

В качестве возможных схем гидропривода с использованием насосов с LS-управлением рассмотрим следующие варианты:

- Один насос подаёт рабочую жидкость для приводов одной ноги (всего 6 насосов) (рис. 1.21).
- Один насос подает рабочую жидкость ко всем 18 исполнительным гидроцилиндрам (рис. 1.22).
- Так как основным режимом движения машины является походка «трешками», то привода, принадлежащие к одной «трешке», движутся одинаково относительно корпуса и, следовательно, имеют близкие по величине нагрузки и скорости. Поэтому можно использовать один насос для гидроцилиндров, принадлежащих к одной «трешке». Движение ног разных «трешек» отличаются только сдвигом по времени на половину периода движения машины (рис. 1.23);
- Один насос подаёт рабочую жидкость к гидроцилиндрам одинаковых степеней свободы, принадлежащие к одной «трешке», т.е. к приводам бедер, голеней, поворотов стоек (6 насосов).

### **Выбор рабочего давления**

Так как количество различного гидрооборудования, размещенного на ШМ достаточно велико, то с целью улучшения массогабаритных характеристик следует использовать более высокое рабочее давление. В настоящее время многие фирмы производители оборудования для гидроприводов используют рабочее давление 350 атм. В связи с этим предполагается в качестве рабочего давления в гидроприводе шагающей машины использовать давление в 350 атм.

Рабочие площади гидроцилиндров выбираем с учетом максимальных нагрузок и величины давления, исходя из нагрузок, которые должны будут преодолевать гидропривода каждой степени свободы.

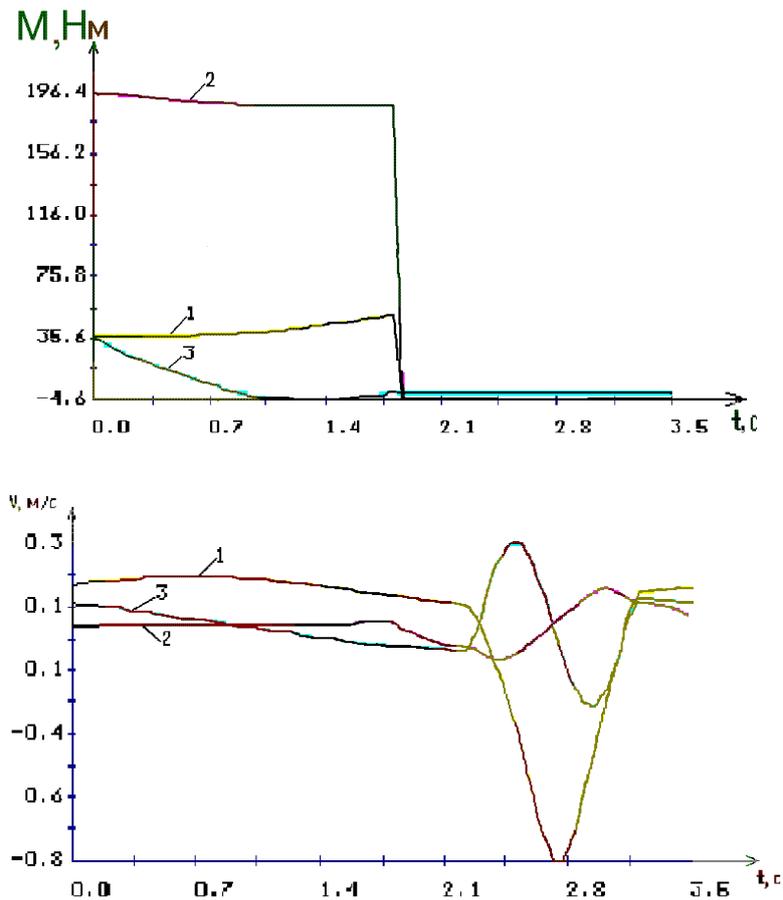


Рис. 1.14. Нагрузки и скорости в шарнирах ноги ШМ при движении «трешками» (1 – поворот стойки, 2 – бедро, 3 – голень).

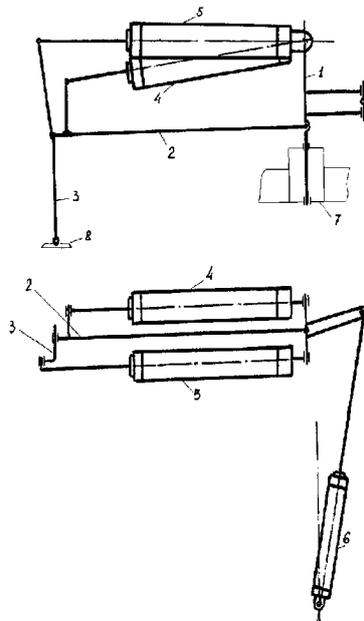


Рис. 1.15. Кинематика и конструктивные параметры ноги (1 – стойка; 2 – бедро; 3 – голень; 4 – ГЦ привода бедра; 5 – ГЦ привода голени; 6 – ГЦ привода поворота стойки; 7 – корпус; 8 – стопа).

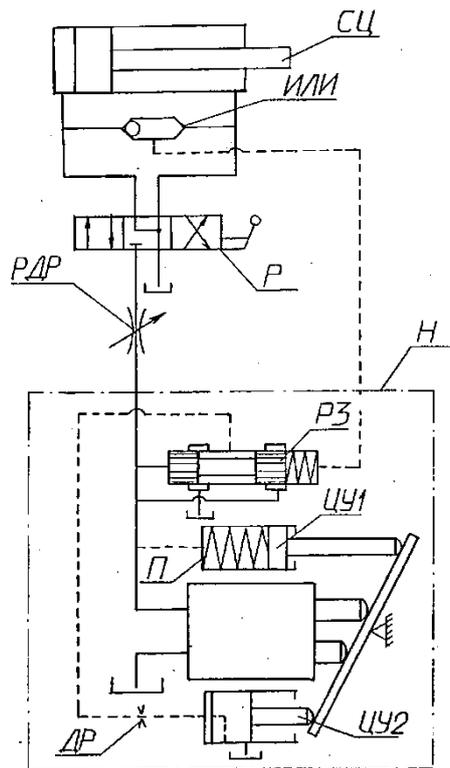


Рис. 1.16. Принципиальная схема насоса с LS-управлением.

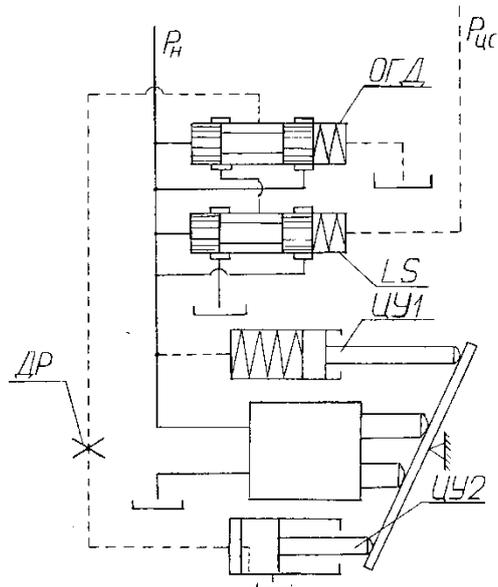


Рис. 1.17. Принципиальная схема насоса с LS-регулирующим и ограничителем давления.

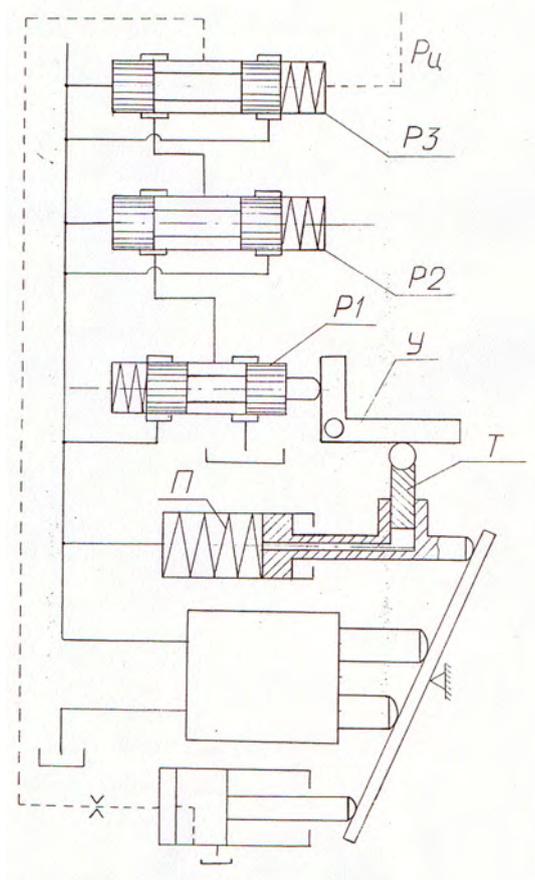


Рис. 1.18. Принципиальная схема насоса с LS-регулирующим и ограничителем мощности.

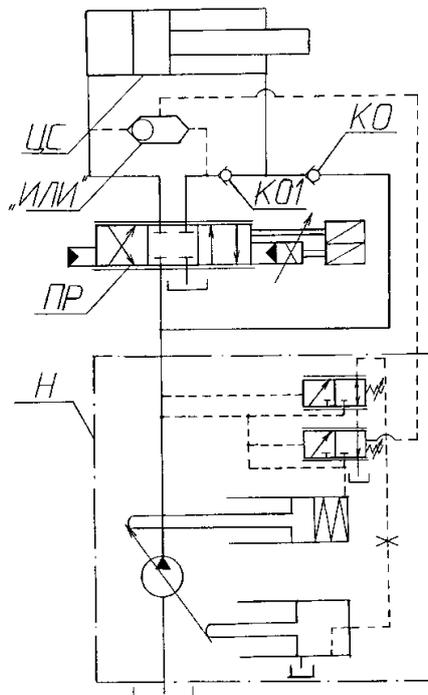


Рис. 1.19. Схема насоса с LS-управлением и ограничителем давления для управления одноштоковым гидроцилиндром.

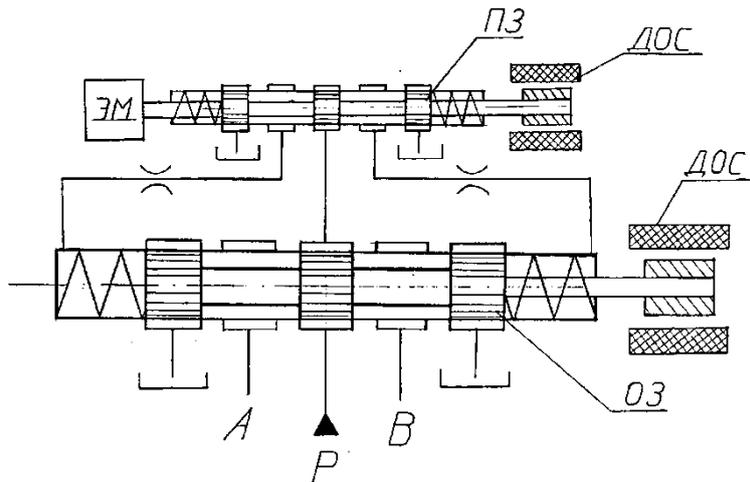


Рис. 1.20. Дросселирующий гидрораспределитель с пропорциональным управлением и электрической обратной связью по положению золотника.

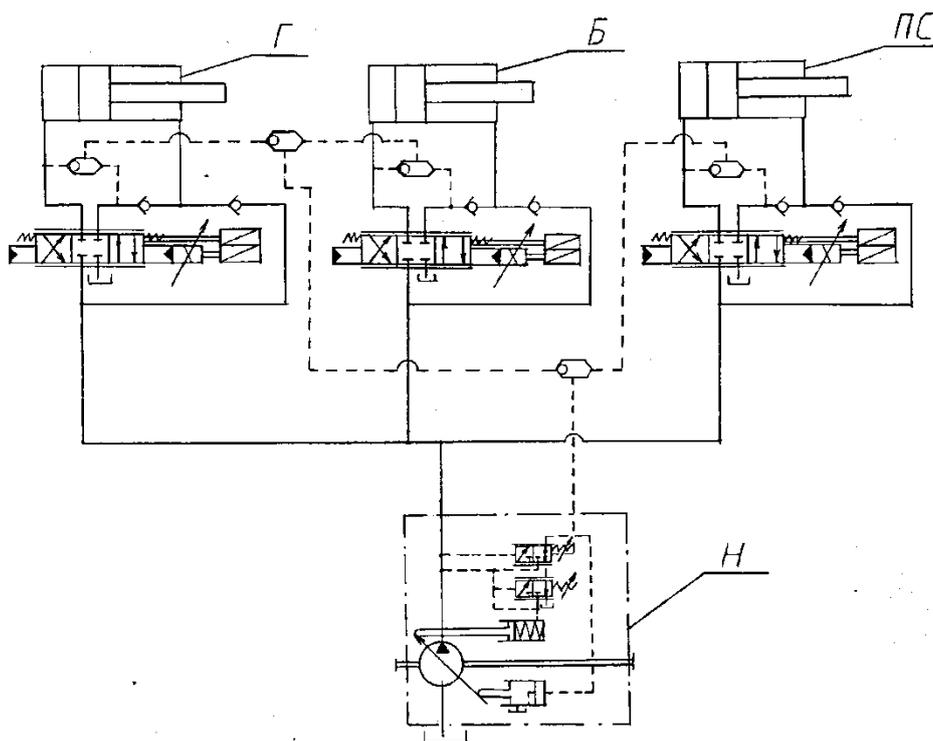


Рис. 1.21. Принципиальная схема гидропривода с насосом с LS-регулированием, дифференциальным гидроцилиндром и пропорциональным распределителем.

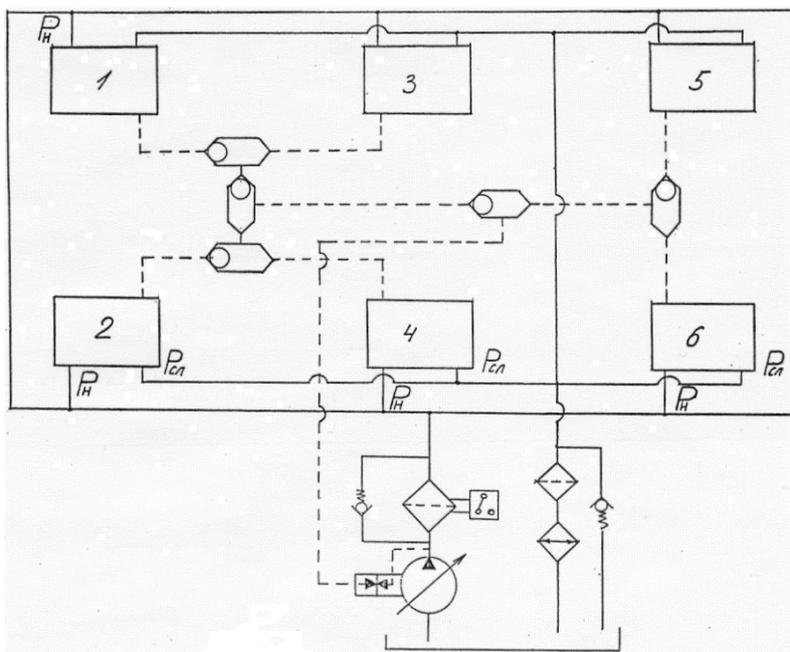


Рис. 1.22. Схема гидросистемы, когда один насос подает рабочую жидкость в исполнительные гидроцилиндры шести ног.

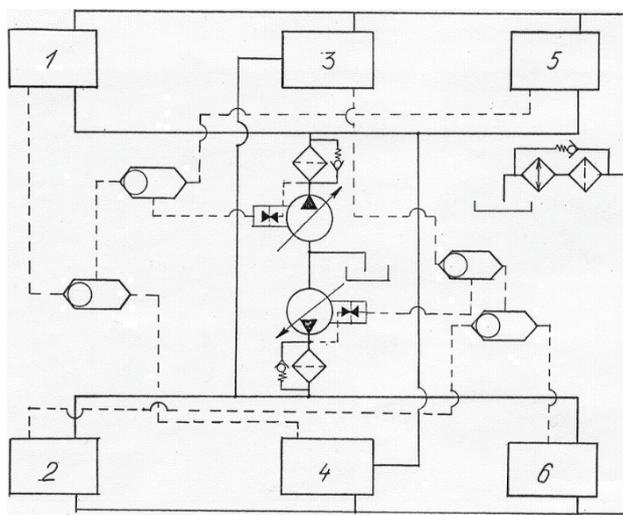


Рис. 1.23. Схема гидросистемы, в которой имеется 2 насоса, каждый из которых обеспечивает рабочей жидкостью ноги одной «трешки».

### Моделирование динамики гидропривода ноги шагающей машины в системе MATLAB Simulink.

В настоящее время имеются два основных подхода - к математическому описанию динамики автоматических систем. Первый подход базируется на передаточных функциях и тесно связанных с ним частотных методах, второй - на методах пространства состояний. Метод передаточных функций и частотные характеристики к середине 50-х годов практически позволили решить проблему проектирования линейных автоматических систем с одним

входом и выходом. Несмотря на тенденцию широкого внедрения ЭВМ в область анализа и синтеза автоматических систем; частотные методы не утратили своего значения и сейчас. Реализация их на ЭВМ дает возможность в короткий срок получить ценную информацию о проектируемой системе. По амплитудно-фазовым частотным характеристикам можно судить о таких качественных показателях, как запасы устойчивости по амплитуде и по фазе, резонансная частота, частота среза и т.д. Комбинирование частотных и корневых методов при автоматизированном анализе и синтезе линейных систем высокого порядка часто позволяет получить достаточно полную информацию для синтеза.

Проектирование систем управления с помощью амплитудно-фазовых частотных характеристик дает возможность анализировать структуры и влияние параметров объекта и отдельных его частей, решать задачи синтеза регулятора путем подбора корректирующих звеньев, выполнять идентификацию по экспериментально снятым частотным характеристикам и решать другие задачи.

В работе приведена линеаризованная математическая модель, описывающая динамику гидропривода ноги шагающей машины (ШМ). В гидроприводе использованы распределители с пропорциональным управлением, насос с LS-управлением и гидроцилиндры.

Моделирование гидропривода, основанное на методах пространства состояний, было проведено выше.

Здесь представлена структурная схема гидропривода, состоящая из передаточных функций соответствующего вида. В результате моделирования динамики гидропривода были получены логарифмические амплитудные фазовые частотные характеристики. На них показаны запасы по фазе и амплитуде, которые позволяют судить об устойчивости системы. Также показаны переходные процессы для гидроприводов каждой степени свободы ноги (голень, поворот стойки, бедро). Результаты моделирования линеаризованной модели гидропривода, проведенного с помощью системы "MATLAB+Simulink", представлены на графиках.

### **Линейная модель гидропривода с LS-регулированием.**

При составлении линейной модели гидропривода ШМ были приняты следующие предположения и допущения: коэффициенты расходов дросселей и рабочих окон золотника являются постоянными; гидродинамические силы, действующие на золотники LS-регулятора, ограничителя давления, пилотного золотника пропорционального распределителя пренебрежимо малы; перетечки рабочей жидкости через радиальные зазоры золотников и гидроцилиндров пренебрежимо малы; нагружающий момент на шайбе насоса создается приведенной к штоку управляющего гидроцилиндра массой подвижных частей; давление слива постоянно; величины вязкости и модуля объёмной упругости не изменяются.

На рис. 1.24, 1.25 приведены соответственно принципиальная и структурная схемы гидропривода с LS-управлением. На рис. 1.26 приведена принципиальная схема пропорционального распределителя.

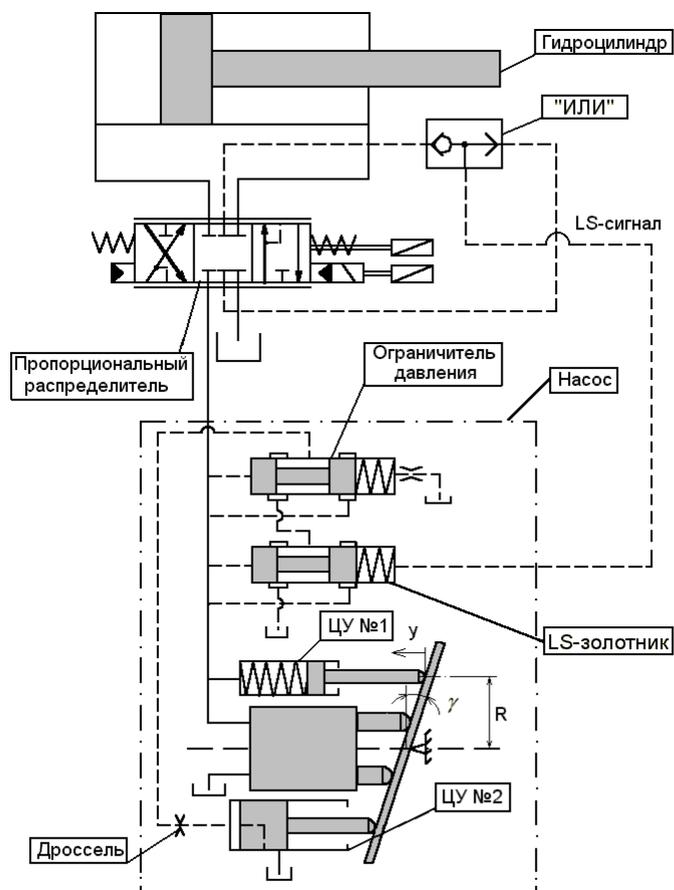


Рис. 1.24. Принципиальная схема привода с LS-управлением.

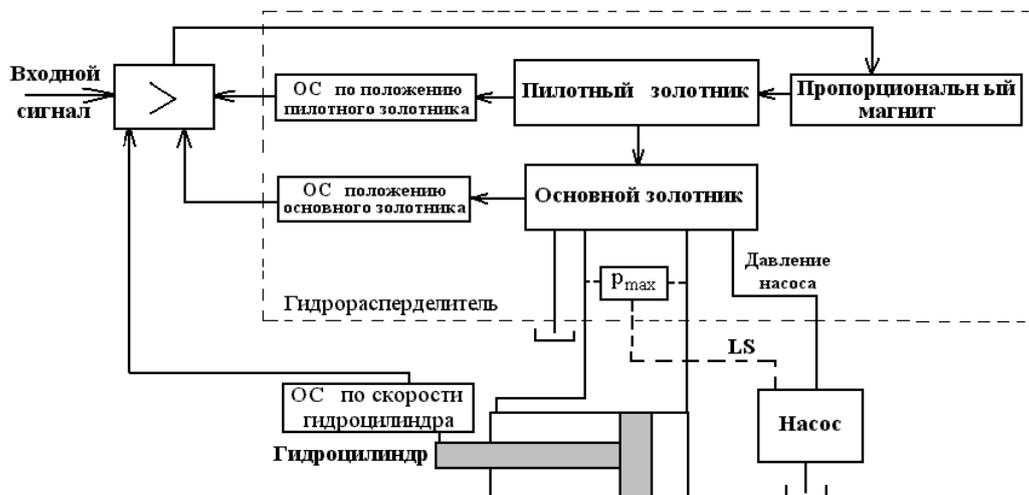


Рис. 1.25. Структурная схема гидропривода с LS-управлением.

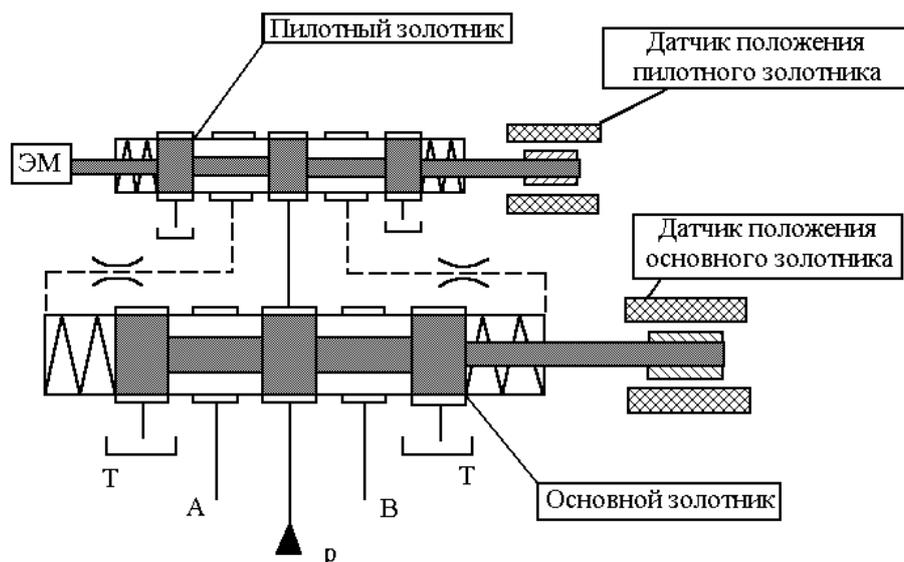


Рис. 1.26. Принципиальная схема пропорционального распределителя.

Математическая модель двухступенчатого гидрораспределителя (рис. 1.11, 1.13) с пропорциональным управлением представляет собой совокупность уравнений, записанных для отдельных его элементов [9, 12].

Уравнение напряжений в электрической цепи магнита (ЭМ):

$$U_{yc} = R_y I_y + L_y \frac{dI_y}{dt} + C_l \frac{dx_{п.з.}}{dt}, \quad (1.8)$$

где  $U_{yc}$  – напряжение на выходе электронного усилителя;  $R_y$  – активное сопротивление обмотки управления электромагнита;  $L_y$  – индуктивность обмотки управления электромагнита;  $C_l$  – коэффициент противо-ЭДС;  $x_{п.з.}$  – смещение якоря электромагнита пилотного золотника.

После преобразований уравнение цепи управления ЭМ будет иметь вид:

$$T_{ynp} \cdot \frac{di_y}{dt} + i_y = k_{iu} \cdot U_y \quad (1.9)$$

Из уравнения цепи управления электромагнита следует, что передаточная функция цепи управления электромагнита будет следующей:

$$W(s) = \frac{k_{Fi} \cdot k_{iu}}{T_{ynp} \cdot s + 1} \quad (1.10)$$

Здесь  $T_{упр} = \frac{L_y}{R_y + R_{вых}}$  – постоянная времени передаточной функции цепи управления электромагнита, где  $L_y$  – индуктивность обмотки управления электромагнита;  $R_y$  – активное сопротивление обмотки электромагнита;  $R_{вых}$  – сопротивление выходного каскада электронного усилителя;  $k_{Fi}$  – постоянная электромагнита,  $k_{iu} = \frac{1}{R_y + R_{вых}}$  – коэффициент передачи электронного усилителя.

Уравнение движения плунжера пилотного золотника:

$$m_{п.з} \frac{d^2 x_{п.з}}{dt^2} + k_{т.п.з} \frac{dx_{п.з}}{dt} + c_{пр.п.з} x_{п.з} + R_{гд.п.з} = P_{элм}, \quad (1.11)$$

где  $m_{п.з}$  – масса плунжера пилотного золотника;  $k_{т.п.з}$  – коэффициент, учитывающий вязкое трение в зазоре между плунжером пилотного золотника и гильзой;  $R_{гд.п.з}$  – гидродинамическая сила, действующая на пилотный золотник со стороны рабочей жидкости;  $P_{элм}$  – сила электромагнита;  $c_{пр.п.з}$  – жесткость пружины пилотного золотника.

Пренебрегая массой пилотного золотника, получим уравнение движения пилотного золотника в виде:

$$k_{мпнз} \cdot \frac{dx_{пз}}{dt} + c_{пз} \cdot x_{пз} = P_{элм} \quad (1.12)$$

где  $k_{мпнз}$  – коэффициент вязкого трения пилотного золотника,  $c_{пз}$  – жесткость центрирующих пружин пилотного золотника,  $P_{элм}$  – сила, действующая со стороны электромагнита.

Отсюда передаточная функция пилотного золотника:

$$W_{пз}(s) = \frac{k_{пз}}{T_{пз} s + 1} \quad (1.13)$$

Здесь  $T_{пз} = \frac{k_{мпнз}}{c_{пз}}$  – постоянная времени пилотного золотника,  $k_{пз} = \frac{1}{c_{пз}}$  – коэффициент усиления пилотного золотника,  $P_{элм} = k_{Fi} \cdot i$  – электромагнитная сила пропорционального магнита, действующая на пилотный золотник.

Расход жидкости через пилотный золотник:

$$Q_y = k_{Qx_{пз}} \cdot x_{пз} - k_{Qp_y} \cdot p_y \quad (1.14)$$

где  $p_y = p_{y1} - p_{y2}$ , коэффициенты линеаризации:

$$k_{Qx_{nz}} = k'_{nz} \cdot \sqrt{\frac{p_n - p_{cl} \mp p_{n0}}{2}}$$

$$k_{Qp_y} = \frac{k'_{nz} \cdot x_{z0}}{2 \cdot \sqrt{2 \cdot (p_n - p_{cl} \mp p_{n0})}}$$
(1.15)

$k'_{nz} = \mu_{nz} \cdot \pi \cdot d_{nz} \cdot k_{nz} \cdot \sqrt{\frac{2}{\rho}}$  – проводимость окон пилотного золотника,  $p_n$  – давление питания,  $p_{n0}$  – давление нагрузки в точке линеаризации,  $\rho$  – плотность рабочей жидкости,  $d_{nz}$  – диаметр пилотного золотника,  $\mu_{nz}$  – коэффициент расхода окон пилотного золотника.

С другой стороны, расход жидкости на управление равен

$$Q_y = F_3 \cdot \frac{dx_3}{dt}$$
(1.16)

Сжимаемостью рабочей жидкости в полостях управления основного золотника пренебрегаем из-за малости объёма.

Уравнение движения плунжера основного золотника имеет вид:

$$m_3 \frac{d^2 x_3}{dt^2} + k_{rp.3} \frac{dx_3}{dt} + c_{np.3} x_3 + R_{гд.3} + R_{rp.3} = F_3 \Delta p_y,$$
(1.17)

где  $m_3$  – масса плунжера основного золотника;  $k_{rp.3}$  – коэффициент, учитывающий вязкое трение в зазоре между плунжером основного золотника и гильзой;  $c_{np.3}$  – жесткость пружины основного золотника;  $R_{гд.3}$  – гидродинамическая сила, действующая на золотник со стороны рабочей жидкости;  $R_{rp.3}$  – сила сухого трения действующая на золотник;  $\Delta p_y$  – перепад рабочей жидкости на торцах основного золотника.

Пренебрегая массой основного золотника, получим:

$$k_{мпз} \cdot \frac{dx_3}{dt} + c_3 \cdot x_3 = p_y \cdot F_3$$
(1.18)

Выразим из этого уравнения  $p_y$  и подставим его в (1.8). Приравняв (1.8) и (1.9) получим

$$\left( \frac{F_3^2 + k_{Qp_y} \cdot k_{мпз}}{k_{Qp_y} \cdot c_3} \right) \cdot \frac{dx_3}{dt} + x_3 = \frac{F_3 \cdot k_{Qx_{нз}}}{k_{Qp_y} \cdot c_3} \cdot x_{нз} \quad (1.19)$$

где  $k_{мпз}$  – коэффициент вязкого трения между плунжером основного золотника и гильзой;  $c_3$  – жесткость центрирующих пружин основного золотника;  $p_y$  – разность давлений в полостях управления основного золотника;  $F_3$  – площадь торца плунжера основного золотника.

После простых преобразований передаточную функцию для основного золотника запишем в виде:

$$W(s) = \frac{k_3}{T_3 s + 1}, \quad (1.20)$$

где  $T_3 = \frac{F_3^2 + k_{Qp_y} k_{мпз}}{k_{Qp_y} c_3}$  – постоянная времени передаточной функции основного золотника,  $k_3 = \frac{F_3 \cdot k_{Qx_{нз}}}{k_{Qp_y} \cdot c_3}$  – коэффициент усиления передаточной функции основного золотника.

Уравнения расходов через основной золотник.

Расход через основной золотник:

$$Q_3 = \mu \cdot \pi \cdot d_3 \cdot \kappa_n \cdot x_3 \sqrt{\frac{2}{\rho}} \sqrt{|p_n - p_{uc}|} \cdot \text{sign}(p_n - p_{uc}) \quad (1.21)$$

Линеаризованное уравнение расхода через основной золотник:

$$Q_3 = Q_{uc} = k_{Qx_3} \cdot x_3 + k_{Qp} (p_n - p_{uc}) \quad (1.22)$$

где

$$k_{Qx_3} = \frac{\partial Q_3}{\partial x_3} = k'_3 \cdot \sqrt{\frac{p_n - p_{cl} \mp p_{uc0}}{2}}$$

$$k_{Qp} = \frac{\partial Q_3}{\partial p_{uc}} = \frac{k'_3 \cdot x_{30}}{2 \sqrt{2 \cdot (p_n - p_{cl} \mp p_{uc0})}} \quad \text{– коэффициенты линеаризации.}$$

С другой стороны, расход через основной золотник равен:

$$Q_3 = F_{uc} \cdot \frac{dz}{dt} + \frac{V_{uc}}{2B_{жс}} \cdot \frac{dp_{uc}}{dt} \quad (1.23)$$

Приравняв уравнения (1.10) и (1.11), после несложных преобразований, получим:

$$\frac{V_{uc}}{2B_{жс} \cdot k_{Qp}} \cdot \frac{dp_{uc}}{dt} + p_{uc} = \frac{k_{Qx_3}}{k_{Qp}} \cdot x_3 + p_n - \frac{F_{uc}}{k_{Qp}} \cdot \frac{dz}{dt} \quad (1.24)$$

где  $V_{uc}$  – объём полости силового гидроцилиндра;  $x_3$  – перемещение плунжера основного золотника пропорционального распределителя.

Структурная схема пропорционального распределителя и электронного усилителя представлена на рис. 1.27.

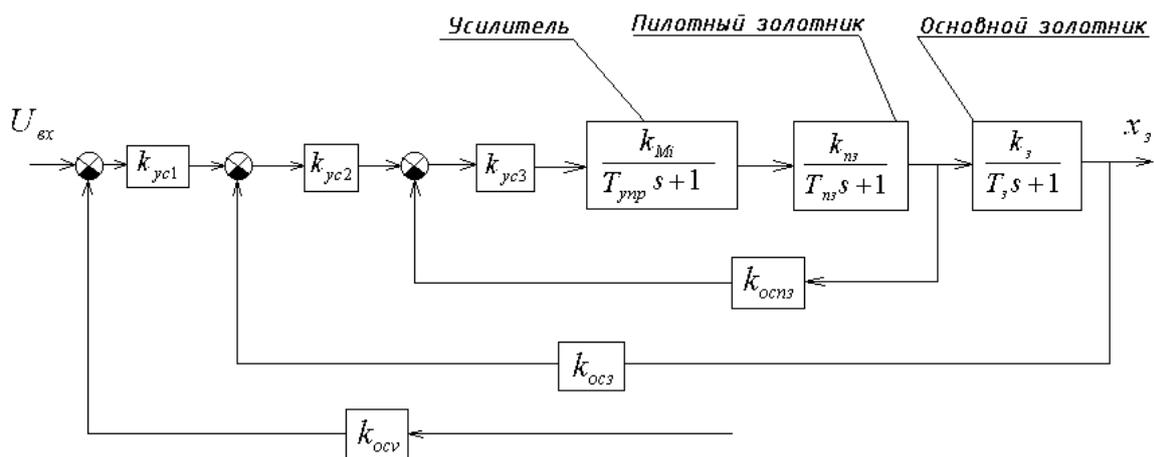


Рис. 1.27. Структурная схема пропорционального распределителя и электронного усилителя. (В этой схеме  $k_{оспз}$  – коэффициент обратной связи по положению пилотного золотника;  $k_{осз}$  – коэффициент обратной связи по положению основного золотника;  $k_{осв}$  – коэффициент обратной связи по скорости основного золотника)

Гидродинамические силы  $R_{гд.з}$ , действующие на золотник, определяются по формулам как для «гидродинамической пружины».

Уравнение движения поршня силового гидроцилиндра имеет вид:

$$m_{ц.с} \frac{d^2 z_{ц.с}}{dt^2} + k_{тр.ц.с} \frac{dz_{ц.с}}{dt} + P^{\Sigma}_{нагр} = p_{пор} F_{порш} - p_{шт} F_{шт}, \quad (1.25)$$

где  $m_{ц.с}$  – приведенная масса поршня и подвижных элементов силового гидроцилиндра;  $k_{тр.ц.с}$  – коэффициент вязкого трения в гидроцилиндре;  $P^{\Sigma}_{нагр}$  – внешняя нагрузка, действующая на шток;  $F_{шт}$  – площадь, равная площади поршня гидроцилиндра за вычетом площади штока;  $F_{порш}$  – площадь поршня;  $p_{пор}$  – давление в поршневой полости гидроцилиндра,  $p_{шт}$  – давление в штоковой полости гидроцилиндра,  $z_{ц.с}$  – перемещение поршня силового цилиндра. Отсюда передаточная функция давления в силовом гидроцилиндре:

$$W_{p1}(s) = \frac{1}{T_{рцс}s + 1} \quad (1.26)$$

где  $T_{рцс} = \frac{V_{цс}}{2 \cdot B_{жс} \cdot k_{qp}}$  – постоянная времени передаточной функции давления силового гидроцилиндра. Структурная схема исполнительного гидроцилиндра представлена на рис. 1.28.

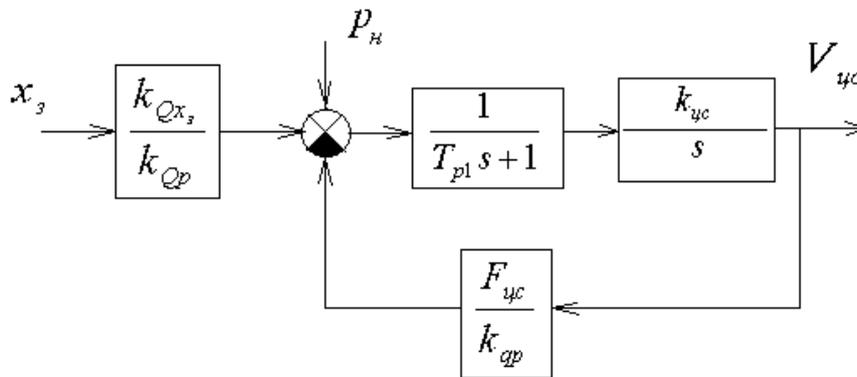


Рис. 1.28. Структурная схема исполнительного гидроцилиндра.

При расчетах характеристик медленной гидравлической шагающей машины учитывалась весомость звеньев ног и не учитывалась их инерционность, следовательно, уравнение движения силового гидроцилиндра запишем в виде:

$$k_{мрцс} \cdot \frac{dz}{dt} = p_{цс} \cdot F_{цс} \quad (1.27)$$

После преобразования уравнения получим:

$$\frac{dz}{dt} = P_{yc} \cdot \frac{F_{yc}}{k_{mpyc}} \quad (1.28)$$

Передаточная функция для силового цилиндра:

$$W(s) = \frac{k_{yc}}{s} \quad (1.29)$$

где  $k_{yc} = \frac{F_{yc}}{k_{mpyc}}$  – коэффициент усиления передаточной функции гидроцилиндра;

Уравнение движения плунжера LS-золотника

$$m_{3.ls} \frac{d^2 x_{3.ls}}{dt^2} + k_{mp.ls} \frac{dx_{3.ls}}{dt} + c_{np.ls} x_{3.ls} + P_{n.ls} + R_{з\partial.ls} = (p_n - p_{y.c}) F_{3.ls}, \quad (1.30)$$

где  $m_{3.ls}$  – масса плунжера LS-золотника;  $x_{3.ls}$  – перемещение плунжера LS-золотника;  $k_{mp.ls}$  – коэффициент, учитывающий вязкое трение в зазоре между плунжером золотника и корпусом;  $c_{np.ls}$  – жесткость пружины;  $P_{n.ls}$  – сила предварительного поджатия пружины;  $R_{з\partial.ls}$  – гидродинамическая сила, действующая на золотник со стороны рабочей жидкости;  $F_{3.ls}$  – площадь торца плунжера золотника.

Пренебрегая массой золотника, получим передаточную функцию LS-золотника:

$$W(s) = \frac{k_{ls}}{T_{ls}s + 1} \quad (1.31)$$

Здесь  $T_{ls} = \frac{k_{mpls}}{c_{ls}}$  – постоянная времени передаточной функции LS-золотника.

Уравнение расхода рабочей жидкости через LS – золотник:

$$Q_{ls} = \mu_{ls} \pi d_{ls} k_n x_{ls} \sqrt{\frac{2}{\rho}} \sqrt{|p_n - p_{y2}|} \cdot \text{sign}(p_n - p_{y2}) \quad (1.32)$$

где  $\mu_{ls}$  – коэффициент расхода окна LS-золотника;  $x_{ls}$  – перемещение LS – золотника;  $p_n$  – давление насоса;  $p_{цу2}$  – давление в ЦУ-2;  $d_{ls}$  – диаметр плунжера LS-золотника.

Расход жидкости через канал в ЦУ-2

$$Q_{кан} = \mu_{кан} \cdot \pi \cdot d_{цу2} \cdot \delta \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{|p_{цу2} - p_{сл}|} \cdot \text{sign}(p_{цу2} - p_{сл}) \quad (1.33)$$

где  $\mu_{кан}$  – коэффициент расхода канала в ЦУ-2;  $d_{цу2}$  – диаметр ЦУ-2;  $\delta$  – зазор между гильзой гидроцилиндра и поршнем;  $\rho$  – плотность рабочей жидкости.

Уравнение баланса расхода через LS-золотника:

$$Q_{ls} = Q_{цу2} = F_{цу2} \cdot \frac{dy}{dt} + \frac{V_{цу2}}{B_{жс}} \cdot \frac{dp_{цу2}}{dt} + Q_{кан} \quad (1.34)$$

После преобразований получим:

$$\frac{dp_{цу2}}{dt} = \frac{B_{жс}}{V_{цу2}} \left( Q_{ls} - F_{цу2} \frac{dy}{dt} - Q_{кан} \right) \quad (1.35)$$

Структурная схема LS-золотника представлена на рис. 1.29.

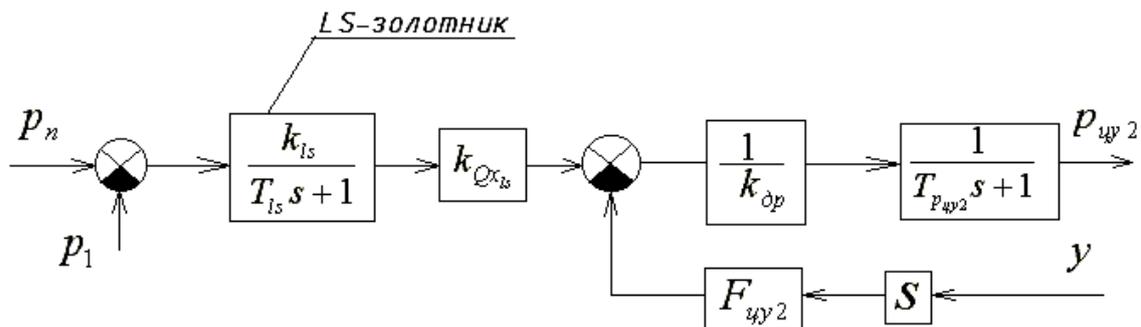


Рис. 1.29. Структурная схема LS-золотника.

Уравнение движения поршней гидроцилиндров управления (ЦУ №1,2 на рис. 1.16):

$$m_{ц.у} \frac{d^2 y}{dt^2} + k_{мп} \frac{dy}{dt} + c_{пр.ц.у1} y + P_{н.н1} + (p_{ц.у1} - p_{сл}) F_{ц.у1} + P_{м.с} = (p_{ц.у2} - p_{сл}) F_{ц.у2}, \quad (1.36)$$

где  $m_{ц.у}$  – суммарная масса поршней цилиндров управления и присоединенных частей;  $k_{мп}$  – коэффициент вязкого трения;  $c_{пр.ц.у1}$  – жёсткость пружины;  $P_{н.н1}$  – сила предварительного поджатия пружины;  $F_{ц.у1}, F_{ц.у2}$  – площади поршней цилиндров управления;  $P_{м.с}$  – суммарный момент сопротивления, приложенный к органу регулирования насоса,  $y$  – перемещение поршней гидроцилиндров управления.

Пренебрегая массой поршней гидроцилиндра управления, получим уравнение движения гидроцилиндра управления:

$$k_{мпц.у2} \frac{dy}{dt} + c_{прц.у1} y = p_{ц.у2} \cdot F_{ц.у2} - p_{н} \cdot F_{ц.у1}. \quad (1.37)$$

Из (1.37) следует, что передаточная функция цилиндра управления:

$$W(s) = \frac{k_{ц.у2}}{T_{ц.у2} \cdot s + 1} \quad (1.38)$$

Здесь  $k_{ц.у2} = \frac{1}{c_{прц.у1}}$  – коэффициент усиления передаточной функции цилиндра управления.

Уравнение расходов в напорной гидролинии, исходя из условия неразрывности течения жидкости, можно записать в виде:

$$\frac{V_{цс}}{B_{жс}} \cdot \frac{dp_{н}}{dt} = (Q_{н} - Q_{с} - Q_{у}) \quad (1.39)$$

где  $Q_{н} = \frac{V_0}{2\pi} \cdot \omega_{н} \cdot \frac{\gamma}{\gamma_{\max}}$  – подача насоса,  $V_0$  – рабочий объём насоса,  $\omega_{н}$  – угловая скорость вала насоса,  $\gamma$  – угол наклона шайбы насоса;  $\gamma_{\max}$  – максимальный угол наклона шайбы насоса.

$Q_{с} = F_{цс} \frac{dz}{dt}$  – расход жидкости, подводимой к гидроприводам;

$Q_{у} = Q_{лс}$  – расход, идущий на управление.

Структурная схема гидропривода поворота стойки состоит из передаточных функций всех элементов, входящих в него (рис. 1.30). Структурная схема гидроприводов привода бедра и голени имеет аналогичный вид. Отличия лишь в значениях коэффициентов обратных связей, коэффициентов усиления, диаметров поршней. Структурная схема гидроприводов ноги (поворота стойки, голени и бедра) состоит из трех блоков аналогичных, представленным на рис. 1.30.

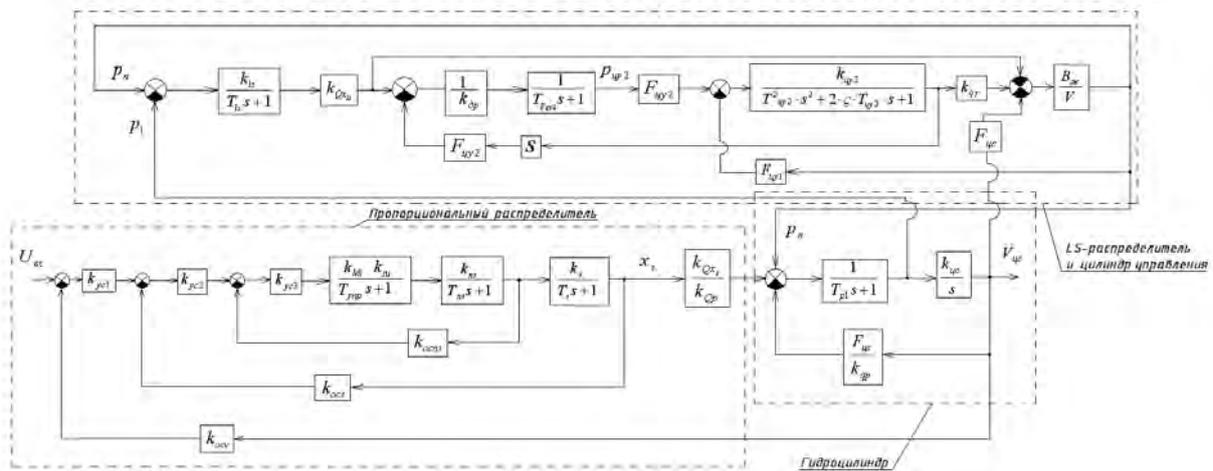


Рис. 1.30. Структурная схема гидропривода поворота стойки ноги ШМ.

### Исследование динамики гидропривода с LS-регулированием

Simulink – это одна из прикладных программ Matlab, которая предназначена для моделирования динамических систем. Программа обладает графическим интерфейсом, которая позволяет формировать структурные схемы систем любой степени сложности. Simulink включает в себя библиотеку различных линейных и нелинейных компонентов, источников входных сигналов, имеет возможность создания собственных блоков. Построение схемы исследуемой системы производится при помощи копирования или перемещения блоков из библиотеки в рабочее окно. Свойства каждого блока задаются при помощи меню Block Parameters, в котором можно задать все необходимые параметры блока. Для просмотра результатов моделирования используются несколько блоков, которые позволяют наблюдать за процессом моделирования в режиме реального времени. В системе “Matlab+Simulink” можно строить характеристики, используемые в теории автоматического управления, например амплитудные фазовые частотные характеристики (диаграммы Боде), диаграмма Найквиста, переходные процессы и. т.д.

Структурная схема гидропривода одной ноги ШМ, состоящая из передаточных функций, входящих в неё элементов, набранная в системе “Matlab+Simulink” представлена на рис. 1.31.

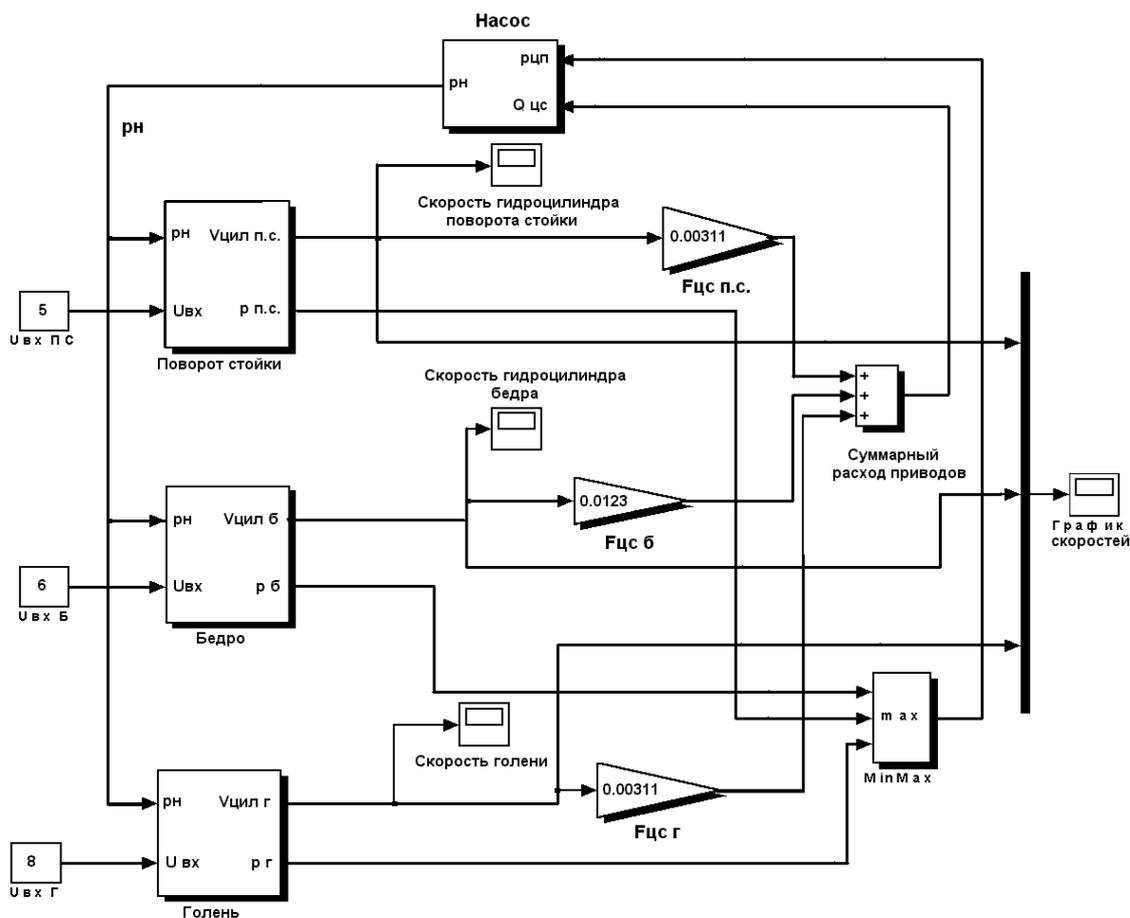


Рис. 1.31. Структурная схема гидроприводов ноги ШМ, набранная в системе MATLAB.

Элементы гидроприводов ноги объединены в макроблоки, каждый из которых имеет соответствующее название («Поворот стойки», «Бедро», «Голень»). Отдельно расположен макроблок, объединяющий элементы, входящие в насос (LS- распределитель, цилиндры управления). Каждый из макроблоков гидроприводов, в свою очередь, содержит макроблоки, описывающие работу пропорционального распределителя и гидроцилиндра. На рис. 1.32 показано содержимое макроблока для привода поворота стойки. Содержимое каждого макроблока можно посмотреть, сделав двойной щелчок левой кнопки мыши на его изображении.

Работа пропорционального распределителя, включая электронный усилитель, описывается передаточными функциями элементов, входящих в него (рис. 1.30). Содержание макроблока «Гидроцилиндр» включает в себя передаточные функции, описывающие гидроцилиндр (рис. 1.34).

В системе «Matlab+Simulink» имеется возможность ввода нелинейностей различного вида, которые позволяют описать процессы неподдающиеся линеаризации. В модели насоса используются нелинейности, которые ограничивают выходную величину. Такие блоки имитируют ограничение перемещения цилиндров управления, смещение LS-золотника.

Блок Fsp позволяет выполнять математические операции над сигналами. Этот блок использовался для ввода нелинейных уравнений, описывающих расходы жидкости через LS-золотник и канал в ЦУ №2 (рис. 1.35). Содержание макроблока «Насос» включает в себя передаточные функции, описывающие функционирование насоса (рис. 1.36).

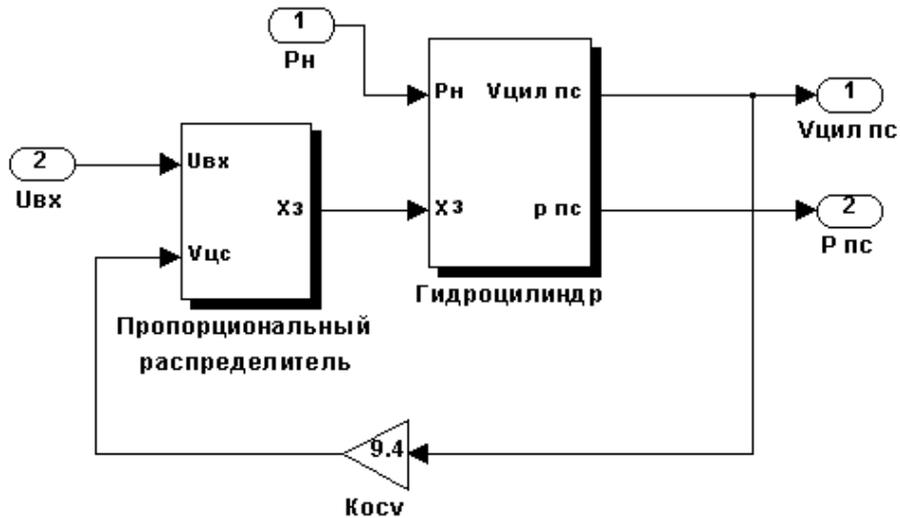


Рис. 1.32. Макроблок для гидропривода поворота ноги.

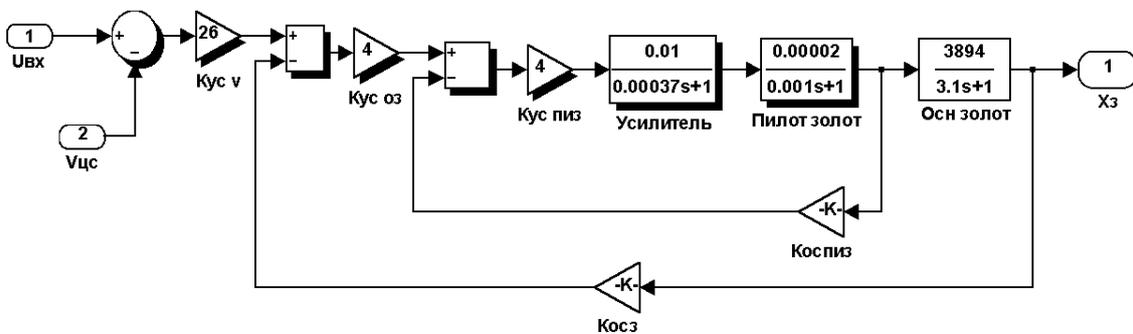


Рис. 1.33. Макроблок для пропорционального распределителя.

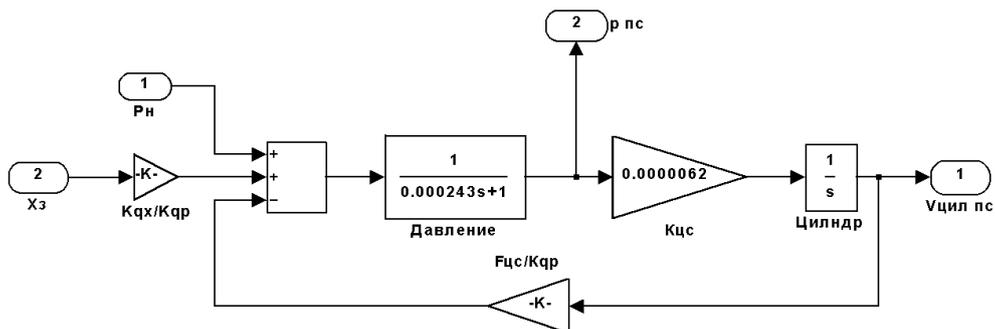


Рис. 1.34. Макроблок для гидроцилиндра

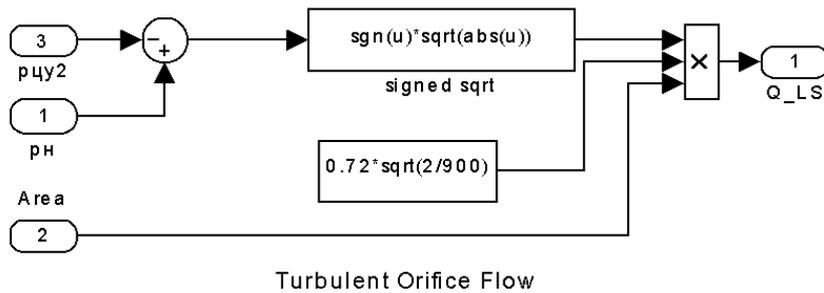


Рис. 1.35. Макроблок, описывающий уравнения расхода жидкости через LS-золотник и канал ЦУ-2

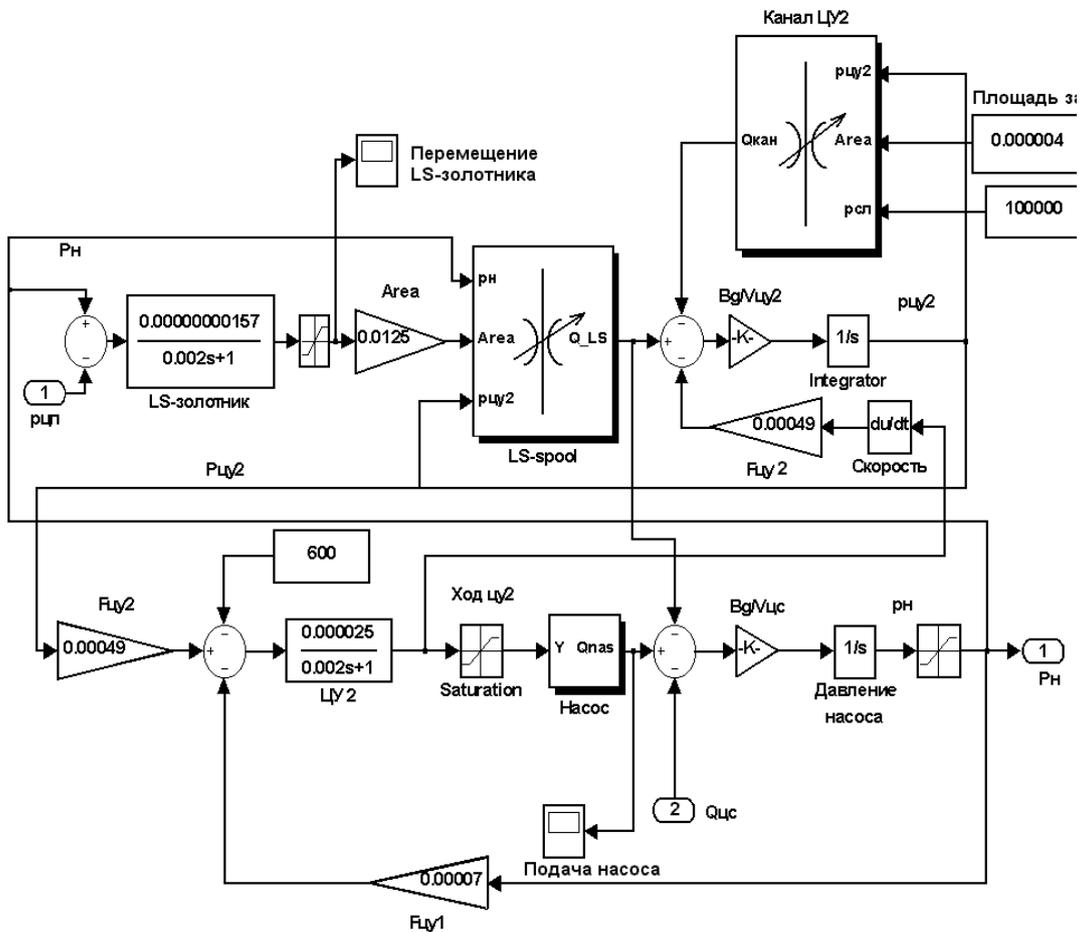


Рис. 1.36. Макроблок для насоса

### Результаты моделирования

На рис.1.37 представлены переходные процессы движений поршней гидроцилиндров в гидроприводах ноги (поворота стойки, голени и бедра). Вид переходных процессов показывает, что замкнутая система гидропривода одной ноги устойчива. Время переходных процессов равно времени переходных процессов, полученных в результате моделирования нелинейной системы [12].

Важной динамической характеристикой систем автоматического управления являются частотные характеристики, достоинство которых состоит в том, что частотные характеристики позволяют просто выявлять влияние того или иного параметра на динамические свойства системы (устойчивость, переходной процесс и т.д.). Частотные характеристики звеньев и систем строятся на основании их комплексных передаточных функций. Для анализа устойчивости исследуемой гидравлической системы управления применяются логарифмические амплитудные фазовые частотные характеристики (ЛАФЧХ) разомкнутой цепи. Для этого необходимо разорвать обратную связь по скорости гидроцилиндра, обозначить точки ввода, “input point”, и точку вывода, “output point”, которые необходимы для получения частотных характеристик.

ЛАЧХ и ЛФЧХ для гидропривода каждой степени свободы ноги представлены на рис 1.25-1.27 (1.38 - 1.40). Запасы по частоте и амплитуде должны быть не менее определенных значений. Рекомендуемые запасы по амплитуде – 6-8 дБ, по фазе –  $40^\circ$ . Для гидропривода поворота стойки (рис. 1.38) запас по амплитуде составляет 9.75 дБ, запас по фазе  $36^\circ$ , что является вполне достаточным. Гидропривод голени ноги (рис. 1.39): запас по амплитуде равен 10 дБ, запас по фазе –  $40^\circ$ . Частотные характеристики гидропривода бедра (рис. 1.40) имеют запас по амплитуде 8 дБ, запас по фазе  $35^\circ$ . Ещё одним способом проверки устойчивости системы является критерий Найквиста, который можно сформулировать следующим образом: «Если разомкнутая цепь системы устойчива, то для устойчивости замкнутой системы необходимо и достаточно, чтобы амплитудно-фазовая частотная характеристика разомкнутой цепи не охватывала точку (-1). На рис. 1.40-1.43 изображены диаграммы Найквиста для гидроприводов соответственно голени, бедра и поворота стойки. Как видно из графиков ни один из них не охватывает точку (-1), которая на каждом из графиков обозначена крестиком. Из диаграмм следует, что гидроприводы устойчивы.

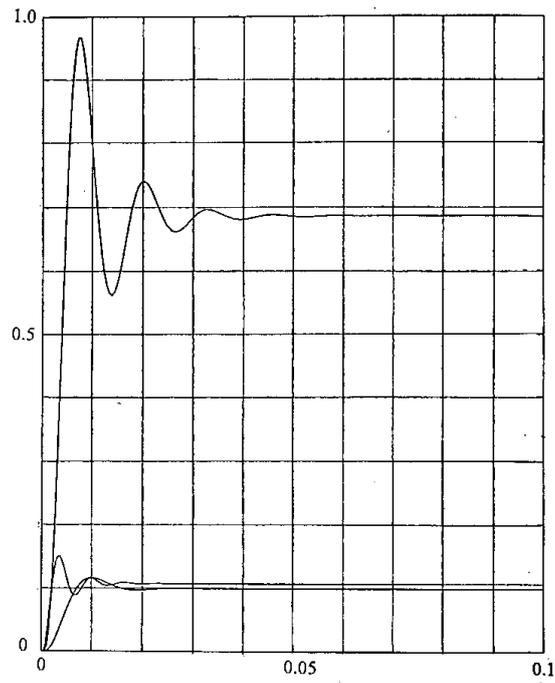


Рис. 1.37. Переходные процессы в гидроприводах ноги ШМ.

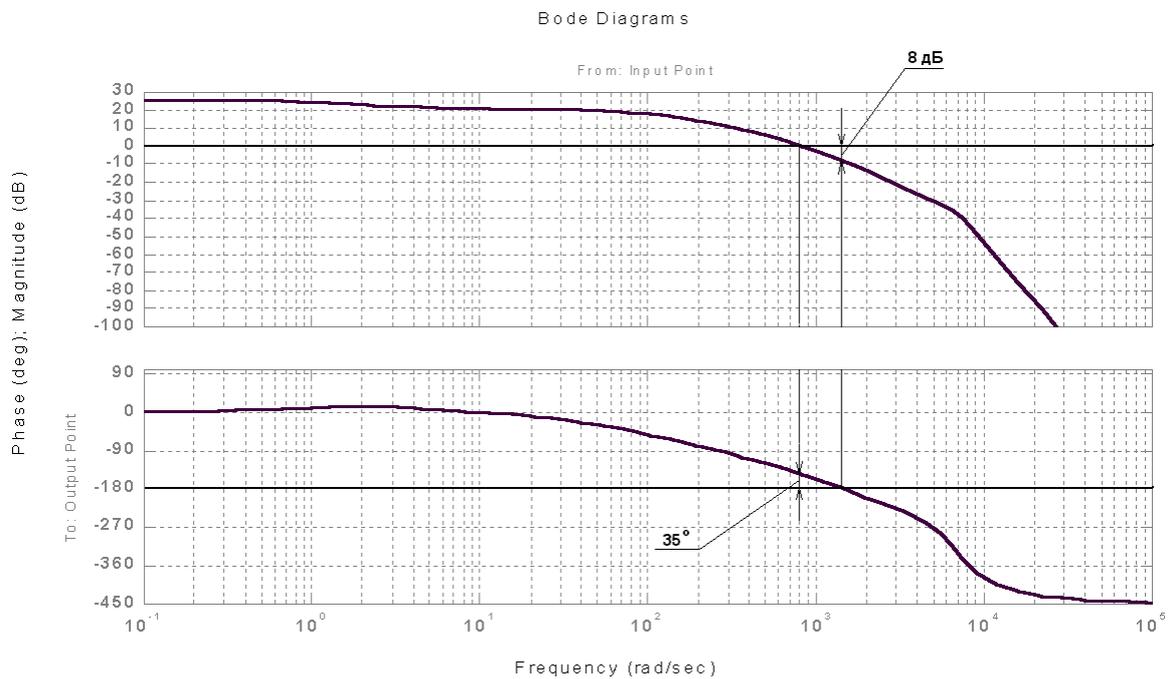


Рис. 1.38. ЛАФЧХ для гидропривода поворота стойки ноги.

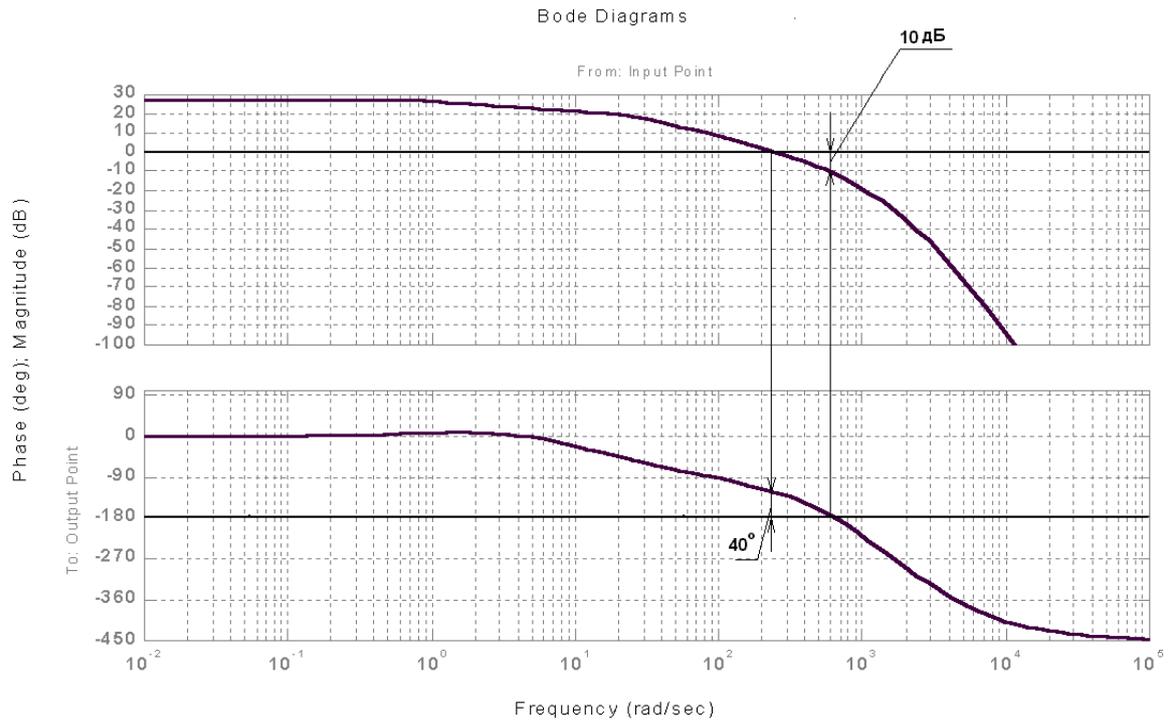


Рис. 1.39. ЛАФЧХ для гидропривода бедра ноги.

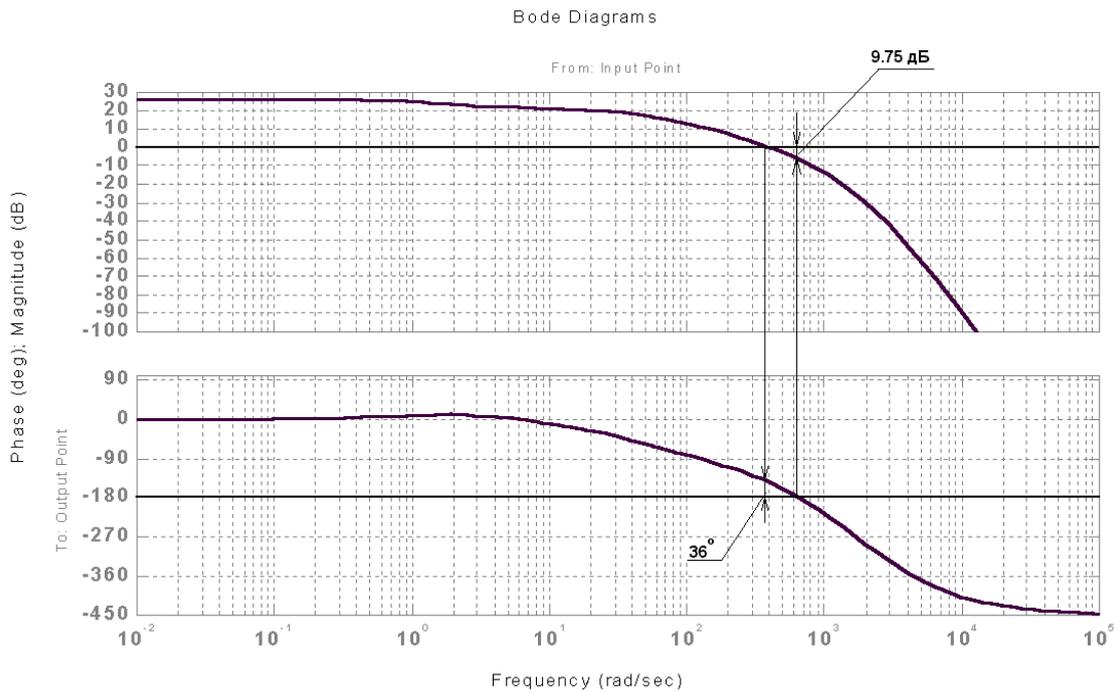


Рис. 1.40. ЛАФЧХ для гидропривода голени ноги.

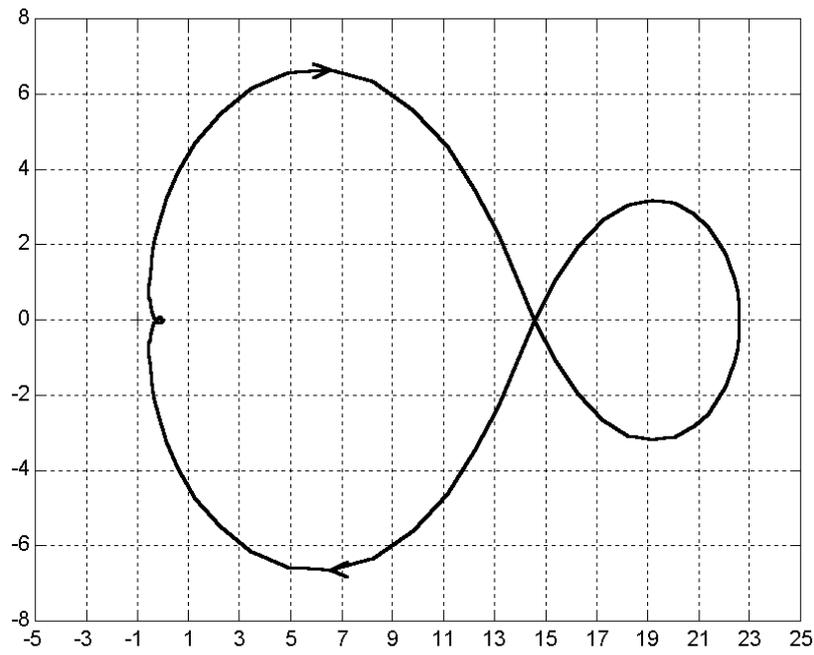


Рис. 1.41. Диаграмма Найквиста для привода голени.

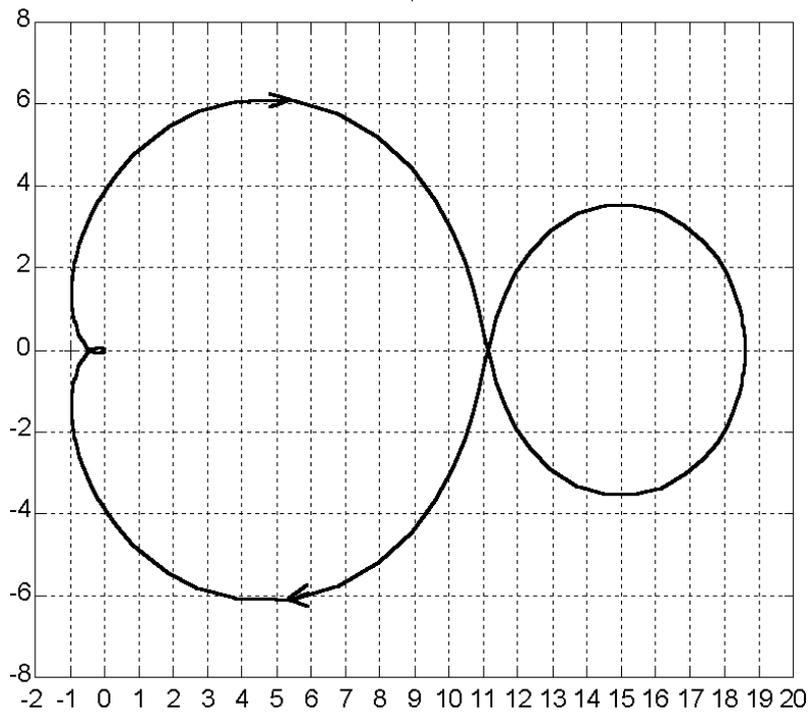


Рис. 1.42. Диаграмма Найквиста для привода бедра.

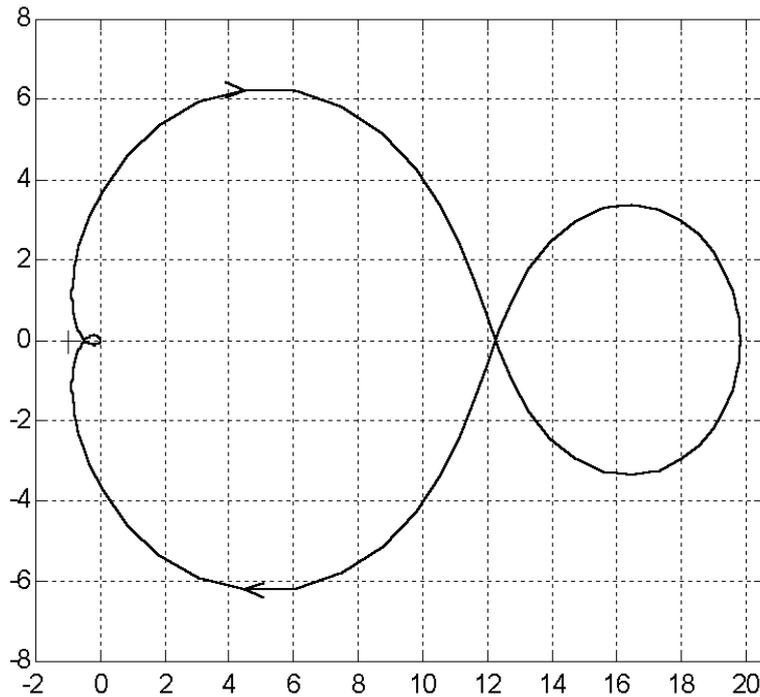


Рис. 1.43. Диаграмма Найквиста для привода поворота стойки.

### 1.3. Гидравлическая система управления экзоскелетона и результаты математического моделирования

Экзоскелетон – это устройство, которое одевается на человека и предназначено для расширения его возможностей. В частности, для увеличения усилий, которые может развивать человек (рис. 1.39). Применение этого устройства может быть полезно при выполнении тяжелых работ, погрузочно-разгрузочных работ в местах, где нет возможности применения колёсных погрузчиков, при ремонтных работах на крупных объектах, при аварийно-спасательных операциях. Кроме того, экзоскелетон может служить для реабилитации людей, получивших серьёзные травмы позвоночника. Экзоскелетон может применяться и в военных целях.

В качестве привода звеньев экзоскелетона целесообразно применить гидропривод, так как гидропривод по сравнению с электро- и пневмоприводами имеет ряд преимуществ. Главными из них являются высокое быстродействие и высокая удельная мощность, что позволяет создать более компактный привод.

В качестве источника энергии может служить двигатель внутреннего сгорания.

Внешний вид экзоскелетона представлен на рис. 1.39. Число активных степеней подвижности на ногах и руках экзоскелетона равно 10-ти. Кинематическая схема экзоскелетона разработана в Robotics Research Centre of Nanyang Technological University.

Принципиальная гидравлическая схема экзоскелетона представлена на рис. 1.40. Она включает в себя десять исполнительных гидродвигателей, по

числу активных степеней подвижности, насосную станцию и управляющие гидроаппараты.

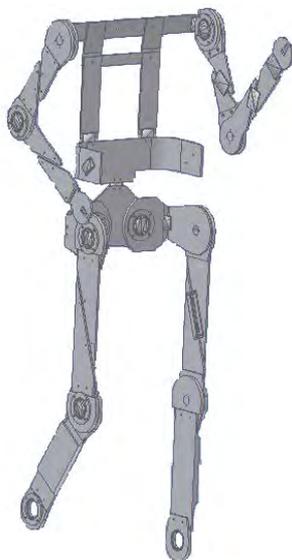
В настоящее время гидропривод, разработанный в Robotics Research Centre, в качестве исполнительных гидродвигателей использует поворотные гидродвигатели, которые монтируются в суставах экзоскелетона.

Весовые характеристики этого гидропривода и элементов экзоскелетона приведены в таблице 1.1.

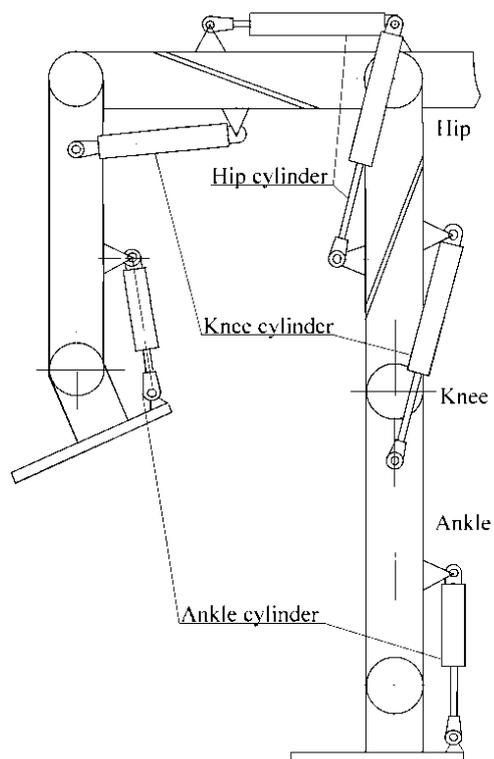
В данной работе рассматривается вариант, в котором в качестве гидродвигателей используются гидроцилиндры, так как это более соответствует физиологическому строению мускулатуры человека. Применение гидроцилиндров в качестве гидродвигателей может привести к созданию более компактной схемы и с лучшими весовыми характеристиками, так как поворотные гидродвигатели имеют достаточно большие габариты и массу и небольшие крутящие моменты, меньшую надёжность и ресурс по сравнению с гидроцилиндрами.

Окончательное решение о применении поворотных гидродвигателей или гидроцилиндров может быть принято после макетного изготовления экзоскелетона и его испытаний.

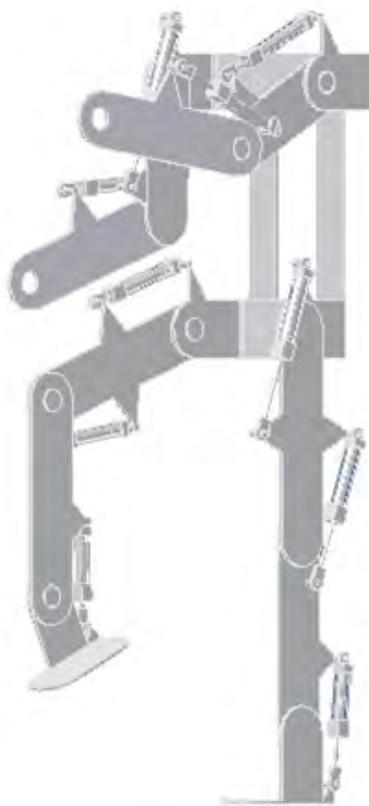
Кинематические схемы ноги и плечевого сустава с применением гидроцилиндров показаны на рис. 1.40, 1.41. На рисунке показаны положения ног и рук в крайних положениях и соответствующие им положения гидроцилиндров.



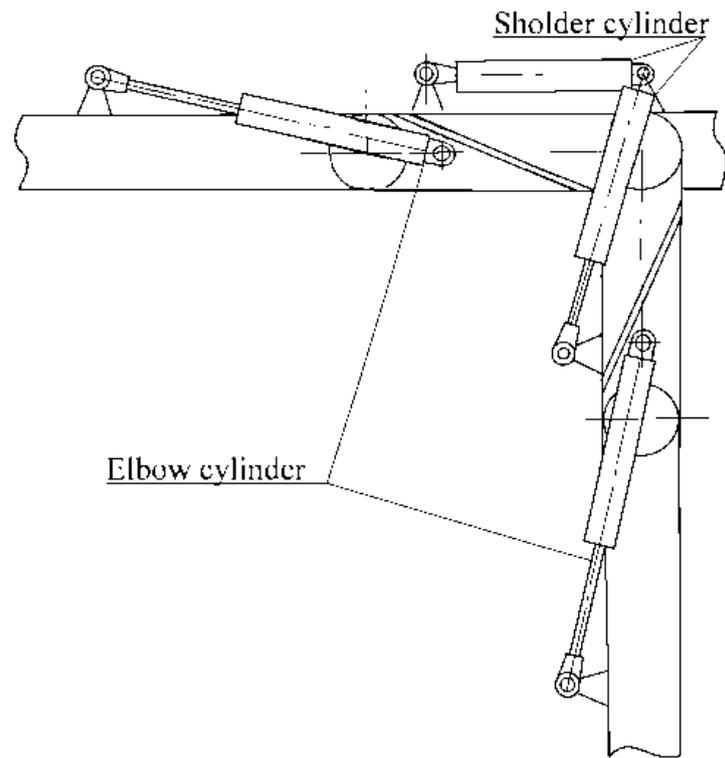
*Рис. 1.39.* Кинематическая схема экзоскелетона.



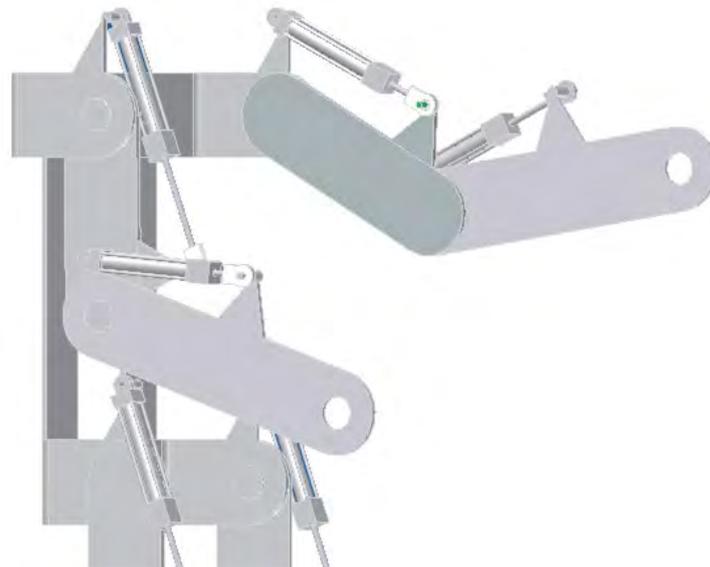
*Рис. 1.40.* Принципиальная кинематическая схема с использованием гидроцилиндров.



*Рис. 1.40а.* Кинематическая схема экзоскелетона с использованием гидроцилиндров.



*Рис. 1.41.* Кинематическая схема плеча и руки с применением гидроцилиндров.



*Рис. 1.41a.* Кинематическая схема рук и плеч экзоскелетона.

Таблица 1.1.

	Torque for actuator weight estimation (Nm)	Unit weight (kg)	No. of parts	
<b>Weight of actuator @ shoulder</b>	296.1	10.00	2	20.00
<b>Weight of link from shoulder to elbow</b>	–	4.29	2	8.58
<b>Weight of actuator @ elbow</b>	164.4	6.00	2	12.00
<b>Weight of link from elbow to wrist</b>	–	2.16	2	4.32
Weight of actuator @ wrist	–	3.00	2	6.00
Weight of hand	–	3.00	2	6.00
Weight of supporting bracket @ back	–	5.61	1	5.61
Weight of components @ back	–	20.00	1	20.00
Weight of actuator @ hip	403.8	15.00	2	30.00
Weight of link from hip to knee cap (thigh)	–	4.67	2	9.34
Weight of actuator @ knee	370.1	12.00	2	24.00
Weight of link from knee cap to ankle (calf)	–	4.39	2	8.78
Weight of actuator @ ankle	370.1	12.00	2	24.00
Weight of foot	–	2.50	2	5.00
Weight of exoskeleton (kg)				<b>183.63</b>

**Насосная станция.** Насосная станция включает в себя насос с регулятором давления. Регулятор давления служит для поддержания постоянного давления нагнетания насоса, при этом насос поставляет только то количество жидкости, которое необходимо гидроприводам. Поэтому данный насос имеет достаточно хорошие энергетические характеристики (рис. 1.46). Давление же будет постоянным и определяется характеристиками пружины распределителя ограничителя давления.

На рис. 1.42 представлена принципиальная схема насоса с регулятором давления фирмы Rexroth A10 VSODR

Для улучшения габаритных характеристик и компоновки гидропривода, а также с целью улучшения теплового баланса, целесообразно разместить насос в баке с рабочей жидкостью (рис. 1.48).

В качестве регулирующей аппаратуры применяются электрогидравлические усилители (ЭГУ) типа сопло-заслонка (серво клапан), которые широко распространены в робототехнике и авиации. Они имеют малые размеры приблизительно 80x46x80 мм и массу ~ 0.64 кг, высокие динамические характеристики. Наиболее широкое распространение в технике получили электрогидравлические усилители с механической обратной связью (рис. 1.49).

Электрогидравлический усилитель представляет собой совокупность электромеханического преобразователя (ЭМП) 1, гидравлического усилителя типа сопло-заслонка 2 и золотника, который соединён с заслонкой при помощи

упругого стержня. Сопла и заслонки вместе представляют собой регулируемые дроссели, сопротивление которых может изменяться при помощи изменения расстояния между соплами и заслонкой.

Электрический сигнал подаётся на обмотки ЭМП, под действием электромагнитной силы, сердечник 6 поворачивается на определённый угол, который зависит от уровня входного напряжения. Заслонка 7, скреплённая с сердечником также поворачивается, при этом изменяется расстояние между заслонкой и соплами. При этом изменяются сопротивления регулируемых дросселей, что приводит к изменению перепада давлений на торцах золотника 3. Под действием это перепада золотник начинает смещаться от нейтрального положения, тем самым, соединяя линии нагнетания и слива с полостями гидродвигателя. Золотник смещается до тех пор, пока момент от стержня не станет равным электромагнитному моменту ЭМП. Смещение золотника, а следовательно, и поток рабочей жидкости через золотник пропорциональны входному электрическому сигналу. Расчетная схема ЭГУ приведена на рис. 1.50.

Поскольку каждое звено экзоскелетона осуществляет независимое движение, то каждый гидродвигатель должен иметь независимый алгоритм управления движением. Поэтому для управления необходимо использовать 10 ЭГУ, по числу активных степеней подвижности. Конструктивная схема ЭГУ фирмы Rexroth приведена на рис. 1.49.

Гидравлическая схема экзоскелетона приведена на рис. 1.42. Она включает в себя насос с регулятором давления, исполнительные гидродвигатели, электрогидравлические усилители.

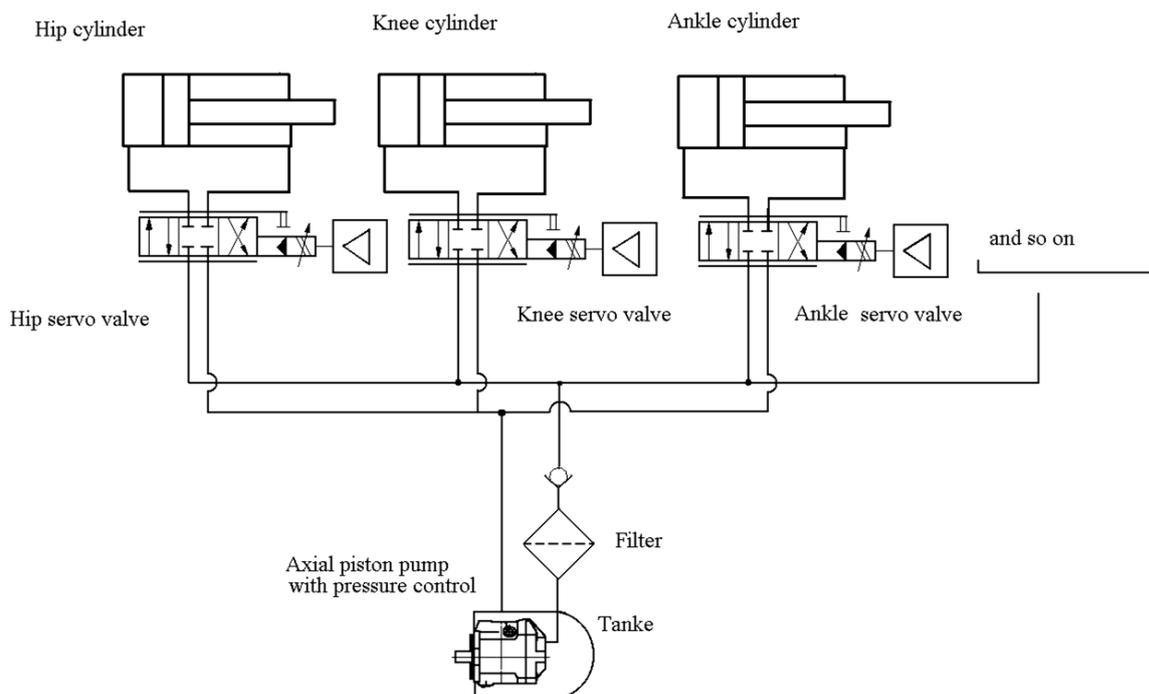


Рис. 1.42. Принципиальная гидравлическая схема экзоскелетона.

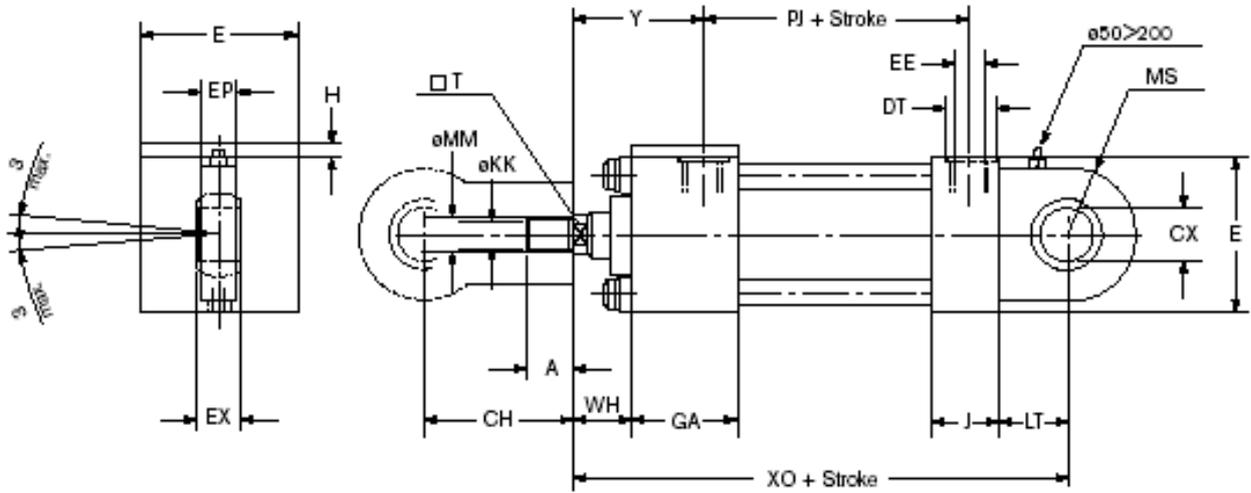


Рис. 1.43. Гидроцилиндр фирмы Bosch Rexroth.

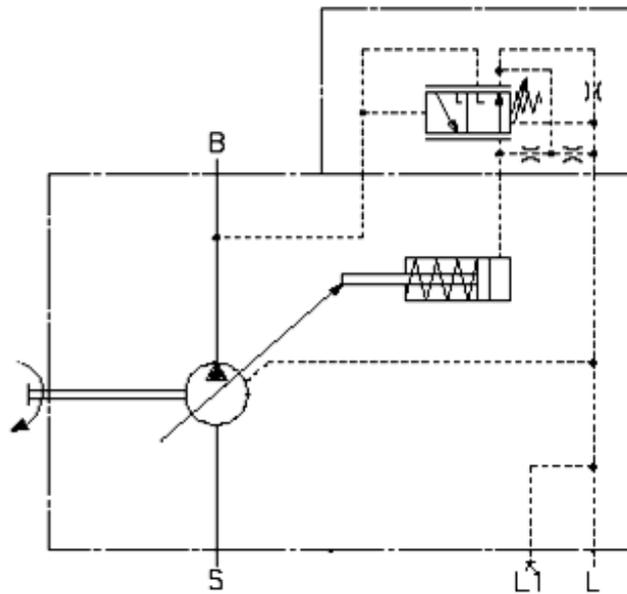


Рис. 1.44. Принципиальная гидравлическая схема насоса с ограничителем давления.

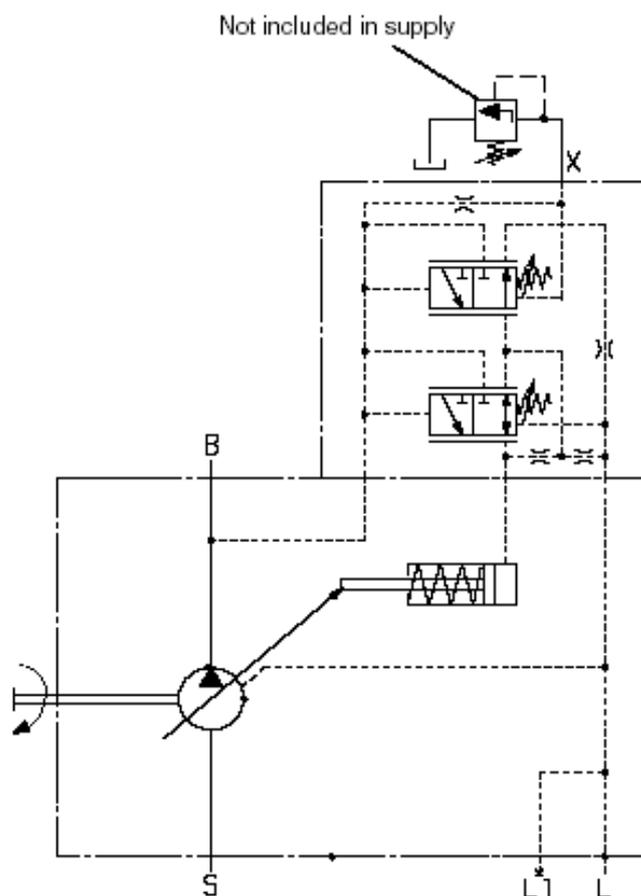


Рис. 1.45. Принципиальная гидравлическая схема насоса с ограничителем давления с дистанционным управлением.

Масса элементов гидропривода экзоскелетона приведены в таблице 1.2.

Таблица 1.2

Actuator name	Weight of Cylinder, kg	Weight of servo-valve and manifold, kg	Weight of pump and tank, kg	Actuator weight of exoskeleton
Knee	2.02	2.64	—	
Hip	2.06	2.64	—	
Ankle	1.71	2.64	—	
Shoulder	1.91	2.64	—	
Elbow	2.132	2.64	—	
Overall weight	$9.832 \times 2 = 19.664$	$13.2 \times 2 = 26.4$	$8 + 10 = 18$	64.1

Общая масса гидропривода экзоскелетона без учёта массы трубопроводов равна ~64.1 кг, что на 46 кг меньше чем при использовании поворотных гидродвигателей.

**Математическая модель электрогидравлического усилителя**

Уравнение напряжений в электрической цепи электромагнита:

$$U_{yc} = R_y \cdot I_y + L_y \cdot \frac{dI_y}{dt} + C_l \cdot \frac{d\varphi_y}{dt} \quad (1.40)$$

где  $U_{yc}$  – напряжение на выходе электронного усилителя;  $R_y$  – активное сопротивление обмотки управления электромагнита;  $L_y$  – индуктивность обмотки управления электромагнита;  $C_l$  – коэффициент противо-ЭДС;  $I_y$  – ток в обмотке управления;  $\varphi_y$  – смещение пилотного золотника и жёстко связанного с ним якоря электромагнита.

Значение коэффициента противо-ЭДС определяют по соотношению

$$C_l = L_y \cdot \frac{l_y^2}{\delta^2} \cdot I_y \quad (1.41)$$

где  $l_y$  – длина средней линии якоря;  $\delta = 3 \cdot \delta_0$ ;  $\delta_0$  – воздушный зазор между якорем и сердечником.

Уравнение движения плунжера пилотного золотника:

$$J_y \frac{d^2\varphi_y}{dt^2} + k_{mp} \frac{d\varphi_y}{dt} + c_{эмп} \varphi_y + M_{z\partial 1} = P_{эзм} \quad (1.42)$$

где  $J_y$  – момент инерции якоря электромагнита;  $k_{mp}$  – коэффициент, учитывающий вязкое демпфирование якоря;  $M_{z\partial 1}$  – момент гидродинамической силы  $P_{z\partial}$ ;  $P_{эзм}$  – сила электромагнита;  $c_{эмп}$  – жесткость пружины подвески заслонки,  $P_{z\partial} = (1,03 - 1,06) \cdot (p_{y1} - p_{y2}) \cdot F_c$ .

$M_n = P_{гд} \cdot l + P_{o.c.} \cdot l_{o.c.}$ , где  $P_{o.c.}$  – сила, изгибающая стержень при повороте заслонки и перемещении золотника.

$P_{o.c.} = \frac{3 \cdot E_{ст} \cdot J_{ст}}{l_{ст}^3} \cdot z_{ст}$ , где  $E_{ст}$  – модуль упругости материала, из которого изготовлен стержень,  $J_{ст}$  – момент инерции сечения стержня относительно главной центральной оси, перпендикулярной к плоскости изгибающей силы,  $z_{ст}$  – прогиб конца стержня, равен сумме прогибов, вызванных поворотом заслонки относительно центра О на угол  $\varphi_y$  и перемещением  $x_3$  золотника.

$$z_{ст} = l_{o.c.} \cdot \varphi_y + x_3.$$

Уравнение расходов рабочей жидкости в управляющем каскаде:

$$\begin{aligned} Q_{y1} &= F_3 \frac{dx_3}{dt} + k_{сжс1} \frac{dp_{y1}}{dt} \\ Q_{y2} &= F_3 \frac{dx_3}{dt} + k_{сжс2} \frac{dp_{y2}}{dt} \end{aligned} \quad (1.43)$$

где  $Q_{yi}$  – расходы рабочей жидкости, обеспечивающие управление основным золотником;  $F_3$  – площадь торца плунжера основного золотника;  $x_3$  – перемещение плунжера основного золотника;  $k_{сжсi} = \frac{V_{yi}}{B_{ж}}$  – коэффициент, учитывающий сжимаемость рабочей жидкости в торцевых камерах основного золотника;  $V_{yi}$  – объём торцевой камеры золотника;  $B_{ж}$  – модуль объёмной упругости рабочей жидкости;  $p_{yi}$  – давление жидкости в камерах управления под торцами плунжера основного золотника.

Расход  $Q_{yi}$  определяется известными из гидравлики соотношениями:

$$Q_{y1} = k_{др1} \sqrt{|p_n - p_{y1}|} \cdot \text{sign}(p_n - p_{y1}) - k'_c \cdot f_1(h) \cdot \sqrt{|p_{y1} - p_{cl}|} \cdot \text{sign}(p_{y1} - p_{cl})$$

$$Q_{y2} = -k_{др2} \sqrt{|p_n - p_{y2}|} \cdot \text{sign}(p_n - p_{y2}) + k'_c \cdot f_2(h) \cdot \sqrt{|p_{y2} - p_{cl}|} \cdot \text{sign}(p_{y2} - p_{cl}),$$

где  $k_{dpi} = \mu_{dpi} \cdot f_{др} \cdot \sqrt{\frac{2}{\rho}}$  – проводимость дросселя гидроусилителя;  $p_n$  – давление насоса;  $\mu_{dpi}$  – коэффициент расхода жидкости через дроссель;  $f_{др}$  – площадь дросселя,  $k'_{ci}$  – удельная проводимость сопла при нейтральном положении заслонки,

$$f_1(h) = \left(1 + \frac{h}{h_0}\right) \cdot h_0, \quad f_2(h) = \left(1 - \frac{h}{h_0}\right) \cdot h_0, \quad f_i(h) - \text{функция, определяющая}$$

проходное сечение между торцом сопла и заслонкой,  $h = l \cdot \varphi_{я}$  – перемещение заслонки,  $h_0$  – расстояние от сопла до заслонки при нейтральном положении.

Уравнение движения золотника гидроусилителя.

$$m_3 \frac{d^2 x_3}{dt^2} + k_{мпз} \frac{dx_3}{dt} + c_{мпз} x_3 + R_{зоз} + R_{мпз} = F_3 \Delta p_y, \quad (1.44)$$

где  $m_3$  – масса золотника;  $k_{мпз}$  – коэффициент, учитывающий вязкое трение в зазоре между плунжером основного золотника и гильзой;  $c_{мпз}$  – жесткость пружины основного золотника;  $R_{зод3}$  – гидродинамическая сила, действующая на плунжер основного золотника со стороны рабочей жидкости;  $R_{мпз}$  – сила вязкого трения действующая на плунжер основного золотника;  $\Delta p_y = p_{y1} - p_{y2}$  – перепад давления рабочей жидкости на торцах плунжера основного золотника;  $F_3$  – площадь торца плунжера золотника.

Уравнение расхода жидкости через основной золотник:

$$Q_{31} = \mu_3 \pi \cdot d_3 \cdot k_{нз} \cdot x_3 \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{|p_n - p_{зод1}|} \cdot \text{sign}(p_n - p_{зод1}),$$

$$Q_{32} = \mu_3 \pi \cdot d_3 \cdot k_{нз} \cdot x_3 \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{|p_{зод2} - p_{сл}|} \cdot \text{sign}(p_{зод2} - p_{сл}),$$

где  $\mu_3$  – коэффициент расхода окна золотника;  $d_3$  – диаметр плунжера основного золотника;  $x_3$  – перемещение плунжера золотника;  $k_{нз}$  – коэффициент перекрытия гильзы золотника;  $p_n$  – давление насоса;  $p_{зодi}$  – давление в полостях гидродвигателя.

Уравнение давлений в полостях гидродвигателя.

$$\left. \begin{aligned} \frac{dp_1}{dt} &= \frac{B_{жс}}{V_{з.д}} \left( Q_{31} - F_{з.д} \cdot \frac{dz_{з.д.}}{dt} \right) \\ \frac{dp_2}{dt} &= \frac{B_{жс}}{V_{з.д}} \left( F_{ум} \cdot \frac{dz_{з.д.}}{dt} - Q_{32} \right) \end{aligned} \right\} \quad (1.45)$$

Здесь  $F_{з.д.}$  – площадь поршня гидродвигателя,  $B_{жс}$  – упругость рабочей жидкости,  $V_{з.д.}$  – объём жидкости под торцом поршня гидродвигателя,  $\frac{B_{жс}}{V_{з.д.}}$  – коэффициент сжимаемости рабочей жидкости.

Уравнение движения гидродвигателя.

$$m_{г.д.} \frac{d^2 z_{з.д.}}{dt^2} + k_{мп.зод} \frac{dz_{з.д.}}{dt} + P_{зод} = (p_1 - p_2) \cdot F_{з.д} \quad (1.46)$$

где  $m_{г.д.}$  – масса звена прикреплённого к гидроцилиндру и поршня гидроцилиндра;  $z_{з.д.}$  – перемещение поршня гидроцилиндра,  $k_{тр.гд}$  –

коэффициент трения гидродвигателя,  $p_1, p_2$  – давление в полостях гидродвигателя,  $P_{с.д.}$  – внешняя сила, приложенная к штоку гидроцилиндра.

Математическая модель насоса с регулятором давления.

Схема насоса с регулятором давления приведена на рис.1.44. Подача насоса  $Q_n$  определяется соотношением:

$$Q_n = \frac{q_n}{\gamma_{\max}} \cdot \omega_n \gamma \quad (1.47)$$

где  $q_n$  – характерный объём насоса;  $\gamma_{\max}$  – максимальный угол поворота шайбы;  $\omega_n$  – угловая скорость вращения вала насоса;  $\gamma$  – текущее значение угла поворота шайбы насоса.

Зависимость между перемещениями поршня управляющего гидроцилиндра и углом наклона шайбы насоса определяется соотношением:

$$\gamma = \arctg \frac{y_{\max} - y}{R} \quad (1.48)$$

где  $y$  – координата, определяющая положение поршня гидроцилиндра;  $y_{\max}$  – максимальное значение, которое может принимать  $y$ ;  $R$  – радиус опорной шайбы насоса.

Трубопроводы, соединяющие насос с гидроцилиндрами, считаем достаточно короткими, чтобы процессы в них можно было рассматривать без учёта распределённости параметров рабочей среды по длине гидролинии. Тогда, исходя из условия неразрывности течения жидкости, уравнение расходов в напорной гидролинии можно записать в виде:

$$K_{сж} \cdot \frac{dp_n}{dt} = Q_n - Q_s - Q_y, \quad (1.49)$$

где  $Q_n$  – подача насоса,  $Q_s$  – расход, необходимый для работы гидроприводов;  $Q_y$  – расход жидкости, идущей на управление;  $K_{сж}$  – коэффициент, учитывающий сжимаемость рабочей жидкости в трубопроводах;  $p_n$  – давление насоса.

Расход жидкости, необходимый для работы гидроприводов:

$$Q_s = \sum_{i=1}^n Q_i, \quad (1.50)$$

где  $Q_i$  – расход, потребляемый  $i$  – м гидроприводом;  $n$  – число гидроприводов, подключенных к насосу.

Уравнение движения плунжера ограничителя давления.

$$m_{3.озд} \frac{d^2 x_{озд}}{dt^2} + k_{мп.озд} \frac{dx_{озд}}{dt} + c_{пр.озд} x_{озд} + P_{н.озд} + R_{зд.озд} = (p_n - p_{сл}) \cdot F_{3.озд} \quad (1.51)$$

где  $m_{3.озд}$  – масса плунжера и присоединенных к нему подвижных частей золотника ограничителя давления;  $x_{озд}$  – перемещение плунжера золотника ограничителя давления;  $k_{мп.озд}$  – коэффициент, учитывающий вязкое трение в зазоре между плунжером золотника и гильзой;  $c_{пр.озд}$  – жесткость пружины ограничителя давления;  $P_{н.озд}$  – сила предварительного поджатия пружины;  $R_{зд.озд}$  – гидродинамическая сила, действующая на золотник со стороны рабочей жидкости;  $F_{3.озд}$  – площадь торца плунжера золотника ограничителя давления.

Уравнение движения шайбы насоса и цилиндров управления

$$m_{цy} \frac{d^2 y}{dt^2} + k_{мпцy} \frac{dy}{dt} + c_{прцy} y + P_{m1} + P_{po} = (p_{цy} - p_{сл}) \cdot F_{цy} \quad (1.52)$$

где  $m_{цy}$  – массы поршней гидроцилиндров управления и присоединенных к ним подвижных частей механизма регулировки подачи насоса;  $k_{мпцy}$  – коэффициенты вязкого трения в зазорах между поршнями и гильзами цилиндров управления;  $c_{прцy}$  – жесткость пружины гидроцилиндра управления №1;  $P_{m1}$  – сила предварительного поджатия пружины;  $F_{цy}$  – площади поршней гидроцилиндров управления;  $P_{po}$  – суммарный момент сопротивления, приложенный к органу регулирования насоса,  $y$  – перемещение штока гидроцилиндра управления,  $p_{цy}$  – давление в полости цилиндра управления.

Каждая нога экзоскелетона имеет три активных степени подвижности. Соответственно на одну ногу приходится три исполнительных гидроцилиндра, три ЭГУ.

На одну руку приходится два гидроцилиндра, и два ЭГУ. Таким образом, экзоскелетон включает в себя 10 гидроцилиндров. Все они питаются от одного насоса.

Система уравнений, описывающих работу гидроприводов двух ног, двух рук и насоса с регулятором давления, представляет собой систему обыкновенных дифференциальных уравнений (ОДУ) порядка 116,

записанных в нормальной форме Коши. Для решения системы ОДУ использовался метод интегрирования Рунге-Кутты в модификации Мерсона.

Для определения адекватности составленных математических моделей элементов, входящих в гидропривод, были проведено математическое моделирование отдельных элементов, в частности ЭГУ.

Характеристики ЭГУ фирмы Rexroth приведены на рис. 1.51, которые представляют собой реакцию ЭГУ на ступенчатый входной электрический сигнал при различных значениях давления питания.

Основными характеристиками переходного процесса является время переходного процесса и его вид.

На рис. 1.52 приведена аналогичная характеристика, полученная при моделировании ЭГУ.

На вход ЭГУ подавался ступенчатый сигнал. Давление питания ЭГУ при этом составляло 210 бар.

Из графика видно, что характер и время переходного процесса, полученные при моделировании работы ЭГУ соответствуют экспериментальным данным, приведённым в каталоге для данного ЭГУ.

Программа для моделирования гидравлической системы управления экзоскелетона позволяет задавать входное напряжение в виде ступенчатого воздействия или в виде гармонического сигнала.

При задании усилий, действующих на исполнительные гидродвигатели можно выбирать либо постоянную нагрузку, либо меняющуюся по гармоническому закону.

Нагрузки, которые действуют на исполнительные гидроцилиндры экзоскелетона при движении, получены из данных Nanyang Technological University. В таблице 1.1 приведены моменты в каждом из суставов. Исходя из геометрических размеров звеньев экзоскелетона, были определены усилия, действующие на гидроцилиндры. Если считать, что моменты, приведённые в материалах Nanyang Technological University, имеют максимальное значение, то расчёт проведён для положений звеньев экзоскелетона, при котором в суставах возникают максимальные моменты. Для ног это положение, когда угол в коленном суставе равен  $90^\circ$ .

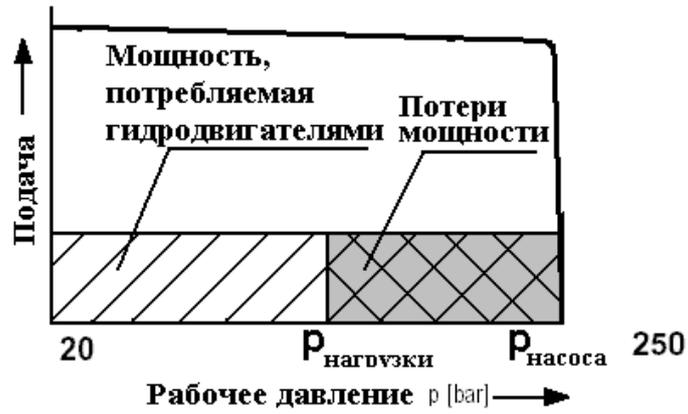


Рис. 1.46. Характеристика гидропривода, работающего на постоянное давление.

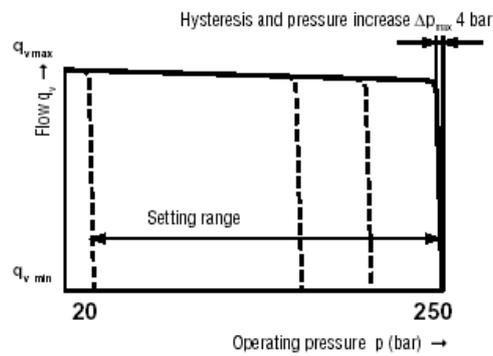


Рис. 1.47. Характеристика насоса, работающего на постоянное давление.

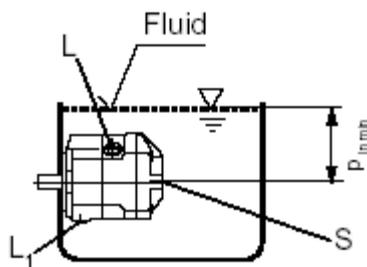


Рис. 1.48. Схема расположения насоса внутри бака.

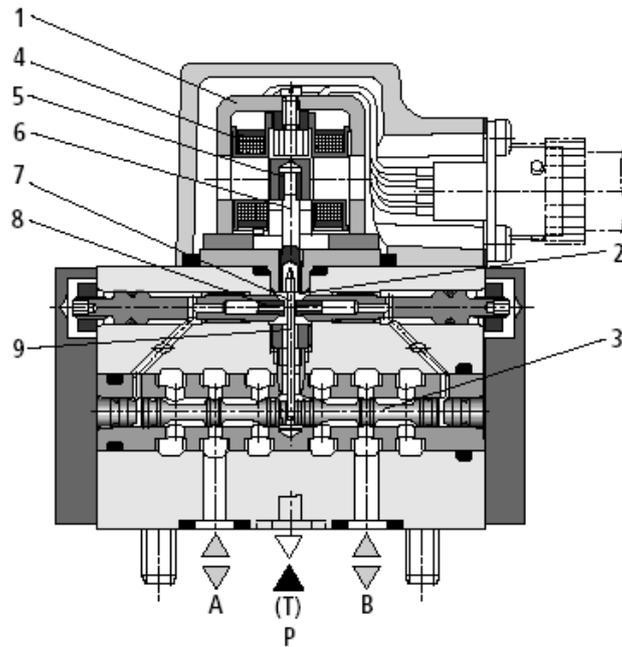


Рис. 1.49. Электрогидравлический усилитель типа сопло-заслонка (servo valve).

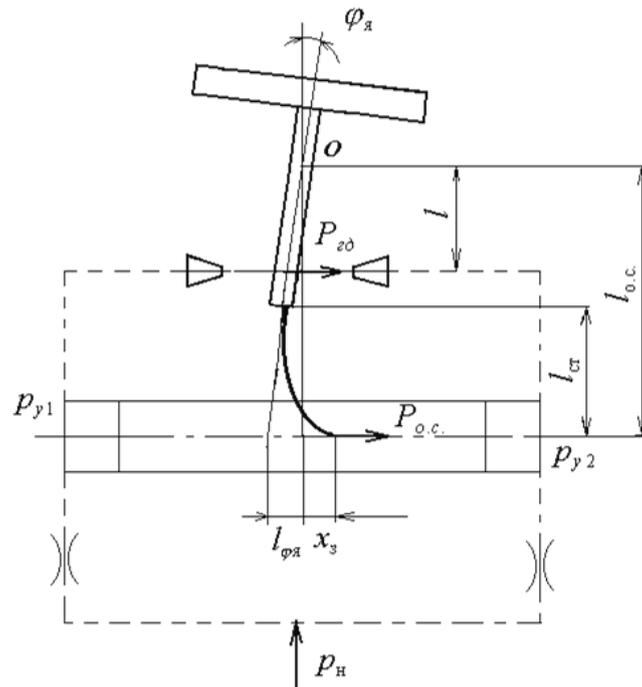


Рис. 1.50. Схема для расчёта электрогидравлического усилителя с силовой обратной связью от золотника к заслонке.

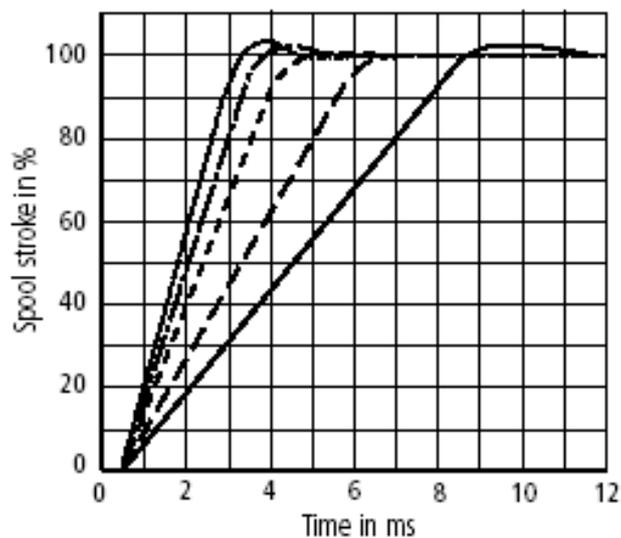


Рис. 1.51. Характеристика ЭГУ при ступенчатом воздействии.

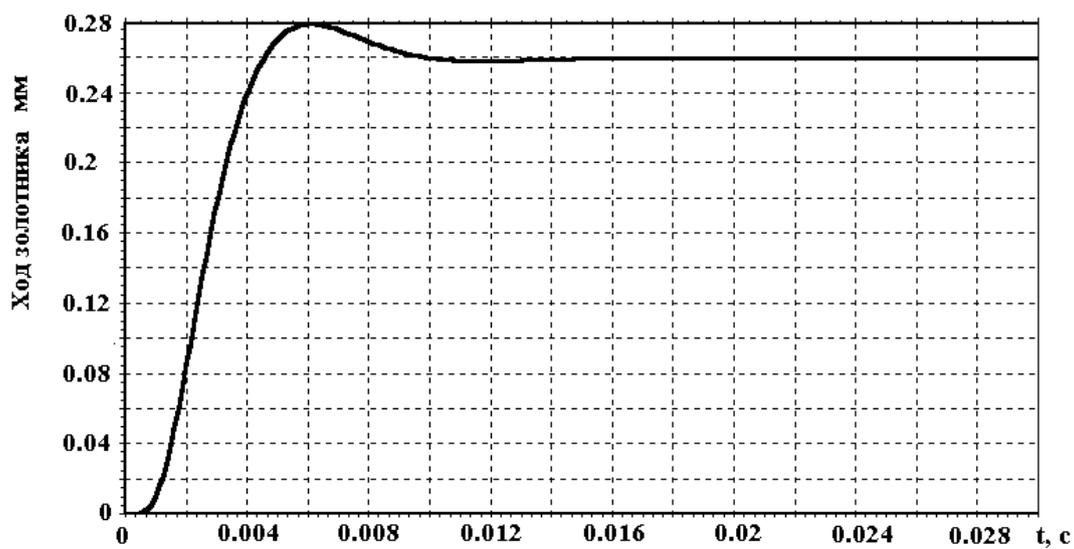


Рис. 1.52. Перемещение золотника ЭГУ при ступенчатом воздействии.

Таблица 1.3

Наименование звена	Величина момента, Н*м	Величина силы, кН
Колено	370.1	2.96
Бедро	403.8	4.038
Голень	370.1	2.96
Плечо	296.1	4.23
Локоть	164.4	2.342

На рис. 1.53-1.63 приведены результаты моделирования работы гидропривода экзоскелетона. Для моделирования были выбраны два режима движения наиболее характерных для человека.

Первый режим соответствует вставанию человека из полусидящего положения и одновременное поднятие рук. Движение осуществляется из положения, когда нога в коленном суставе согнута на  $90^\circ$ , звено от колена до таза находится в горизонтальном положении. Конечное положение соответствует моменту, когда человек находится в вертикальном положении. Такое движение характерно для человека при поднятии тяжестей.

На вход каждого следящего привода подавались напряжения, имеющие синусоидальный характер. Период синусоиды соответствует вставанию экзоскелетона из полусидящего положения за 1.5с и приседание за тот же период. Максимальные нагрузки, действующие на гидроцилиндры получены исходя из величин крутящих моментов в каждом из суставов, полученных в Технологическом университете Сингапура. Исходя из кинематической схемы экзоскелетона, в которой в качестве исполнительных гидродвигателей применяются гидроцилиндры, были получены нагрузки, которые будут приложены к штокам гидроцилиндров.

Нагрузка, действующая на гидроцилиндры, имеет постоянное значение. Значение величин нагрузок приведены в таблице 1.3.

При вставании гидроцилиндры, принадлежащие противоположным сторонам экзоскелетона, движутся в одной фазе с одинаковыми скоростями. Перемещения штоков гидроцилиндров соответствуют максимальным значениям перемещений, полученными из кинематической схемы экзоскелетона.

Из графиков перемещения гидроцилиндров бедра (рис. 1.55), колена (рис. 1.54), голени (рис. 1.53), локтя (рис. 1.56), плеча (рис. 1.57) видно, что гидроприводы отслеживают входной сигнал.

На рис. 1.58 приведены графики мощности насоса и полезной мощности, развиваемой гидроцилиндрами всех звеньев экзоскелетона. Эти графики характеризуют энергетическую эффективность гидропривода.

Как видно из графика коэффициент полезного действия гидропривода составляет в среднем 50% и выше, что для гидропривода с объёмно-дрессельным регулированием является вполне удовлетворительным.

Второй режим движения моделирует ходьбу человека. При движении одна нога находится в фазе опоры, эта фаза характеризуется высокими нагрузками и малыми скоростями, вторая нога в фазе переноса, которая характеризуется малыми нагрузками и высокими скоростями.

Поскольку зависимости между величинами перемещений гидроцилиндров и нагрузок, приложенных к ним нет, то для моделирования был выбран синусоидальный закон изменения нагрузок с амплитудой равной максимальной нагрузке.

Нагрузки, действующие на гидроцилиндры, а также их скорости могут значительно отличаться при разных режимах работы экзоскелетона.

Поскольку данных о нагрузках, возникающих в суставах экзоскелетона при ходьбе в настоящее время не имеется, то в качестве нагрузочных характеристик были приняты синусоидальные законы изменения. Нагрузка изменялась по синусоиде с частотой 2 Гц для гидропривода ног и частотой 1 Гц для гидропривода рук. Амплитуда нагрузок в степенях подвижности: колено – 1000 Н, голень – 1200 Н, бедро – 1500 Н, плечо – 1000 Н, локоть – 1000 Н. При этом как входной сигнал, так и нагрузки, действующие на одноименные гидроцилиндры, принадлежащие противоположным сторонам, изменяются в противофазе.

Гидропривод хорошо отслеживает входные сигналы (рис. 1.60-1.62). Перемещения гидроцилиндров, принадлежащих противоположной стороне имеют тот же вид графиков, только с противоположной фазой.

Энергетические характеристики можно оценить по графику мощностей насоса и полезной мощности (рис. 1.63). Коэффициент полезного действия в среднем равен 30%. Было проведено математическое моделирование нескольких режимов с различными значениями частоты и амплитуды.

Для всех режимов гидропривод хорошо отслеживает входной сигнал и имеет к.п.д. 30-40%, что является хорошим значением для гидропривода с объёмно-дрессельным регулированием.

Более точные результаты можно получить, зная нагрузочные характеристики для каждого сустава экзоскелетона.

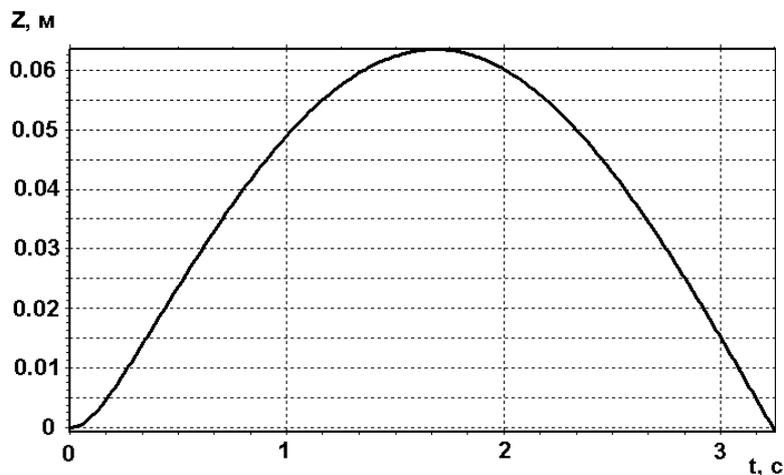


Рис. 1.52. Перемещение гидроцилиндра голени при вставании и приседании экзоскелетона.

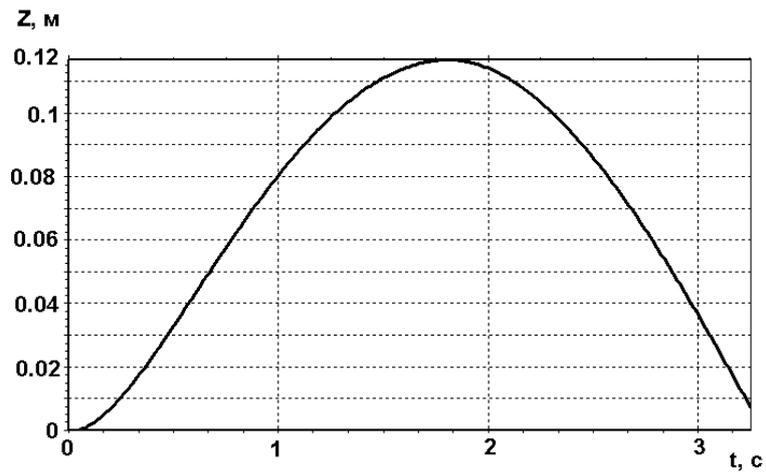


Рис. 1.53. Перемещение гидроцилиндра колена при вставании и приседании экзоскелетона.

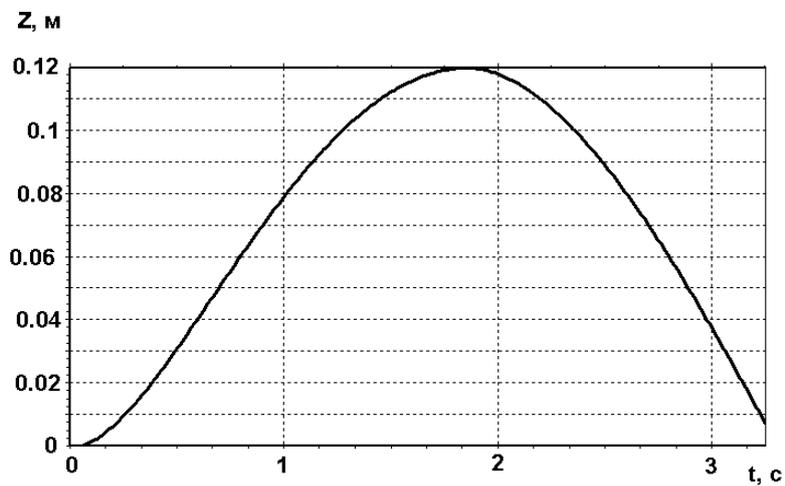


Рис. 1.54. Перемещение гидроцилиндра бедра при вставании и приседании экзоскелетона.

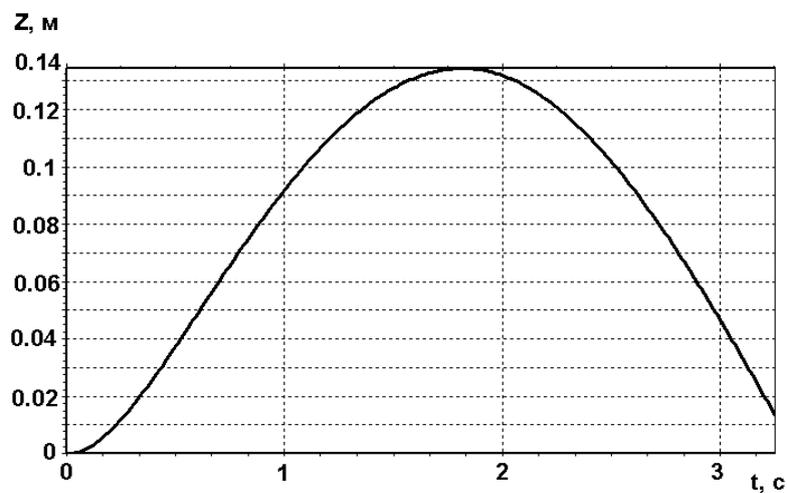


Рис. 1.55. Перемещение гидроцилиндра локтя при поднятии и опускании рук экзоскелетона.

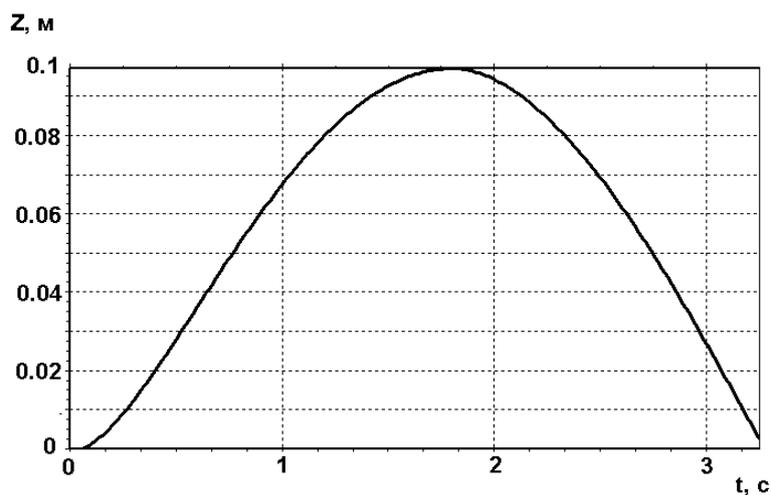


Рис. 1.56. Перемещение гидроцилиндра плеча при поднятии и опускании рук экзоскелетона.

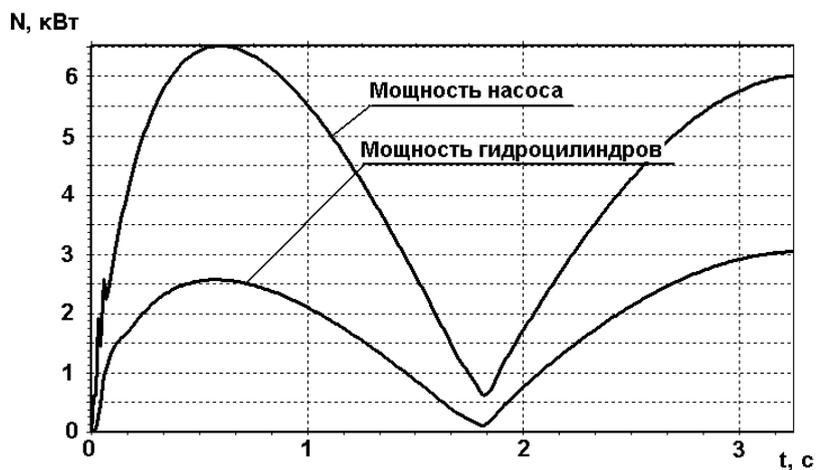


Рис. 1.58. Мощности насоса и полезной мощности всех гидроцилиндров.

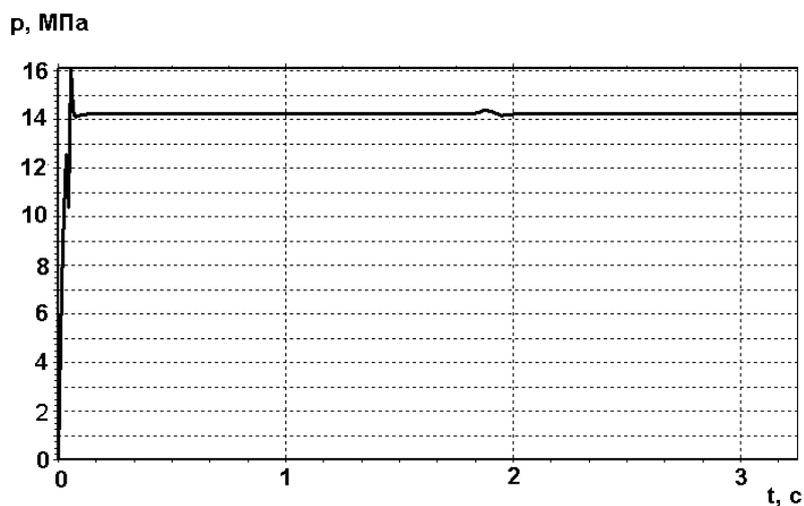


Рис. 1.59. Давление насоса.

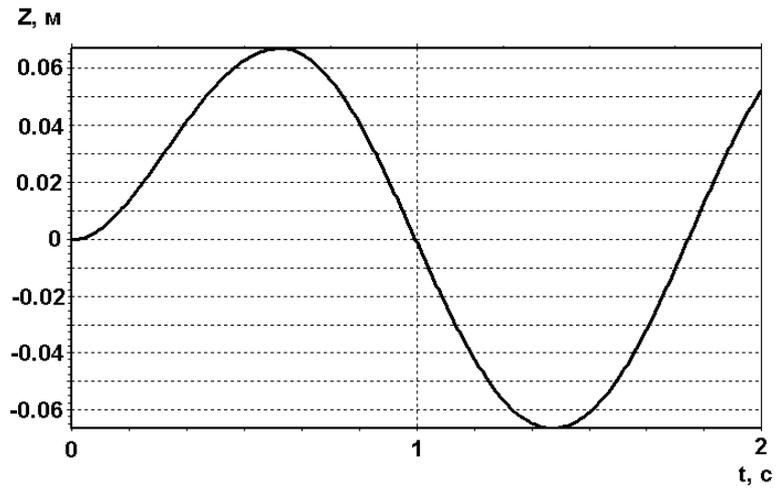


Рис. 1.60. Перемещение гидроцилиндра бедра при ходьбе.

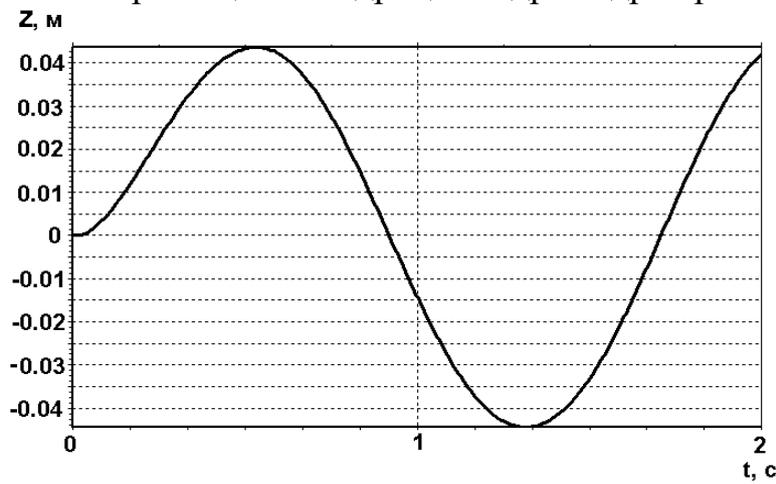


Рис. 1.61. Перемещение гидроцилиндра голени при ходьбе.

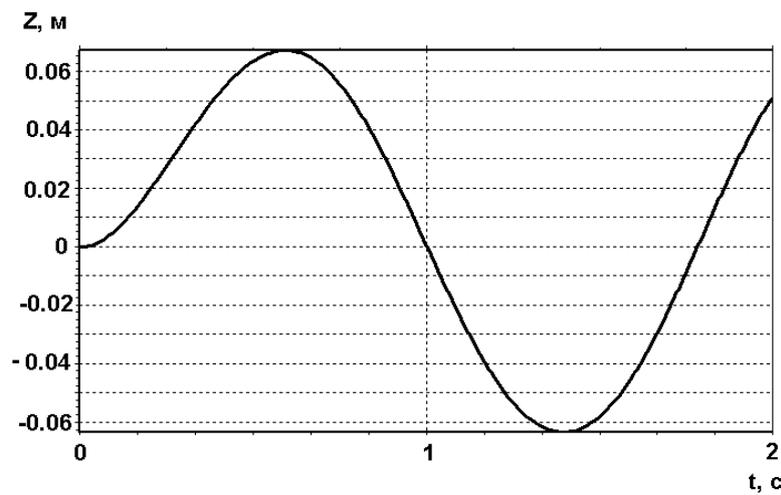
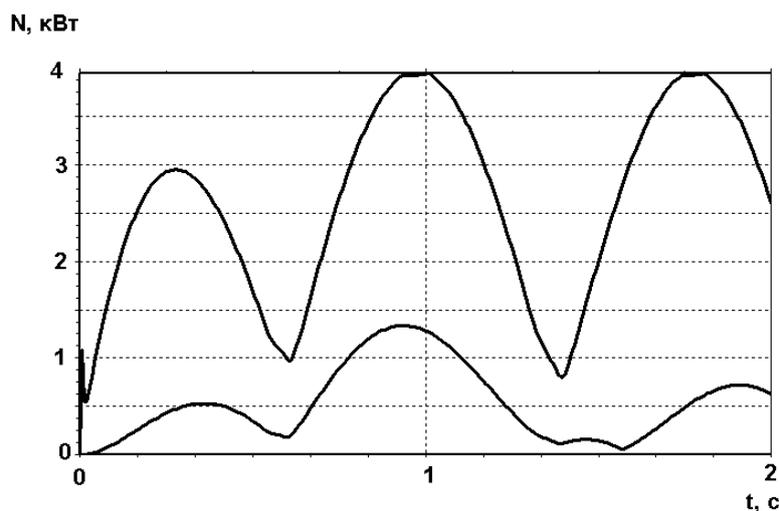


Рис. 1.62. Перемещение гидроцилиндра колена при ходьбе.



*Рис. 1.63.* Графики мощностей насоса и полезной мощности гидроцилиндров экзоскелетона.

По полученным результатам моделирования предлагаемого гидропривода можно сделать следующие выводы. Насос, работающий при постоянном давлении, поддерживает в системе постоянное давление. Поэтому такой насос не может обеспечить высокие энергетические характеристики во всём диапазоне изменения рабочих давлений гидроприводов экзоскелетона. Это видно из энергетической характеристики, приведенной на рис. 1.46. Если давление в гидроцилиндрах близко к давлению настройки насоса, то потери в гидросистеме будут минимальными, так как потери давления на сервоклапанах будут меньше. Таким образом, регулировка рабочего давления насоса в зависимости от режима движения экзоскелетона может улучшить энергетические характеристики гидропривода.

Регулятор давления может изменять рабочее давление насоса в диапазоне от 20 до 210 бар. Это обеспечивается величиной предварительного поджатия пружины золотника ограничителя давления. Поскольку в течение работы экзоскелетона нет возможности регулировать величину поджатия пружины, то существует модификация насоса с регулятором давления с дистанционным управлением. Принципиальная гидравлическая схема такого насоса приведена на рис. 1.45.

Например, когда экзоскелетон преодолевает значительные нагрузки и требуется высокое давление, то в гидроцилиндрах регулятор давления должен быть настроен на высокое давление. Примером могут служить результаты моделирования первого режима: когда экзоскелетон встаёт из полусидящего положения с нагрузкой. Во время же ходьбы при малой нагрузке, когда не требуется высокого давления в гидроцилиндрах, регулятор может быть настроен на меньшее давление, и тем самым снизить потери давления на сервоклапанах. Таким образом, изменяя давление настройки регулятора давления насоса в зависимости от нагрузки и режима движения можно оптимизировать энергетические характеристики.

Сигналом для изменения настройки регулятора давления насоса может служить сигнал с датчика нагрузки (тензодатчика), который может быть установлен в каком-либо звене экзоскелетона, например, спине, тазе и т.д. В зависимости от величины сигнала с этого датчика можно определить нагрузки, которые будут действовать на гидроцилиндры и в зависимости от этого установить рабочее давление насоса.

В целом рассмотренный вариант гидропривода обеспечивает работу экзоскелетона при различных режимах работы.

#### **1.4. Некоторые особенности моделирования электромеханических и гидравлических систем управления**

Как видно из приведенных функциональных и структурных схем электромеханических и гидравлических систем управления, в них присутствуют нелинейные элементы. Математические модели СУ должны учитывать различные нелинейности, имеющиеся в СУ, объекте регулирования, исполнительных и регулирующих устройствах. При моделировании самого регулятора и алгоритма управления им необходимо учитывать возмущающие нагрузки, воздействующие на объект управления. Использование в алгоритме управления ШИМа приводит математические модели СУ к системе обыкновенных дифференциальных уравнений (ОДУ) с разрывными правыми частями [11, 28, 37, 40, 42].

Перечисленное выше накладывает существенные ограничения на выбор и использование численных методов интегрирования при моделировании СУ и их элементов. Выбор численных методов усложняется наличием нелинейностей и неудерживающих связей в СУ и ее механической части.

Прежде всего, следует выбирать численные методы, которые соответствовали бы классу решаемых задач. Здесь в качестве используемых численных методов интегрирования положены алгоритмы одношаговых явных численных схем, которые имеют некоторые преимущества перед многошаговыми. В частности, многошаговые методы приводят в некотором смысле к осреднению решения («срезанию экстремумов»), что при моделировании некоторых динамических объектов делает их неприемлемыми. Кроме того, поскольку правая часть ОДУ исследуемых динамических объектов разрывная, то применение многошаговых методов является малоэффективным. Достаточно полный обзор работ, посвящённых численному решению задачи Коши многошаговыми методами, содержится в [7, 45, 47, 48]. Использование без итерационных методов интегрирования позволяет сравнительно легко оценить затраты на шаг интегрирования при проведении расчётов и упрощает программную реализацию алгоритмов интегрирования.

Эти проблемы моделирования рассмотрены на примерах решения задач исследования показателей качества электроприводов постоянного и переменного тока и гидроприводов, используемых в СУ механическими объектами.

Системы ОДУ, описывающие электроприводы постоянного и переменного тока приведены в [21, 23, 25, 26, 34, 43, 46]. Эти уравнения состоят из уравнений, описывающих электродвигатель, механическую передачу, а также из уравнений, описывающих собственно СУ электродвигателя (иногда эту систему управления называют преобразователем). Система управления (преобразователь) состоит из регуляторов, усилителей мощности, устройства широтно-импульсной модуляции (ШИМ) и разного рода других устройств, которые описываются, как правило, нелинейными функциями. Более подробно об этом см., например, [6, 21, 22].

Динамику электропривода постоянного тока можно описать следующей системой дифференциальных и алгебраических уравнений:

$$\begin{aligned}\frac{dx_1}{dt} &= [k_m x_2 - M_c - F_6(x_1)]/J \\ \frac{dx_2}{dt} &= [U_n (F_4(F_1(k_y y_{11}) + U_p) + F_5(F_1(k_y y_{11} + U_p))) - k_\omega x_1 - R x_2]/L \\ \frac{dx_3}{dt} &= \left[ \frac{1}{k_{pm_2}} \left( y_{15} + T_{pm_{33}} \frac{dy_{15}}{dt} \right) - x_3 \right] / T_{pm_2} \\ \frac{dx_4}{dt} &= (k_{om} x_2 - x_4) / T_{om} \\ \frac{dx_5}{dt} &= \left[ \left( y_{14} + T_{pc_2} \frac{dy_{14}}{dt} \right) \frac{1}{k_{pc}} - x_5 \right] / T_{pc_1} \\ \frac{dx_6}{dt} &= (k_{oc} x_1 - x_6) / T_{oc}\end{aligned}$$

$$y_{11} = y_{12} - y_{10}; \quad y_{12} = y_{13} - x_4; \quad y_{10} = x_3 + y_8;$$

$$y_8 = \left[ F_1(k_y y_{11}) + T_{pm_1} \frac{dF_1(k_y y_{11})}{dt} \right] / k_{pm_1};$$

$$y_{13} = F_2(F_1(k_y y_{16})); \quad y_{14} = F_1(k_y ((U_{zc} - x_6) - x_5));$$

$$y_{15} = F_6(F_1(k_y y_{11})); \quad y_{16} = U_{zc} - x_6 - x_5$$

С начальными условиями:

$$x_1(t_0) = x_{10}; \quad x_2(t_0) = x_{20}; \quad x_3(t_0) = x_{30};$$

$$x_4(t_0) = x_{40}; \quad x_5(t_0) = x_{50}; \quad x_6(t_0) = x_{60};$$

В этих уравнениях в качестве независимых переменных являются  $x_1$  – угловая скорость вала двигателя;  $x_2$  – ток в обмотке якоря двигателя;  $x_3$  – промежуточная переменная в модели регулятора тока;  $x_4$  – сигнал с датчика обратной связи по току;  $x_5$  – промежуточная переменная в модели регулятора скорости;  $x_6$  – сигнал с датчика обратной связи по скорости. Остальные

обозначения соответствуют обозначениям, приведенным на структурной схеме.

Математическая модель гидравлической системы основана на уравнениях, определяющих расходы жидкости через дроссельные устройства, на уравнениях неразрывности течения, на уравнениях состояния рабочей жидкости и на уравнениях движения отдельных механических звеньев.

Математическая модель гидропривода состоит из математических моделей аксиально-поршневого насоса с регулируемой подачей; гидравлических линий; исполнительного гидроцилиндра; пропорционального распределителя.

Из сравнения уравнений, описывающих процессы в асинхронном электроприводе, и уравнений для электроприводов постоянного тока, видно, что уравнения, описывающие асинхронный электропривод, значительно сложнее уравнений, описывающих электропривод постоянного тока. Причина этого в том, что процессы, протекающие в асинхронном электродвигателе, имеют более сложный характер, в электродвигателе постоянного тока, а также в том, что управление асинхронным электродвигателем формируется более сложно. Вообще говоря, интегрирование таких дифференциальных уравнений не представляет особых затруднений, если в алгоритме управления используется обычный аналоговый сигнал. В этом случае использование одношаговых методов типа явных методов Рунге-Кутты оказывается достаточно эффективным как с точки зрения точности, так и времени интегрирования. Однако в СУ исполнительными органами многих механических объектов широко используется ШИМ, обладающая целым рядом преимуществ по сравнению с непрерывными способами управления. В частности, применение ШИМа позволяет уменьшить потери в обмотках электродвигателя и существенно снизить пульсации действующих моментов. Это, как правило, приводит к уменьшению энергозатрат. Поэтому использование в алгоритме управления электродвигателем ШИМа, а также учет в математической модели СУ других нелинейных функций и учет наличия неударживающих связей, в качестве которых являются, например, механический упор и зазор (люфт) в механической передаче, приводит к системе ОДУ с существенно негладкими и даже разрывными правыми частями. Эти особенности затрудняют использование метода интегрирования при численном моделировании СУ и их элементов. Как показал опыт, методы с автоматическим выбором шага и контролем точности могут иметь недостаточную эффективность для расчета управления электродвигателя с часто меняющимся управлением, кусочно-гладкими, имеющими разрывы правыми частями.

Ниже будут рассмотрены некоторые особенности численного интегрирования дифференциальных уравнений электропривода, управляемого с помощью ШИМа.

Интегрирование с учетом ШИМа в математической модели СУ имеет свои особенности и зависит от математической модели ШИМа (о математических моделях ШИМа см., например, [6, 7, 11, 20, 21]).

Рассмотрим кратко **математические модели ШИМа**. Преобразователь, осуществляющий ШИМ, есть управляющее звено привода, которое является сложным нелинейным импульсным элементом. Он преобразует непрерывную функцию входного управляющего сигнала в дискретную функцию моментов открывания ключей управляющей схемы. В следящих приводах импульсное регулирование двигателем реализуется обычно с помощью мостовой схемы, в которой в качестве переключающих элементов используются транзисторы. Различают два основных закона коммутации транзисторов моста – симметричный (или двухполярный) и несимметричный (однополярный). В первом случае полученный в результате сигнал представляет собой серию разнополярных импульсов, а во втором – серию однополярных импульсов. Пульсация тока в двигателе при несимметричном управлении меньше, чем при симметричном, следовательно, и дополнительные потери мощности также меньше. Поэтому однополярный ШИМ применяется несколько чаще двухполярного. Дадим краткое описание устройства формирования ШИМа. В основу управления положен способ, заключающийся в том, что управляющие импульсы формируются в результате сравнения опорного синусоидального напряжения с постоянным напряжением управления. Фаза управляющего импульса определяется равенством опорного синусоидального напряжения и напряжения управления. Упрощенная блок-схема устройства ШИМа может иметь вид, представленный на рис. 1.64.

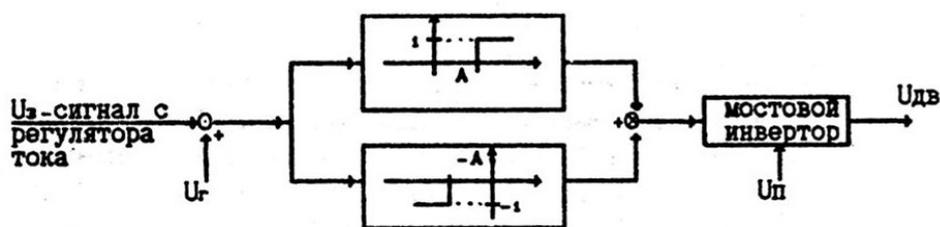


Рис. 1.64. Упрощенная блок-схема устройства ШИМа.

Здесь  $U_z$  – сигнал задания напряжения,  $U_g$  – вырабатываемый генератором периодический сигнал напряжения треугольной формы с заданной амплитудой  $U_g$  и частотой  $f_g$ , необходимых для формирования ШИМа.  $U_p$  – напряжение с источника питания,  $U_{дв}$  – выходной сигнал напряжения с амплитудой и формой, выработанными устройством ШИМа. Вид вырабатываемого генератором периодического сигнала напряжения представлен на рис. 1.65.

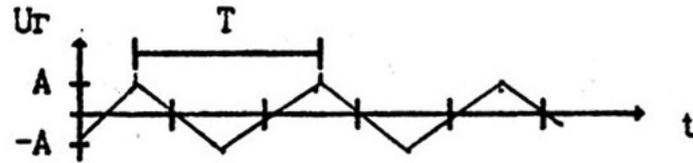


Рис. 1.65. Вид периодического сигнала напряжений, вырабатываемого генератором.

В устройстве формирования ШИМа происходит алгебраическое суммирование заданного входного напряжения  $U_3(t)$ , сигнала с генератора  $U_r(t)$  и сигналов смещения и формирование на основе полученной суммы сигналы управления силовыми ключами. Затем с помощью усилителей (они на блок-схеме не показаны) эти сигналы повышаются до уровня, необходимого для управления силовыми ключами. И, наконец, мостовой инвертор осуществляет переключение двигателя к источнику питания в соответствии со сформированными сигналами управления силовыми ключами. Принципиальная схема мостового инвертора приведена на рис. 1.66.

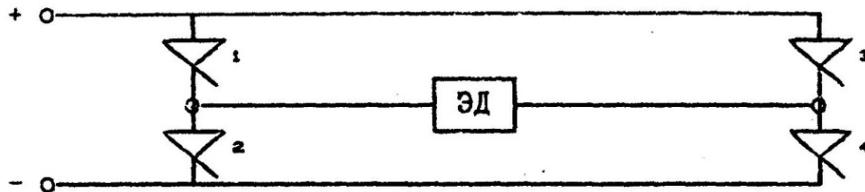


Рис. 1.66. Принципиальная схема мостового инвертора.

Инвертор имеет два полностью управляемых состояния. Когда открыты ключи 1, 4 (ключи 2, 3 открыты), то ЭД питается напряжением  $+V_{п}$ . Когда открыты ключи 2, 3 (ключи 1, 4 закрыты), то ЭД питается напряжением  $-V_{п}$ . В результате будет сгенерирован выходной сигнал, имеющий форму, приведенную на рис. 1.67.

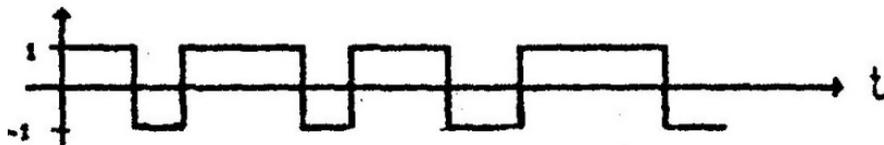


Рис. 1.67. Выходной сигнал после работы устройства формирования ШИМа.

Ширина «полки» ШИМа будет зависеть от величины и формы входного управляющего сигнала  $U_3$ , определяющего требуемый закон управления исполнительным органом.

Так как цифровое управление имеет дискретный характер, то на вид управляющего сигнала существенное влияние оказывает величина такта управления. Поэтому сигнал управления, который, например, имеет форму

гармонического сигнала, аппроксимируется кусочно-постоянными отрезками, длина которых определяется величиной такта управления. Значение же такта управления может меняться в зависимости от требуемого качества управления и характеристик, используемой для управления вычислительной техники.

### Рассмотрим алгоритм формирования сигналов ШИМа.

Схему процесса формирования ШИМа, приведенную выше, можно записать в виде ниже следующих соотношений [40]. На заданный момент времени, определяемом счетчиком тактов таймера, находим значение  $U_r$ . Для этого в задающем периодическом сигнале треугольной формы выделяем три участка:

1.  $0 \leq \tau \leq T/4$
2.  $T/4 < \tau \leq 3 \cdot T/4$
3.  $3 \cdot T/4 < \tau \leq T$

$$\tau = \frac{t - t_0}{T} = \left[ \frac{t - t_0}{T} \right],$$

где  $t$  – текущее время (имеет размерность числа тактов таймера),  $t_0$  – время отсчета,  $T$  – период задающего периодического треугольного сигнала. Угол наклона отрезка прямой, определяющего первую четверть заданного сигнала, определим, как  $K = A/(T/4)$ , в котором  $A$  – амплитуда задающего треугольного сигнала. Тогда для определения значения  $U_r$  имеем следующие соотношения:

- для первого участка  $U_r = K \cdot \tau$ ;
- для второго участка  $U_r = -K \cdot (\tau - T/4)$ ;
- для третьего участка  $U_r = -K \cdot (\tau - 3 \cdot T/4)$ ;

Сигнал на входе в сравнивающее устройство определяют, как  $U_1 = U_{вх} + U_r$ . Работу сравнивающего устройства можем описать следующими соотношениями:

- если  $U_1 > -A$ ,  $\forall U_2 = 0, \bar{U}_2 = 0$ ;
- если  $U_1 \geq A$ ,  $\forall U_2 = 1, \bar{U}_2 = 0$ ;
- если  $U_1 \leq -A$ ,  $\forall U_2 = 0, \bar{U}_2 = -1$ .

И, окончательно, в каждый момент времени для заданного сигнала управления  $U_z$  получаем, что сигнал на вход в мостовой инвертор будет определяться, как  $U_k = U_2 + \bar{U}_2$ . Значение  $U_k$  в каждый момент времени будет являться 0 или 1 с длительностью (скважность), определяемой требуемым сигналом управления и амплитудой и частотой, вырабатываемого генератором периодического треугольного напряжения.

Существуют и другие способы вычисления значений  $U_k$ . Один из таких способов состоит в следующем. Определяются точки пересечения заданного периодического треугольного сигнала  $U_r(t)$  с требуемым управляющим

сигналом  $U_3(t)$ . Далее выделяются участки, на которых значения треугольного напряжения больше (меньше) значения управляющего сигнала. На тех участках, где значение треугольного сигнала больше управляющего, имеем участок постоянного напряжения  $U = -A$ , а там, где значение треугольного сигнала меньше управляющего, имеем участок постоянного напряжения  $U = A$ . Особенности реализации этого способа формирования ШИМа состоят в представлении в ЭВМ периодического треугольного напряжения.

Опишем два способа его представления.

В первом способе периодический треугольный сигнал может быть задан кусочно-линейной кривой с некоторой амплитудой  $A$  и периодом  $T=1/f$  ( $f$  – частота).

Во втором способе задания этого сигнала он задается в виде аналитической зависимости, полученной как результат разложения периодического сигнала в ряд Фурье. Эту зависимость можно записать в виде:

$$U_r = \frac{4A}{\pi} \cdot \left[ -\sin \omega t + \frac{1}{9} \sin 3\omega t - \frac{1}{25} \sin 5\omega t + \frac{1}{49} \sin 7\omega t - \frac{1}{81} \sin 9\omega t + \dots \right]$$

Здесь  $A$  – амплитуда задаваемого периодического треугольного напряжения, а  $\omega = 2\pi f$ . Из-за различного представления в ЭВМ периодического треугольного напряжения соответственно каждому способу представления соответствует свой алгоритм определения точек пересечения треугольного и заданного управляющего сигналов. Если треугольный сигнал представлен в виде аналитической зависимости, то точки пересечения могут быть определены в результате решения нелинейного алгебраического уравнения  $F = U_r - U_3$ , в котором  $U_r$  – функция, представляющая периодический треугольный сигнал, а  $U_3$  – функция, характеризующая изменение управляющего сигнала. Это уравнение удобно решать с помощью метода Ньютона, для которого в качестве нулевого приближения целесообразно брать пересечение треугольного сигнала с осью абсцисс. В таком случае имеем сходимость за 2-3 итерации. Если же периодический треугольный сигнал представлен в виде кусочно-линейной кривой, то искомые точки пересечения можно определить как пересечение постоянного сигнала  $U = A$  (или  $U = -A$ ) с отрезком прямой треугольного сигнала. В этом случае в вычислениях будут преобладать логические операции.

**Рассмотрим особенности интегрирования системы ОДУ,** описывающей СУ электропривода при использовании первой модели ШИМа. Эта модель использовалась при моделировании асинхронного электропривода. Согласно алгоритма и математической модели формирования ШИМа после определения значений напряжений во всех трех фазах делается переход на интегрирование системы. Интегрирование ведется на интервалах постоянства значений включенных или выключенных транзисторных ключей. Напряжения  $U_a$ ,  $U_b$ ,  $U_c$ , подаваемые на обмотку электродвигателя,

вычисляются один раз в начале этого интервала. Интегрирование на каждом интервале выполнялось классическим методом Рунге-Кутты четвертого порядка с постоянным шагом, меньшим или равным рекомендуемому шагу интегрирования. Это означает, что в момент изменения управления (разрыв в правых частях) для выполнения требуемой локальной точности на каждом шаге интегрирования происходит уменьшение шага. Далее интегрирование ведется с очень малым шагом. Едва шаг успевает увеличиться, как опять меняется управление. Использование методов с автоматическим выбором шага может приводить в данном случае к не эффективным затратам машинного времени и потери точности интегрирования в точках разрыва. Уменьшение потери точности в точках разрыва требует точного определения этих точек разрыва и изменение шага вблизи них. Интегрирование по участкам постоянства ШИМа, позволяет уменьшить погрешности, возникающие в точках разрыва. Для этого достаточно выбирать шаг, целое число раз укладывающийся в интервале постоянства ШИМа. Однако заранее не известна длина интервала, поэтому можно только рекомендовать шаг. Пусть, например,  $T_0$  – время начала,  $T_1$  – время конца интервала интегрирования,  $h$  – рекомендуемый шаг. Поясним, как выбирается шаг  $h$  в данной модификации метода Рунге-Кутта. Пусть  $n = (T_1 - T_0) / h$  – число шагов, которое надо было бы сделать. Это число может быть не целым. Пусть  $n_0$  – наименьшее целое число, такое, что  $n_0 \geq n$ . Тогда  $h = (T_1 - T_0) / n_0$ . Таким образом, интегрирование ведется с максимально возможным шагом, не превосходящим рекомендуемый. Поскольку локальная погрешность на шаге пропорциональна шагу  $h$ , то интегрирование при таком выборе шага  $h$  ведется с минимальной погрешностью с постоянным шагом. При такой же точности этот метод оказывается быстрее методов с автоматическим выбором шага.

При использовании второй модели ШИМа выполняется аналогичная проверка на выход по времени на начало момента выполнения ШИМа, конца его выполнения, а также его длительности. В эти моменты в программе при вычислениях правых частей происходит переключение на вычисление правых частей, учитывающие эти разрывы в правых частях. При этом возможно изменение заданного шага интегрирования, для «дошагивания» до конца интервала интегрирования.

Аналогичный прием используется и при интегрировании уравнений, в которых имеются другие нелинейности, а также учет неударживающих связей, приводящих к разрывным правым частям дифференциальных уравнений (например, учет сухого трения, люфта, а также определения момента выхода и схода с ограничений типа «упор»).

Хорошие результаты для интегрирования дифференциальных уравнений электромеханических систем имеет метод Рунге-Кутты с модификацией Гилла. Достоинство этого метода состоит в уменьшении требуемой памяти, что существенно для любых ЭВМ, а для малых особенно.

Указать, какая формула из некоторого выбранного семейства формул является наилучшей, сложно. Однако попробуем провести некоторые

рассуждения по этому поводу на примере применения классического метода Рунге-Кутты четвертого порядка и так называемого «правила 3/8». Если рассмотрим дифференциальное уравнение  $y' = f(x)$ , то для классического метода Рунге-Кутты четвертого порядка локальная погрешность на шаге

интегрирования есть  $\varepsilon_1 = \frac{f^{(4)}(x) \cdot h^5}{2880} + O(h^6)$ , а для «правила 3/8» погрешность

на шаге интегрирования определяется как  $\varepsilon_2 = \frac{f^{(4)}(x) \cdot h^5}{6480} + O(h^6)$ . Сравнивая

формулы для  $\varepsilon_1$  и  $\varepsilon_2$  видим, что «правило 3/8» имеет преимущество для рассматриваемого класса уравнений. То есть оно имеет преимущество для дифференциальных уравнений, для которых значение  $f_y \approx 0$ . Анализ

дифференциальных уравнений электромеханических систем подтверждает то, что у них для одной части зависимых переменных это условие выполняется, а для другой –  $f_y$  может иметь разные значения: как положительные, так и отрицательные. Но при достаточно малом шаге интегрирования  $h$  значение

$h \cdot f_y$  принимает малые значения. Поэтому «правило 3/8» предпочтительнее классического метода Рунге-Кутты при интегрировании дифференциальных

уравнений, описывающих электромеханические системы. В тоже время, преимущества этого метода сказываются при очень малых значениях  $h \cdot f_y$ . И

если уравнение интегрируется с умеренной точностью, то практика вычислений указывает на предпочтительность использования формул классического метода Рунге-Кутты. В [7] приводятся соображения в пользу того, что расчетные формулы классического метода Рунге-Кутты более предпочтительны при  $f_y < 0$  и гладких решениях, т.к. они допускают

интегрирование с более крупным шагом без потери чувствительности метода к влиянию вычислительной погрешности. Однако заведомо можно отметить,

что решения рассматриваемых уравнений, описывающих электромеханические системы, свойству гладкости не удовлетворяют. Если

правая часть дифференциальных уравнений непрерывна и ограничена вместе со своими четырьмя производными (и эти производные не очень велики), то достаточно хорошие результаты дает схема четвертого порядка благодаря

очень малому коэффициенту в остаточном члене и быстрому возрастанию точности при уменьшении шага. Дифференциальные уравнения, описывающие электропривод в случае аналогового управления, имеют правые

части, удовлетворяющие этим условиям, по крайней мере, на большей части участка интегрирования. Если же правая часть не имеет указанных производных, то предельный порядок точности этой схемы не может

реализовываться. Это имеет место в случае управления с помощью ШИМа. В таком случае не худшие результаты дают схемы меньшего порядка точности,

равного порядку имеющихся производных, например, для двукратно непрерывно дифференцируемых правых частей – несложные схемы Эйлера: модифицированного или исправленного метода Эйлера. Использование схем

Эйлера может давать результаты по точности соизмеримые с классическим методом Рунге-Кутты, но лучшие по времени интегрирования. В то же время традиционное использование методов интегрирования второго порядка отдает предпочтение модифицированному методу Эйлера, хотя он имеет большую величину главного математического ожидания квадрата погрешности.

### Рассмотрим несколько примеров.

Дифференциальные уравнения контура тока с автоколебательным однополярным широтно-импульсным преобразователем (ШИП) [6] имеют вид:

$$\begin{aligned}\frac{dX_1}{dt} &= \frac{R}{L} \cdot X_1 - \frac{F_1 \cdot F_2}{L} \cdot U_i \\ \frac{dX_2}{dt} &= \frac{1}{T_{dc}} \cdot X_2 + \frac{K_{dc}}{T_{dc}} \cdot X_1,\end{aligned}$$

здесь  $F_1, F_2$  – функции, определяющие функционирование ШИМа (см. [21]). Якобиан этой системы

$$J = \begin{pmatrix} -\frac{R}{L} & 0 \\ \frac{K_{dc}}{T_{dc}} & -\frac{1}{T_{dc}} \end{pmatrix}$$

Определитель Якобиана  $J \neq 0$  и может принимать большие значения, величина которого зависит от параметров обмотки двигателя и характеристики датчика тока. Для двигателя с параметрами  $R=100\text{Ом}$ , индуктивности  $L=0.5\text{Гн}$  и  $T_{dc}=0.001\text{сек}$  получаем, что  $J=2 \cdot 10^5$ . Собственные числа этого Якобиана для двигателя с указанными параметрами:  $\lambda_1 = -200$ ,  $\lambda_2 = -1000$ . Отсюда следует, что решение данного уравнения имеет не колебательный характер. Это подтверждается и численными расчетами [21]. При численном интегрировании требуется шаг, величина которого определяется из условия  $h \cdot \text{Re}(\max|\lambda|) < 1$ , т.е.  $h < 0.001$ .

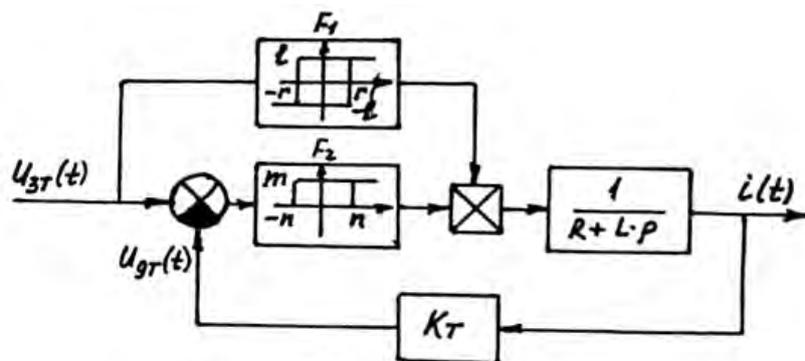


Рис. 1.68. Структурная схема токового контура с автоколебательным однополярным ШИП

Рассмотрим систему дифференциальных уравнений, описывающих контур тока с однополярным ШИП и ПИ-регулятором (рис. 1.68) [21]. Эти уравнения запишем в виде:

$$\begin{aligned}\frac{dX_1}{dt} &= -\frac{1}{L} \cdot (-R \cdot X_1 + (F_4 \cdot (F_2 \cdot X_3 + U_r) + F_5 \cdot (F_2 \cdot X_3 + U_r))) \cdot U_p) \\ \frac{dX_2}{dt} &= -\frac{1}{T_{dc}} \cdot X_2 + \frac{K_{dc}}{T_{dc}} \cdot X_1, \\ \frac{dX_3}{dt} &= \frac{K_{rc}}{T_{dc}} \cdot U_{zc} + T_{rc} \cdot \frac{dU_{zc}}{dt} + K_{rc} \cdot \frac{T_{rc}}{T_{dc}} \cdot (X_2 - K_{dc} \cdot X_1)\end{aligned}$$

Правые части этих уравнений обозначим соответственно через  $f_1, f_2, f_3$ .

Якобиан  $J = \frac{\partial f}{\partial X}$  этой системы ДУ (здесь  $f = (f_1, f_2, f_3)$ , а  $X = (X_1, X_2, X_3)$ )

можно записать:

$$J = \begin{pmatrix} -\frac{R}{L} & 0 & U_p \cdot \left[ \frac{\partial F_4}{\partial U} \cdot \frac{\partial U}{\partial F_2} \cdot \frac{\partial F_2}{\partial X_3} + \frac{\partial F_5}{\partial U} \cdot \frac{\partial U}{\partial F_2} \cdot \frac{\partial F_2}{\partial X_3} \right] \\ \frac{K_{dc}}{T_{dc}} & -\frac{1}{T_{dc}} & 0 \\ \frac{K_{rc} \cdot T_{rc}}{T_{dc}} \cdot K_{dc} & \frac{K_{rc} \cdot T_{rc}}{T_{dc}} & 0 \end{pmatrix}$$

Обозначим  $A = U_p \cdot \left[ \frac{\partial F_4}{\partial U} \cdot \frac{\partial U}{\partial F_2} \cdot \frac{\partial F_2}{\partial X_3} + \frac{\partial F_5}{\partial U} \cdot \frac{\partial U}{\partial F_2} \cdot \frac{\partial F_2}{\partial X_3} \right]$ . Тогда определитель

Якобиана  $|J| = \frac{2 \cdot K_{rc} \cdot T_{rc} \cdot K_{dc}}{T_{dc}^2} \cdot A$ . Для имеющихся параметров контура тока

коэффициент  $A$  может быть как нулем, так и принимать очень большие значения. Результат вычисления выражения перед коэффициентом  $A$  имеет порядок  $10^5$ . Следовательно, и определитель Якобиана  $|J|$  может принимать значения равные нулю или очень большому числу. Собственные числа Якобиана для параметров данной схемы могут принимать следующие значения:  $\lambda_{1,2} = -600 \pm 1000 \cdot i$ ,  $\lambda_3 = 0$ . Решения этой системы имеют колебательный характер. Это также подтверждается численными расчетами [6].

Рассмотрим теперь математическую модель электропривода традиционной схемы, уравнения которой приведены в [21, 22]. Якобиан этой системы уравнений имеет следующий вид:

$$|J| = \begin{vmatrix} -\frac{1}{J_a} \cdot \frac{\partial F_6}{\partial X_1} & \frac{1}{J_a} \cdot K_m & 0 & 0 & 0 & 0 \\ -\frac{1}{J_a} \cdot K_\omega & -\frac{R}{J_a} & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & -\frac{1}{K_{PT_2}} \cdot T_{PT_2} & a_{34} & a_{35} & a_{36} \\ 0 & \frac{K_{DT}}{T_{DT}} & 0 & -\frac{1}{T_{DT}} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & -1 & a_{46} \\ \frac{K_{DC}}{T_{DC}} & 0 & 0 & 0 & 0 & -\frac{1}{T_{DC}} \end{vmatrix}$$

$$a_{23} = U_p \cdot \left( \frac{\partial F_4}{\partial U} \cdot \frac{\partial U}{\partial F_2} + \frac{\partial F_5}{\partial U} \cdot \frac{\partial U}{\partial F_2} \right), \quad a_{24} = U_p \cdot \left( \frac{\partial F_4}{\partial U} \cdot \frac{\partial U}{\partial F_2} - \frac{\partial F_5}{\partial U} \cdot \frac{\partial U}{\partial F_2} \right), \quad a_{25} = 0,$$

$$a_{26} = U_p \cdot \left( \frac{\partial F}{\partial U} \cdot \frac{U}{\partial F_2} + \frac{\partial F_1}{\partial U} \cdot \frac{\partial U}{\partial F_2} \right), \quad a_{34} = -\frac{\partial F_1}{\partial U}, \quad a_{35} = -1, \quad a_{36} = -K_u \cdot \frac{\partial F_1}{\partial U}.$$

После подстановки значений коэффициентов в Якобиан получаем  $|J| \cong 10^{10}$ . Собственные значения Якобиана на интервале интегрирования для принятых значений параметров схемы электропривода для ступенчатого управляющего сигнала могут принимать значения  $\lambda_1 = 0$ ,  $\lambda_2 = -20$ ,  $\lambda_3 = -1000$ ,  $\lambda_4 = -1$ ,  $\lambda_5 = -1000$ ,  $\lambda_6 = -40$ . Переходной процесс для электропривода данной схемы с выбранными параметрами не имеет перерегулирования и колебательности (см. [21, 22]).

При моделировании этих и других математических моделей электромеханических систем управления использовались явные одношаговые методы: методы Эйлера второго порядка, классический метод Рунге-Кутты четвертого порядка, методы Рунге-Кутты-Гилла четвертого порядка, Рунге-Кутты-Фельберга пятого порядка, Рунге-Кутты-Мерсона четвертого порядка.

### Выводы

Из используемых методов решения задачи Коши электромеханических систем наиболее эффективными методами оказались методы Рунге-Кутты-Мерсона и Рунге-Кутты-Фельберга. Метод Рунге-Кутты-Мерсона – один из самых эффективных среди явных методов типа Рунге-Кутты четвертого порядка точности. В нем используется пять вычислений правой части дифференциальных уравнений. Пятое вычисление правых частей не дает увеличение порядка до пятого, но позволяет расширить интервал устойчивости и оценить величину локальной устойчивости.

При моделировании гидравлических систем управления имеют место всё, о чём говорилось о электромеханических системах. Но в случае гидравлических систем надо иметь в виду, что такие системы

дифференциальных уравнений являются жёсткими, что налагает при моделировании таких систем дополнительные трудности при выборе метода решения таких уравнений. Следует отметить, что для жёстких систем ОДУ характерен большой разброс постоянных времени и для эффективного решения жёстких уравнений метод численного интегрирования должен позволять изменять шаг в широких пределах, сохраняя при этом вычислительную устойчивость. Тогда интегрирование можно начинать с малого шага, определённого требуемой точностью для вычисления «быстрых» компонент переходных процессов, а затем увеличивать шаг при окончании переходного процесса.

При математическом моделировании гидравлических систем управления использовались следующие методы:

1. Классический метод Рунге-Кутты четвертого порядка точности.
2. Явный метод Рунге-Кутты четвертого порядка точности с модификацией Гилла.
3. Неявный линейный многошаговый метод Хемминга типа предиктор-корректор четвертого порядка точности.
4. Метод Рунге-Кутты-Мерсона четвертого порядка точности.
5. Алгоритм DIFSUB, который использует два подкласса линейных многошаговых метода: методы Адамса-Башфорта-Мултона переменного шага и порядка  $p$  ( $p=1,2, \dots, 7$ ) и методы Гира переменного шага и порядка  $p$  ( $p=1,2, \dots, 6$ ), реализованного по схеме  $P(EC)^m$  ( $m=1,2,3$ ).
6. Методы Рунге-Кутты-Фельберга четвертого порядка точности.
7. Метод Адамса-Башфорта типа предиктор-корректор четвертого порядка точности.
8. Неявный метод Рунге-Кутты пятого порядка точности.
9. STIFF-объединяет неявные линейно-многошаговые методы типа предиктор-корректор, методы интегрирования по Адамсу до первого порядка точности и по Гиру до шестого порядка точности. Выбор и смена порядка метода и шага интегрирования осуществляется автоматически.

Результаты математического моделирования показали, что из рассмотренных методов наиболее эффективным оказался неявный метод Рунге-Кутты пятого порядка точности.

### **Литература**

1. Borovin G.K. Computer Simulation of Hydraulic Control System of the Walking Machine. Proceedings 2<sup>nd</sup> Tampere Intern. Conf. on Machine Automat, ICMA'98, p.p. 179-192.
2. Borovin G.K., Kostyuk A.V., Seet G. Computer Simulation of the Hydraulic Control System for Exoskeleton. Preprint KIAM RAS #79, 2004, 24 p.
3. Manessmann Rexroth, Brueninghaus Hydromatik. "Product Catalogue Axial Piston Units", Horb, Germany, 1997.

4. Mischenko V.A., Mischenko N.I. Method of Induction motor control and electric drive realizing method, UK Patent Appl., GB #2201308A, INT CL 4 HO2P 5/34, 24.10.1988, London, p. 330.
5. Mischenko V.A., Mischenko N.I., Satt och anordning for styrd en induktionsmotor, Uttagyningskrift, Soerige 456467, internationell klass 4 ho2p 5/28, 02.02.1988, Stockholm.
6. Pattel S., Hort R. Generalized Techniques of Harmonic elimination and voltage control in thyristor inverters: Part 1 – Harmonic Elimination. IEEE TRANCTIONS on industry applications, vol. IA-9, NO.3, May/June, 1973, pp. 310-317.
7. Zutec J., Abbondanti A., Norby C.I., Pulse-width Modulated Inverter Motor Drivers with Improved Modulation. IEEE TRANSACTIONS on industry applications, vol.IA-11, No.6, November/December, 1975, pp. 695-703.
8. Бахвалов Н.С. Численные методы. М.: Наука, 1973, 632 с.
9. Боровин Г.К. Математическое моделирование гидравлической системы управления шагающей машины // Препринты ИПМ им. М.В. Келдыша. 1995. №106.
10. Боровин Г.К. Нелинейности в системах управления: математические модели, особенности моделирования // Препринты ИПМ им. М.В. Келдыша. 1993. №56. 29 с.
11. Боровин Г.К., Костюк А.В. Математические модели гидравлического привода с LS-управлением шагающей машины // Препринты ИПМ им. М.В. Келдыша РАН №56, 2000.
12. Боровин Г.К., Костюк А.В. Математическое моделирование гидравлического привода с LS-управлением шагающей машины // Препринты ИПМ им. М.В. Келдыша. 2001. №54.
13. Боровин Г.К., Костюк А.В. Математическое моделирование гидравлической системы управления шагающей машины. // Труды научной школы-конференции «Мобильные роботы и мехатронные системы», Москва, МГУ, декабрь 2001, 15 с.
14. Боровин Г.К., Костюк А.В. Математическое моделирование гидросистемы шагающей машины. // Материалы XI Научно-технической конференции «Экстремальная робототехника», С-Петербург: Изд-во СПбГТУ, 2001. с. 96-106.
15. Боровин Г.К., Костюк А.В. Математическое моделирование систем управления шагающих машин // Тез. докл. Научно-техн. конф. «Гидромашины, гидроприводы и гидропневмоавтоматика». М.: МГТУ им. Н.Э. Баумана, 1999.
16. Боровин Г.К., Костюк А.В. Моделирование динамики гидропривода ноги шагающей машины // Препринты ИПМ им. М.В. Келдыша. 2002. №8. 28 с.
17. Боровин Г.К., Костюк А.В. Программный комплекс для моделирования динамики гидравлической системы управления шагающей машины // Препринты ИПМ им. М.В. Келдыша. 2002. № 46. 28 с.

- 18.Боровин Г.К., Костюк А.В., Сит Дж., Ястребов В.В. Моделирование гидравлической системы экзоскелетона. Математическое моделирование, №10, 2006, с. 39-54.
- 19.Боровин Г.К., Костюк А.В. Математическое моделирование гидравлического привода с LS-управлением шагающей машины // Препринты ИПМ им. М.В. Келдыша. 2001. № 54. 28 с.
- 20.Боровин Г.К., Мищенко В.А., Мищенко Н.И., Рутковский С.В. Математическое моделирование асинхронного электропривода с векторным управлением. // Институт прикладной математики им. М.В. Келдыша АН СССР. М. 1989, 144, 27 с.
- 21.Боровин Г.К., Рутковский С.В. Пакет прикладных программ для математического моделирования асинхронного электропривода с векторным управлением. Сб. «Программирование прикладных систем», М: Наука, 1992, 12 с.
- 22.Боровин Г.К., Рутковский С.В. Пакет программ для моделирования асинхронного электропривода с векторным управлением. // Институт прикладной математики им. М.В. Келдыша АН СССР. М. 1991, 109, 27 с.
- 23.Боровин Г.К., Слепцов В.В. Исследование контуров тока в следящих электроприводах методом математического моделирования. ИПМ им. М.В. Келдыша АН СССР, №24, 1986, 20 с.
- 24.Боровин Г.К., Слепцов В.В. Математическое моделирование электроприводов постоянного тока мехатронных систем. Межотраслевой Научно-технический сборник «Технология» серия Гибкие производственные системы и робототехника. Вып. 3-4, 1993, с. 62-68.
- 25.Боровин Г.К., Слепцов В.В., Гарипов В.К. Исследование методом математического моделирования показателей качества электроприводов постоянного тока с преобразователями серии ПРШ. ИПМ им. М.В. Келдыша АН СССР, №189, 23 с.
- 26.Боровин Г.К., Слепцов В.В., Гарипов В.К. Математическое моделирование электропривода с положительной обратной связью по скорости. ИПМ им. М.В. Келдыша АН СССР, №143, 27с.
- 27.Гельднер К., Кубик С. Нелинейные системы управления. Пер. с нем. – М.: Мир, 1987, 368 с.
- 28.Данилов Ю.А., Кирилловский Ю.Л., Колпаков Ю.Г. Аппаратура объёмных гидроприводов: рабочие процессы и характеристики. – М.: Машиностроение, 1990, 272 с.
- 29.Ковач К.П., Рац И. Переходные процессы в машинах переменного тока. М.: Госэнергоиздат, 1963. 744 с.
- 30.Кулешов В.С., Лакота Н.А. Динамика систем управления манипуляторами. М.: Энергия, 1971, 324 с.
- 31.Лещенко В.А. Гидравлические следящие приводы станков с программным управлением. М. Машиностроение, 1975, 288 с.
- 32.Медведев В.С., Потемкин В.Г. Control System Toolbox. Matlab 5 для студентов (под ред. к.т.н. В.Г. Потемкина) – М.: Диалог-МИФИ, 1999.

33. Михайлов О.П. Динамика электромеханического привода металлорежущих станков. М.: Машиностроение, 1989, 224 с.
34. Мищенко В.А. Векторный метод управления электромеханическими преобразователями. // Электротехника / 2004. № 7. с. 47-51.
35. Мищенко В.А. Теория, способы и системы векторного и оптимального векторного управления электроприводами переменного тока. Монография. Информэлектро. 2002.
36. Острем К., Виттенмарк Д. Системы управления с ЭВМ. Пер. с англ. М.: Мир, 1987, 410.
37. Охоцимский Д.Е., Девянин Е.А., Платонов А.К., Боровин Г.К., Буданов В.М., Лапшин В.В., Мирный В.М. Основные проблемы и особенности проектирования многоцелевого гидравлического шагающего шасси // Препринты ИПМ им. М.В. Келдыша. 1995. №72. 28 с.
38. Пенлеве П. Лекции о трении. М.: Гос. изд-во технико-теоретической литературы, 1954, 316 с.
39. Попов Д.Н. Динамика и регулирование гидро- и пневмосистем, М: Машиностроение, 1987, 464 с.
40. Попов Д.Н. Динамика и регулирование гидро- и пневмосистем. М.: Машиностроение, 1987. 464 с.
41. Попов Д.Н., Ермаков С.А., Лобода И.Н., Фомичёв В.М., Шумилов И.С. Инженерные исследования гидроприводов летательных аппаратов. М.: Машиностроение, 1978, 142 с.
42. Свешников В.К. Гидрооборудование: международный справочник. Номенклатура, параметры, размеры, взаимозаменяемость. В 3-х кн. М.: ООО «Изд. центр «Техинформ» МАИ». Кн. 1. Насосы и гидродвигатели. Первое издание 2001. 360 с.; второе издание 2010. 552 с. Кн. 2. Гидроаппаратура. 2002. 508 с. Кн. 3. Вспомогательные элементы гидропривода. 2003. 480 с.
43. Сипайлов Г.А., Лоос А.В. Математическое моделирование электрических машин на АВМ. М.: Высшая школа, 1980, 176.
44. Современные численные методы решения обыкновенных дифференциальных уравнений. Под ред. Д. Холла, Дж. Уатта. пер. с англ. М.: Мир, 1979.
45. Справочник по автоматизированному электроприводу. Под ред. В.А. Елисеева и А.В. Шинянского. – М.: Энергоатомиздат, 1983, 615 с.
46. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. М.: Мир, 1990.
47. Штеттер Х. Анализ методов дискретизации для обыкновенных дифференциальных уравнений. М.: Мир, 1978.

## 2. МАНИПУЛЯЦИОННЫЕ РОБОТЫ И ИХ ЗАДАЧИ

Манипуляционные роботы – обширный класс роботизированных механизмов, особенностями которого является наличие устройства, называемого манипулятором. Он используется для перемещения деталей и оборудования. Манипулятор – механизм, состоящий из последовательных или параллельных сегментов (звеньев), которые обеспечивают перемещение переносимого объекта относительно основания манипулятора. При работе звенья перемещаются друг относительно друга.

На каждом из них имеется привод для перемещения соседнего звена. Приводы могут оснащаться тормозами, которые используются при аварийном торможении и удерживании его в выключенном состоянии. Звенья нумеруются последовательно. Количество степеней подвижности манипулятора совпадает с числом звеньев. Первое звено манипулятора имеет опору, которая неподвижно закреплена на основании. Основание может быть стационарным или мобильным.

Манипулятор может быть оснащен захватным устройством (схватом), которое используется для фиксации переносимого предмета относительно последнего звена. На рис. 2.1 – 2.3 приведены примеры манипуляторов с различным числом степеней свободы.



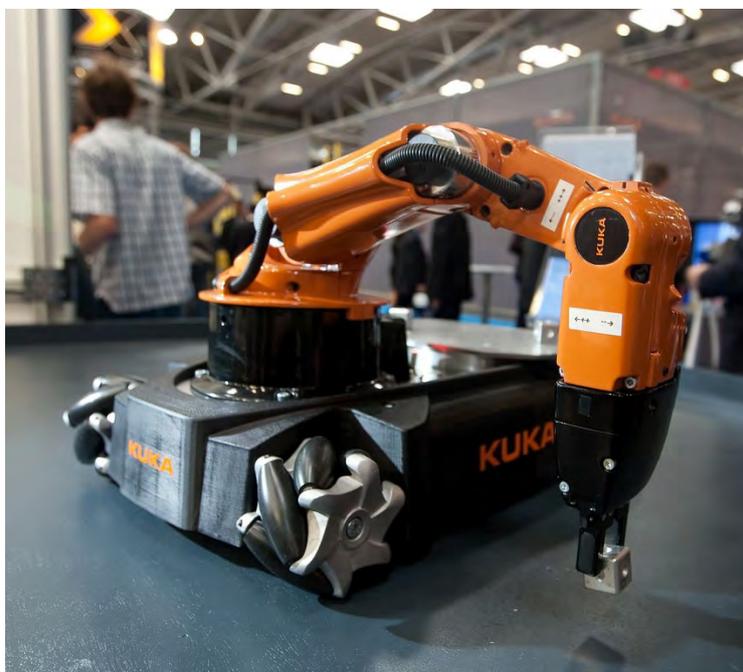
*Рис. 2.1. Робот-манипулятор фирмы FANUC\_M-10iA с пятью степенями подвижности.*

Манипуляционным роботом называется манипулятор, перемещение приводов которого осуществляется в соответствии с управляющей программой. Для этого манипуляционный робот оснащается датчиками измерения положения степеней подвижности, системой управления и блоками для управления приводами.



*Рис. 2.2.* Манипулятор фирмы YASKAWA MYS650LF с тремя степенями подвижности.

Современные манипуляционные роботы достаточно дорогие устройства. Необходимость получения прибыли от роботизированных производственных процессов вынуждает производителей искать пути снижения издержек производства манипуляционных роботов. Одним из них является универсализация конструкции роботов, которая позволяет существенно расширить область применения конкретного изделия. Универсальный манипулятор – это подкласс манипуляторов, конструкция и характеристики которого удовлетворяют техническим требованиям широкого круга возможных областей и задач применения.



*Рис. 2.3.* Оснащенный схватом манипулятор с шестью степенями свободы на мобильном основании KUKA youBot.

Концепция универсального манипулятора сложилась на раннем этапе развития робототехники в попытке найти решение стоявших перед робототехникой проблем. Современные универсальные манипуляторы достигли высочайших показателей точности (десятые доли миллиметра), скоростей перемещения (метры в секунду). Благодаря этому область их применения достаточно широка. Они используются в учебных целях, для выполнения сложных технологических процессов в разных областях науки и техники (рис. 2.4 и 2.5).

*Рис. 2.4. Универсальный робот-манипулятор UR-10.*

*Рис. 2.5. Универсальный манипулятор фирмы АВВ на сборке.*

Следует, однако, заметить, что достижение указанных характеристик невозможно без применения высокоточных датчиков и редукторов, мощных

двигателей, высокопроизводительных микроконтроллеров и электронных устройств для управления мощными приводами. Оснащение манипулятора перечисленным современным оборудованием существенно увеличивает стоимость универсальных манипуляторов. Это в свою очередь ведет к увеличению сроков окупаемости производственных процессов, в которых они используются.

Важным подклассом манипуляционных роботов является промышленные манипуляторы. Их конструктивные элементы соответствуют стандартам, предъявляемым к промышленному оборудованию. Это обеспечивает надежность и высокий ресурс работы. В современных промышленных манипуляторах наработка на отказ достигает сотен тысяч часов, что соответствует нескольким годам двухсменной работы.

Современный промышленный манипуляционный робот – автоматически управляемый манипулятор, управляющая программа которого может быть настроена на выполнение различных технологических процессов. Промышленный робот включает в себя:

- манипулятор и различные исполнительные механизмы (схват, шуруповерт, краскопульт и т.д.);
- управляющий контроллер в составе системы управления и средства его программирования, включая высокоуровневый язык управления роботом;
- включая средства для обучения: пульт ручного пульт обучения и программные средства разработки и отладки рабочей программы;
- аппаратно-программные средства считывания датчиков и управления внешним технологическим оборудованием.

## **2.1. Классификация манипуляционных роботов**

Многообразие манипуляционных роботов может быть разделено на достаточно большое количество подклассов. Рассмотрим самые крупные из них.

Манипуляционные роботы можно разделить по грузоподъемности на сверхлегкие (до 10 Н), легкие (до 100 Н), средние (до 1000 Н), тяжелые (до 10000 Н) и сверхтяжелые (свыше 10000 Н). Любопытно, что линейные размеры манипуляторов растут с увеличением их грузоподъемности, т.е. манипулятор большей грузоподъемности имеет больший размер. При этом точность позиционирования современных манипуляторов большой грузоподъемности остается достаточно высокой. Например, все манипуляторы фирмы KUKA грузоподъемностью 1300Н в состоянии позиционировать деталь, зажатую в схвате, с точностью до 0,1 мм (рис. 2.6).

Специфические различия видов производств и вследствие этого особенности конструкции технологических устройств и программного обеспечения системы управления позволяет классифицировать манипуляционные роботы по принадлежности тому или иному производству.

В промышленности можно встретить сварочные манипуляторы, манипуляторы для механической обработки, сборочного производства, окрасочные и транспортно-складские. Технические характеристики манипуляционных роботов, применяемых в промышленности, в значительной степени различаются. Остановимся подробнее на нескольких из них.



Рис. 2.6. Промышленный манипулятор фирмы KUKA Titan KR 1000.

В сборочном производстве важнейшим требованием к манипулятору является производительность, так как одним из условий роботизации является получение выгоды от замены ручного труда. Поэтому на сборке используют манипуляторы, которые способны быстро перемещаться (до 2 м/с), имеют высокую точность позиционирования (порядка 0,1 мм). Грузоподъемность не превышает нескольких килограммов.

В сварочном производстве используются роботы большой грузоподъемности, так как манипулятор вынужден перемещать достаточно тяжелые приспособления для сварки: сварочные клещи или сварочную головку. Важным также является размер рабочей зоны и плавность перемещения между промежуточными точками. К скорости перемещения не предъявляются существенных требований, так как дуговая сварка является достаточно медленной технологической операцией. При контактной сварке скорость перемещения между позициями ограничена ввиду большой массы сварочных клещей.

Большое распространение в промышленности получили роботы-манипуляторы для укладки и складирования готовой продукции. Требования, предъявляемые к ним, определяются условиями применения. Если целью роботизации рабочего места является высвобождение ручного труда при

погрузке и транспортировке тяжелых и габаритных грузов, то основным требованиями, предъявляемыми к манипулятору, является большая грузоподъемность и размеры рабочей зоны. При необходимости обеспечить высокую производительность раскладки продукции небольшого веса и размера, выпускаемой автоматической линией, используют высокоскоростные манипуляторы специальной конструкции, например, так называемые дельта-роботы. При сравнительно небольших величинах перемещений (до 0,5 м) этот манипулятор обеспечивает рабочий цикл длительностью порядка 1 сек. Заметим, что программа функционирования погрузочно-разгрузочных манипулятором достаточно проста, так как рабочая траектория представляет собой последовательность программных точек. Это делает возможным использование сравнительно простой системы управления с целью снижения стоимости робота.

Учебные манипуляторы – обширный класс манипуляционных роботов, которые используются в учебном процессе. Их технические характеристики достаточно разнообразны. Для ознакомления с основами робототехники могут использоваться манипуляторы, точность позиционирования которых несколько миллиметров, размер рабочей зоны до 1 м (рис. 2.7).



Рис. 2.7. Учебный робот Paskal delta отечественной фирмы Paskal.

В настоящее время наблюдается размытость технических требований к характеристикам учебных роботов. При этом спрос на них со стороны образовательных учреждений достаточно высок. Это обстоятельство объясняется двумя факторами: сложностью создания недорогого манипуляционного робота с достаточно высокими техническими характеристиками и скудностью выбора технических и программных средств для обучения молодежи технологиям манипуляционной робототехники.

Манипуляционная робототехника – быстроразвивающаяся область техники. В пользу этого утверждения говорит большое количество экспериментальных исследований по использованию манипуляторов в разных

областях человеческой деятельности и уже разработанных экспериментальных технологий, которые проходят проверку в реальных условиях эксплуатации. Остановимся на некоторых из них подробнее.

В исследованиях космоса манипуляторы используются уже достаточно давно. В качестве примера можно привести манипуляторы, установленные на Международной космической станции и на американских марсоходах Opportunity и Curiosity.

Первым медицинской манипуляционной системой, которая стала широко использоваться в медицинской практике, является хирургический робот da Vinci [1] (рис. 2.8). Он включает 4 антропоморфных манипуляторов, системы управления ими, антропоморфного пульта ручного управления в виде специализированных джойстиков, стереоскопической видеосистемы и устройств управления медицинскими инструментами внутри пациента (портами).



*Рис. 2.8.* Хирургический робот da Vinci и его система управления.

Роботизированная хирургия берет свое начало с конца 1980-х годов, когда армия США стала интересоваться разработками для проведения дистанционных хирургических вмешательств рядом с полем боя с целью уменьшения потерь личного состава от сложных ранений. Хирургическая система da Vinci была одной из первых роботизированных аппаратов в хирургии. Рабочий прототип был разработан в конце 1980-х годов в рамках контракта с армией США. Этим, возможно, определяется схожесть его манипуляторов с распространенными в то время манипуляторами фирмы Unimate PUMA-560.

На 2015 год построено более 3000 таких аппаратов. Стоимость системы «da Vinci» 2 млн долл. В США на июль 2014 года клиники располагали 2153

системами «da Vinci». Основная проблема роботизированной хирургии – высокая стоимость операций. Она обуславливается высокой стоимостью роботизированных систем и эксплуатационных расходов. Американские специалисты утверждают, что рентабельность «Да Винчи» может быть достигнута только, если в клинике в течение шести лет ежегодно проводится 150–300 операций с применением этой системы. При этом однозначные показания для применения этой системы имеются только для ограниченного круга операций (гистерэктомия, миомэктомия и радикальной простатэктомии), т.е. ее использование целесообразно только в крупных специализированных отделениях стационаров.

С позиций настоящего времени использование универсальных манипуляторов представляется не самым удачным решением не только ввиду их высокой стоимости, но и технологической сложности, громоздкости и потенциальной опасности для пациента. Альтернативное решение, представленное отечественной группой исследователей [2], в основе которого лежит специализированное манипуляционное устройство, частично решающее отмеченные проблемы. Та же точность манипулирования хирургическими инструментами достигается в нем с помощью оборудования, которое умещается в чемодане, и его конструкция является более простой, вследствие чего можно рассчитывать на существенное уменьшение стоимости хирургического робота.

Уровень разработки других робототехнических систем еще не достиг состояния, которое позволяет использовать их в повседневной медицинской практике. К ним относятся робот R2D2 (Robotic Retinal Dissection Device) для операций на глазах, робот ARTAS для операций по пересадке волос, роботы для стоматологии и массажный робот и др. Это позволяет заключить, что медицинская робототехника находится еще в стадии становления. Группы исследователей в разных странах ведут активный поиск направлений применения манипуляционной техники в медицине и накопление опыта использования разработанных систем.

Еще одним направлением развития современной робототехники являются сервисные роботы. Они представляют собой манипулятор на подвижном шасси. Мобильное основание позволяет доставить манипулятор к месту работы. Сервисные роботы необходимы при автоматическом и дистанционном обслуживании опасных объектов или, если человек не в состоянии приблизиться к месту работы, например, для дистанционного исследования Луны и планет, для помощи инвалидам или обслуживании человека в домашних условиях и выполнения домашней работы. Во всех этих применениях подвижность манипулятора существенно расширяет его рабочую зону и позволяет обойтись одним устройством вместо нескольких.

Армейские подразделения саперов для разминирования используют манипуляторы на колесном или гусеничном шасси. Машины управляются оператором дистанционно на безопасном расстоянии. Манипуляторы оснащаются различными вспомогательными устройствами, позволяющими

разминировать опасные предметы или уничтожить их содержимое на месте (рис. 2.9, 2.10).



*Рис. 2.9.* Робот-сапер обезвреживает опасный предмет.



*Рис. 2.10.* Робот для разминирования «Богомол».

Часть исследовательских групп работает над проектами дистанционным использованием сервисных роботов для обслуживания техники или в чрезвычайных ситуациях (рис. 2.11).



*Рис. 2.11.* Мобильный манипулятор для тушения пожаров.

Большинство разработок в области сервисных манипуляторов не вышли из стадии исследований. Значительные усилия разработчиков направлены на создание коммерческих продуктов для использования в домашней обстановке, барах, кафе или ресторанах (рис. 2.12).



*Рис. 2.12.* Робот-бармен от фирмы KUKA.

Мобильное шасси увеличивает неопределенность в знании взаимного положения обслуживаемого устройства и манипулятора, что существенно усложняет создание алгоритмов для ремонта и сервисного обслуживания. Кроме этого, к традиционным задачам управления манипулятором добавляются задачи навигации и определения местоположения и ориентации шасси.

## 2.2. Кинематические схемы манипуляторов

Сложность реализации заданных характеристик манипуляционного робота зависит от способа соединения его соседних звеньев. В теории механизмов и машин он называется кинематической парой. Совокупность кинематических пар составляет кинематическую схему манипулятора. В манипуляционной робототехнике сформировался определенный набор таких схем, который по мере развития производственных возможностей техники постепенно расширяется.

Каждая из кинематических схем имеет свои достоинства и недостатки, которые учитываются разработчиками технологических процессов при выборе манипулятора и конструировании производственного процесса. Традиционно название кинематической схемы манипулятора совпадает с названием системы координат, в которой происходит транспортное перемещение рабочего органа манипулятора.

### 2.2.1. Ортогональная кинематическая схема

Манипулятор с ортогональной кинематической схемой – манипулятор, транспортные степени подвижности которого перемещаются вдоль осей прямоугольной системы координат.

Примеры роботов с ортогональной кинематической схемой приведен на рис. 2.13. Ортогональная кинематическая схема позволяет обеспечить одинаковую точность позиционирования во всех точках рабочей зоны манипулятора. Ее достоинством является возможность достаточно просто реализовать рабочую зону большого размера хотя бы по одному или даже нескольким направлениям.

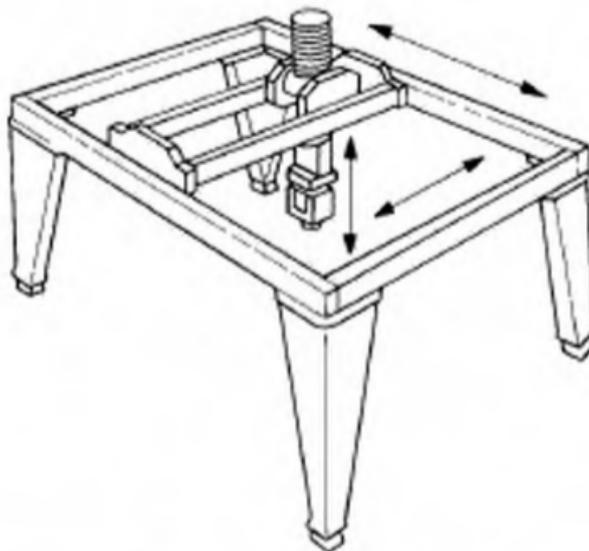


Рис. 2.13. Манипулятор с ортогональной кинематической схемой.

Традиционным недостатком манипулятора с ортогональной кинематической схемой является ограниченность ускорения и скорости

движения, так как первая степень подвижности перемещает *большую* часть массы всего манипулятора, которая для поступательных степеней подвижности оказывается сравнительно *большой*. Существенным недостатком является также их *большие* размеры манипуляторов.

### 2.2.2. Цилиндрическая кинематическая схема

Манипулятор с цилиндрической кинематической схемой – это манипулятор транспортное перемещение которого осуществляется одной вращательной и двумя поступательными степенями подвижности. Примеры таких манипуляторов приведены на рис. 2.14.

Основными преимуществами манипуляторов с кинематической схемой является компактность и высокая точность управления по двум координатам.

Транспортная скорость перемещения таких манипуляторов существенно выше, чем у манипуляторов с ортогональной кинематической схемой. Это связано с тем, что при повороте скорость рабочего органа тем больше, чем дальше выдвинута поступательная степень подвижности (часто называемая стрелой). При этом каждая из поступательных степеней подвижности перемещает только часть массы манипулятора.

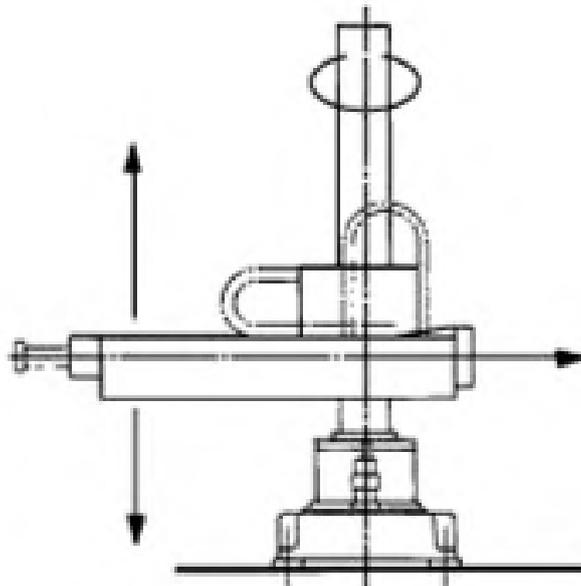


Рис. 2.14. Манипулятор с цилиндрической кинематической схемой.

Недостатки такой схемы являются обратными продолжением их достоинств. Точность позиционирования зависит от того, насколько далеко выдвинута стрела. Рабочая зона таких манипуляторов ограничена размерами стрелы, так как во втянутом положении перемещениям может мешать ее задний срез.

### 2.2.3. Сферическая кинематическая схема

Манипулятором со сферической кинематической схемой называют манипулятор, транспортные перемещения которого осуществляются двумя вращательными степенями подвижности с общим центром вращения и одной поступательной степенью подвижности (рис. 2.15).

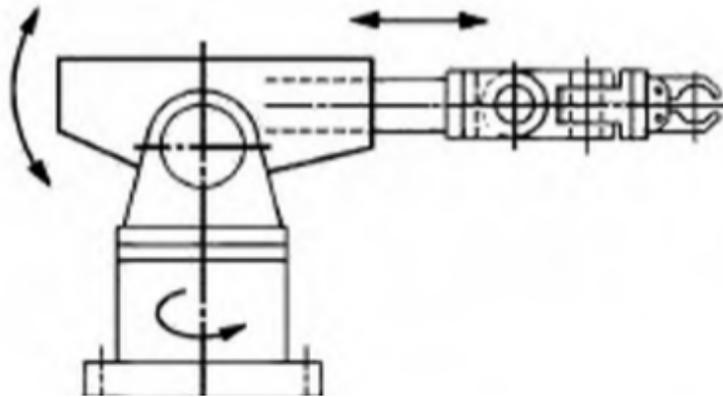


Рис. 2.15. Манипулятор со сферической кинематической схемой.

В настоящее время манипуляторы со сферической кинематической схемой встречаются достаточно редко. Основным недостатком сферической кинематической схемы является ограниченность рабочей зоны в горизонтальной плоскости и вследствие этого сложность организации производственного процесса, в котором заготовки располагаются в одной плоскости.

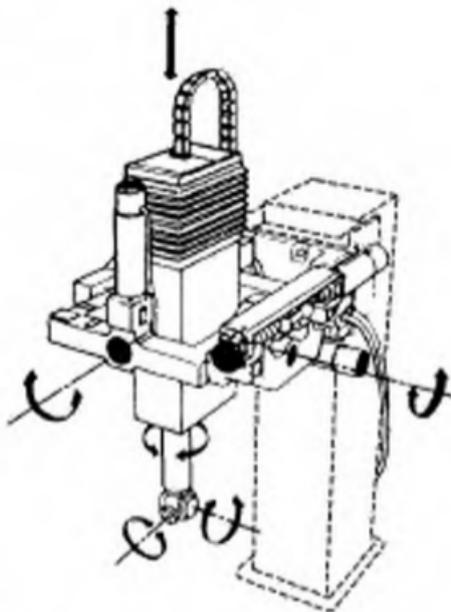


Рис. 2.16. Манипулятор маятникового типа.

Достаточно известной разновидностью манипуляторов со сферической кинематической схемой является манипуляторы маятникового типа (рис. 2.16).

На первом этапе развития робототехники эти манипуляторы обеспечивали рекордную скорость транспортных перемещений, и, поэтому, в первую очередь использовались для роботизации сборочных процессов. Высокая скорость перемещений достигалась за счет минимизации статического момента путем точной балансировки манипулятора относительно центра вращения.

Однако малые размеры рабочей зоны в горизонтальной плоскости затрудняло реализацию сборки более, чем одной детали и объективно вело к удорожанию сборочного процесса. По этой причине существенного распространения манипуляторы такой конструкции не получили.

#### 2.2.4. Антропоморфная кинематическая схема

В антропоморфной кинематической схеме для транспортных перемещений используются первые три вращательных степени подвижности. Оси второй и третьей степеней параллельны и перпендикулярны оси вращения первой степени (рис. 2.17).

В настоящее время это самая распространенная кинематическая схема манипуляционных роботов. Ее преимуществом является возможность сориентировать ось последнего звена под любым заданным углом к детали в рабочей зоне достаточно большого размера. Во многих приложениях эта возможность является решающей при выборе манипулятора.

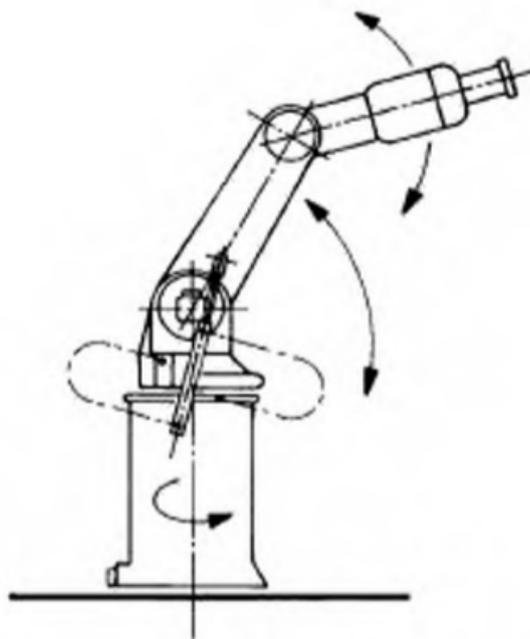


Рис. 2.17. Антропоморфный манипулятор.

Первым промышленным манипулятором, в котором была реализована антропоморфная кинематическая схема был манипулятор фирмы Unimate Puma-560 (рис. 2.18). Он имеет грузоподъемность 2.5 кг. Диаметр его рабочей зоны составляет почти 1.5 м. Максимальная скорость перемещения 1 м/с. Точность позиционирования до 0.2 мм. Система управления позволяет перемещать зажатую в хвате деталь по любой прямолинейной траектории, которая не выходит за границы рабочей зоны.

В целом удачная конструкция манипулятора и системы управления объясняют популярность этих роботов в научных коллективах, которые занимаются исследованиями роботизированных технологий. Даже по прошествии почти 30-ти лет их технические характеристики находятся на уровне лучших образцов современных манипуляторов с антропоморфной кинематической схемой.

Любопытно, однако, что робот Puma-560 в незначительной степени использовался для автоматизации работ в промышленности. Это объясняется достаточно высокой стоимостью (примерно \$100 000 в ценах 1990 годов) при медленной динамике разгона по сравнению с манипуляторами, выпускаемыми в те годы. В результате сроки окупаемости производственных процессов с его использованием оказывались слишком большими.



Рис. 2.18. PUMA-560 с системой управления (1983 г.).

В нашей стране представителем семейства роботов Puma-560 является манипулятор РМ-01, основным отличием которого является отечественная стойка управления Сфера-36.

### 2.2.5. Робот типа SCARA

Главным конкурентом манипулятора Puma-560 в задачах роботизации производственных процессов был манипулятор с оригинальной кинематической схемой SCARA, который в 1981 году был разработан совместными усилиями японских фирм Sankyo Seiki, Pentel и NEC. Он имеет две или три вращательных степени подвижности и одну поступательную, направление перемещения которой совпадает с осями вращательных степеней (рис. 2.19).

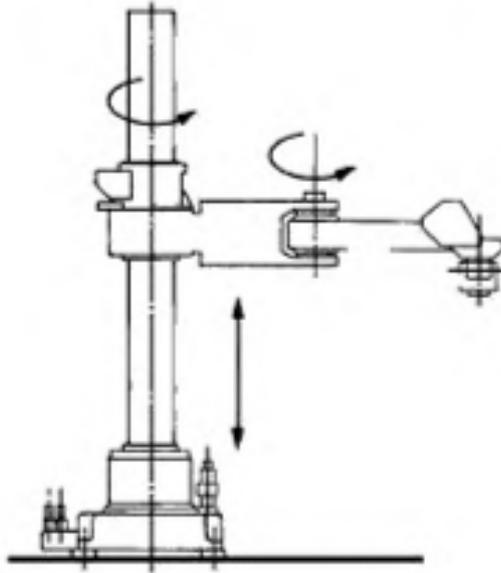


Рис. 2.19. Манипулятор с кинематической схемой SCARA.

Использование двух вместо трех вращательных степеней подвижности оправдано для уменьшения стоимости манипулятора и увеличения динамики разгона в задачах, в которых не требуется ориентация конечного звена. Достаточно часто вертикальное перемещение выполняется пневматическим приводом, что способствует существенному уменьшению момента инерции манипулятора при транспортировке легких деталей.

В результате стоимость облегченных версий манипуляторов SCARA оказывалась в 2-3 раза меньше стоимости Puma-560 при сравнимых размерах рабочей зоны. Ускорение разгона больше почти в два раза, причем, если расположение и ориентация деталей на рабочем столе позволяла использовать в транспортных перемещениях одновременное вращение степеней подвижности в одном направлении, то максимальная скорость движения схвата могла быть увеличена еще в 1.5 раза.

В сборочных процессах и транспортировке легких деталей SCARA вытеснила манипулятор Puma-560.

Достоинством кинематической схемы манипулятора SCARA является простота конструкции, которая обеспечивает высокую скорость транспортных перемещений в горизонтальной плоскости. Недостатком является малые размеры рабочей зоны таких манипуляторов, если требуется осуществлять

перемещения параллельно декартовым осям и то, что выигрыш в скорости транспортных перемещений достигается только, если детали специальным образом расположены и сориентированы на рабочем столе. **2.2.6. Манипуляторы с параллельной кинематической схемой**

Все рассмотренные выше кинематические схемы являются последовательного типа, так как звенья манипуляторов располагаются одно за другим. В 1987 году швейцарская фирма Demagex получила лицензию на выпуск манипулятора типа Дельта-робот, в котором в транспортных перемещениях одновременно участвуют все степени подвижности (рис. 2.20). Кинематическая схема использует только вращательные пары, что упрощает конструкцию и позволяет достичь высокой динамики разгона при транспортировке легких грузов.

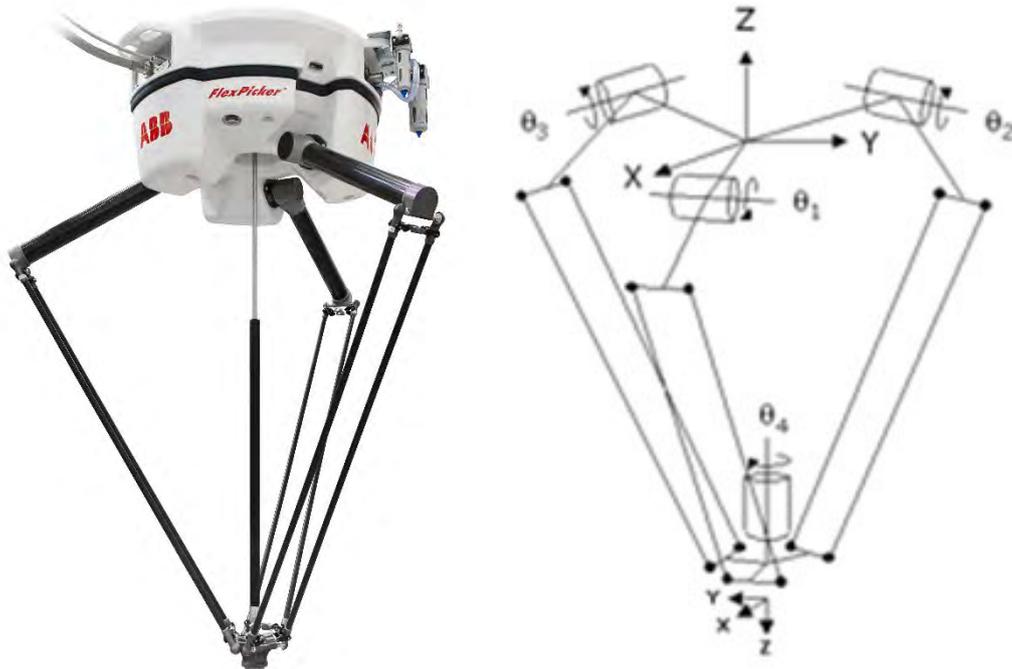


Рис. 2.20. Дельта-робот FlexPicker фирмы АВВ и его кинематическая схема.

Однако только в 1999 году шведская фирма АВВ смогла разрешить все вопросы, связанные с точностью перемещений, выполняемых с высокой скоростью в нелинейной кинематической схеме робота (рис. 2.21). Рабочий цикл, включающий перемещение к детали, захват ее, перемещение к месту сборки и освобождение детали, составляет порядка 1 с.

Недостатком таких манипуляторов является ограниченность рабочей зоны, которая не превышает 0,5 м. По этой причине они находят применение на конвейерных линиях в операциях упаковки мелких предметов, имеющих достаточно простую форму.

Следует заметить, что манипуляторы с параллельными кинематическими схемами появились совсем недавно и, поэтому в этом классе

устройств можно ожидать появление кинематических схем с новыми полезными свойствами.

### **2.3. Операционная среда функциональных задач применения манипуляторов**

Для того, чтобы описать пространство, внутри которого перемещается манипулятор, используют понятие рабочей зоны (рабочего пространства). В настоящее время трактовка этого термина зависит от того, кто его применяет.

Производитель манипуляторов предпочитает говорить о пространстве, ометаемом некоторой выделенной точкой последнего звена (которую называют референсной) при перемещениях всех степеней подвижности в допустимых диапазонах. В этом случае оказывается, что рабочее пространство меньше, чем пространство, ометаемое всеми движущимися деталями манипулятора, схватом и зажатым в нем предметом.

Разработчики производственного приложения и те, кто им пользуется, предпочитают понимать под рабочей зоной совокупность точек, которую может занять манипулятор и переносимые им предметы в процессе работы. Смысл этой трактовки сочетает в себе понятие области, нахождение в которой опасно для персонала, и одновременно реально достижимую область, размеры которой характеризуют возможности технологического процесса.

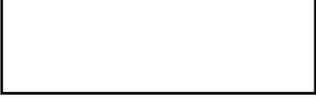
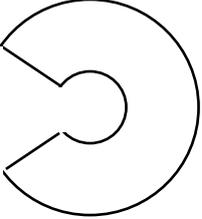
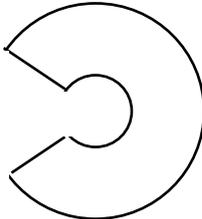
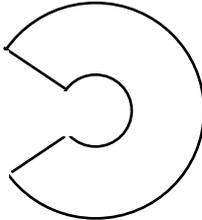
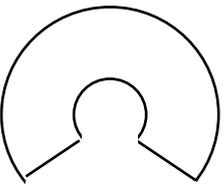
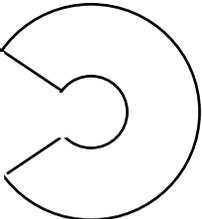
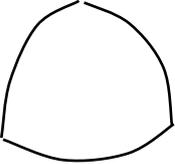
В таблице 2.1 показаны сечения рабочих зон горизонтальной и вертикальной плоскостями в осях ОХУ для манипуляторов с рассмотренными кинематическими схемами.

#### **2.3.1. Организованное и неорганизованное рабочее пространство**

Сложность разработки технологического роботизированного процесса определяется тем обстоятельством, насколько точно известно положение объектов, с которыми предстоит работать манипулятору. Для того, чтобы характеризовать условия разработки, применяют термины организованное и неорганизованное рабочее пространство.

Организованным рабочим пространством называют пространство, в котором системе управления известны координаты всех элементов среды. В противном случае рабочее пространство называют неорганизованным.

Таблица 2.1.

Кинематическая схема	Сечение рабочей зоны в горизонтальной плоскости	Сечение рабочей зоны в вертикальной плоскости
<b>Ортогональная</b>		
<b>Цилиндрическая</b>		
<b>Сферическая</b>		
<b>Маятниковая</b>		
<b>Антропоморфная</b>		
<b>SCARA</b>		
<b>Дельта-робот</b>		

Считается, что в роботизированных промышленных процессах рабочее пространство организовано и неорганизованное рабочее пространство встречается только в реальной обстановке в работе сервисного робота. Однако это не так.

Создание высокоорганизованного рабочего пространства – сложная и достаточно затратная задача. В большинстве случаев для этого приходится разрабатывать новый технологический процесс. Но даже в этом случае остаются технологические операции, которые проще и дешевле реализовать путем оснащения робота адаптивными устройствами. Такой процесс по определению является частично организованным.

Требования к точности положения объектов в рабочей зоне можно существенно ослабить, если использовать сложные адаптивные алгоритмы технологических операций. Это было показано в работах по многооперационной сборке, выполненных под руководством академика Д.Е. Охоцимского, проф. А.К. Платонова сотрудниками Института прикладной математики С.С. Камынина, Е.И. Кугушева и В.А. Карташева [9].

Перед началом сборки роботу было известно приближенное положение всех деталей, причем без адаптивных движений сборка вообще не могла быть выполнена, некоторые детали перед сборкой могли отсутствовать или неверно лежать. Робот самостоятельно распознавал все возможные ситуации и выполнял сборку правильно, либо, если это было невозможно, останавливался и информировал оператора о причине остановки. Упомянутые работы демонстрируют возможность успешной работы сборочного робота в плохо организованной среде при использовании сложных интеллектуальных алгоритмов управления.

Если рабочая среда робота не организована, то требуются дополнительные усилия по выяснению состояния среды перед началом работы управляющего алгоритма. После этих действий среду можно уже называть частично организованной.

Приведенные выше рассуждения показывают, что степень организованности среды не является абсолютным свойством среды, а характеризует сложность технических решений и алгоритмов, которые потребуется разработать для того, чтобы обеспечить выполнение поставленной задачи и вероятность успешного ее выполнения.

#### **2.4. Интеллектуальные системы управления манипуляционными роботами с элементами искусственного интеллекта**

Управление движениями манипуляционного робота осуществляется системой управления – устройством, которое обеспечивает движение манипулятора по заданной траектории, создание и исполнение программы функционирования робота, взаимодействие с оператором. Различия в особенностях решения каждой из перечисленных задач выражается в том, что

аппаратно система управления реализуется в виде двух физических уровней верхнего и нижнего (рис. 2.21).

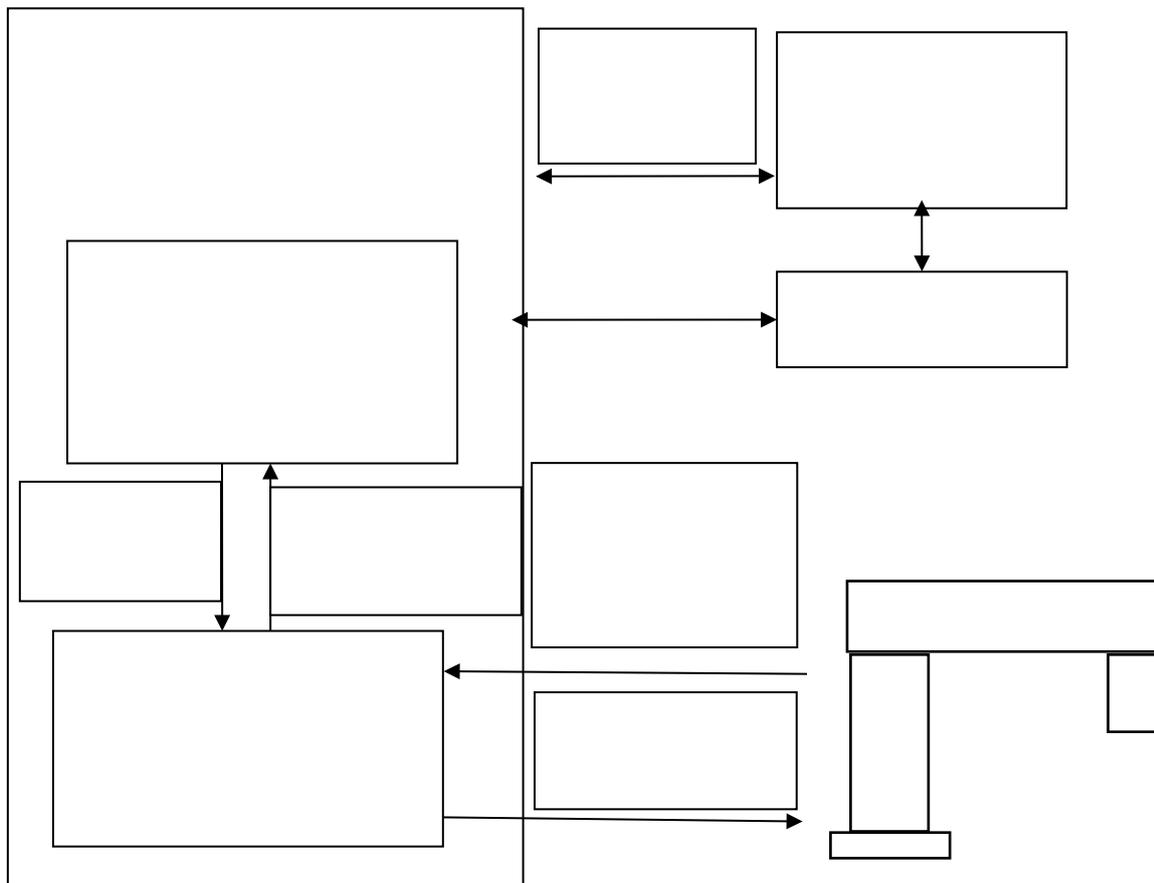


Рис. 2.21. Блок-схема системы управления роботом-манипулятором.

Традиционно нижний уровень управления отвечает за вычисление координат степеней подвижности, управление двигателями и взаимодействие с верхним уровнем по шине данных. Эти функции выполняются в масштабе реального времени. Для того, чтобы обеспечить достаточно высокую точность движения по программной траектории вычисление уставок напряжения необходимо производить с частотой порядка 1 кГц. Поэтому основным ограничением на сложность управляющей программы следящей системы выступает производительность контроллера.

Верхний уровень управления осуществляет взаимодействие с нижним уровнем по шине данных, обеспечивает управление процессами, связанными с обучением и исполнением программы функционирования робота, обычно называемой рабочей программой. Она включает в себя команды движений в целевые точки, условия перемещения в них, которые учитывают показания датчиков и состояния органов правления технологическим процессом.

Перед каждым перемещением в целевую точку система управления вычисляет промежуточные точки траектории и передает их нижнему уровню.

Для решения задач верхнего уровня управления в наибольшей степени подходят аппаратные и программные возможности ЭВМ типа IBM PC. По этой причине этот компьютер часто используется в качестве управляющей ЭВМ верхнего уровня.

Создание управляющей программы для робота называется обучением.

Возможности программиста по удобству разработки и сложности алгоритма управления определяются способом представления рабочей программы.

В простейшем случае также, как и в станках с числовым программным управлением (ЧПУ), рабочая программа представляется последовательностью кадров. Каждый кадр задает команду и параметры движения или переход по некоторому условию на кадр с заданным номером.

Популярность покадрового представления программы основана на минимальных требованиях к аппаратной части системы управления, при которых такое представление реализуемо. При этом такая форма представления программы не ограничивает функциональные возможности манипулятора, так допускает использование сложных специализированных адаптивных движений, которые задаются кадром.

Сложные системы управления предоставляют пользователю более комфортные условия работы на языке управления роботом. Он представляет собой универсальный язык программирования, который расширен командами задания движения и считывания показаний датчиков. Основным преимуществом использования текстуального описания действий манипулятора является возможность создавать новые алгоритмы движений и накапливать их в виде подпрограмм. Это существенно упрощает разработку программ технологических движений.

Последовательность разработки таких программ следующая. Сначала на языке управления роботом описывается алгоритм выполнения технологической операции в виде последовательности команд движения и условий их окончания. Целевые точки перемещений манипулятора обозначаются идентификаторами. Координаты промежуточных точек вычисляются из выражений, в которые входят координаты целевых точек. Такой способ разработки программы движений называется аналитическим программированием.

Координаты целевых точек определяются при позиционном обучении. Для этого используется пульт ручного управления, который представляет собой устройство, управляющие органы которого позволяют пользователю перемещать степени подвижности в нужном направлении. Координаты целевой точки запоминаются при обращении на пульте управления к команде чтения координат степеней подвижности.

### **2.4.1. Унифицированные алгоритмы функций нижнего уровня управления и их развитие**

Основная задача нижнего уровня системы управления является выведение всех степеней подвижности в заданную программную позицию. По этой причине программный блок, реализующий эту функцию называют следящей системой. В соответствии с современными представлениями теории автоматического регулирования чем выше частота работы алгоритма следящей системы, тем точнее отслеживается заданное движение. По этой причине каждая степень подвижности управляется своим контроллером.

Управляющая программа каждой степени подвижности обеспечивает считывание показаний датчика положения, вычисление текущей координаты и скорости, вычисление и выдачу управляющего напряжения (формировании широтно-импульсного сигнала), подаваемого на двигатель, обмен информацией с верхним уровнем по шине данных, выведение степени подвижности в начальное положение (обычно эта операция называется калибровкой), контроль достижимости программной точки (рис. 2.22).

Следящее управление реализуется с помощью ПИД-регулятора. Для манипулятора РМ-01 точность выведения в целевую позицию достигает 0.2 мм. Точность отслеживания программной траектории зависит от скорости движения по ней. При движении схвата со скоростью 1 м/с ошибка не превышает нескольких миллиметров. Если схват движется со скоростью 0.1 м/с, отклонение от программной траектории уменьшается до нескольких десятых миллиметра.

В промышленных роботах установка параметров ПИД-регулятора недоступна для пользователя. Траектория движения в целевую точку и изменение скорости вдоль нее полностью определены настройками регулятора.

При позиционной системе управления движение в целевую точку осуществляется с максимальной скоростью, величина которой аппаратно и программно ограничивается пользователем. Такой способ управления востребован в том случае, когда требуется сократить до минимума длительность транспортных перемещений. Траекторию перемещения задать нельзя. Однако, она прогнозируется с достаточной степенью точности, если известны величины перемещений по каждой степени подвижности. При определенном навыке пользователь имеет возможность формировать достаточно сложные непрямолинейные траектории.

При траекторном управлении пользователь может задать движение схвата манипулятора в целевую точку по прямой. Сложные траектории представляют собой ломаные линии, вершины которых – промежуточные точки, задаваемые пользователем. Система управления самостоятельно осуществляет плавный переход к движению в очередную точку.

Таким образом, штатный способ задания движения промышленных манипуляторов существенно ограничивает точность реализации перемещений, в которых траектория и скорость движения по ней рассчитаны

заранее. К таким перемещениям относятся, например, любые траектории, полученные путем компьютерного моделирования.

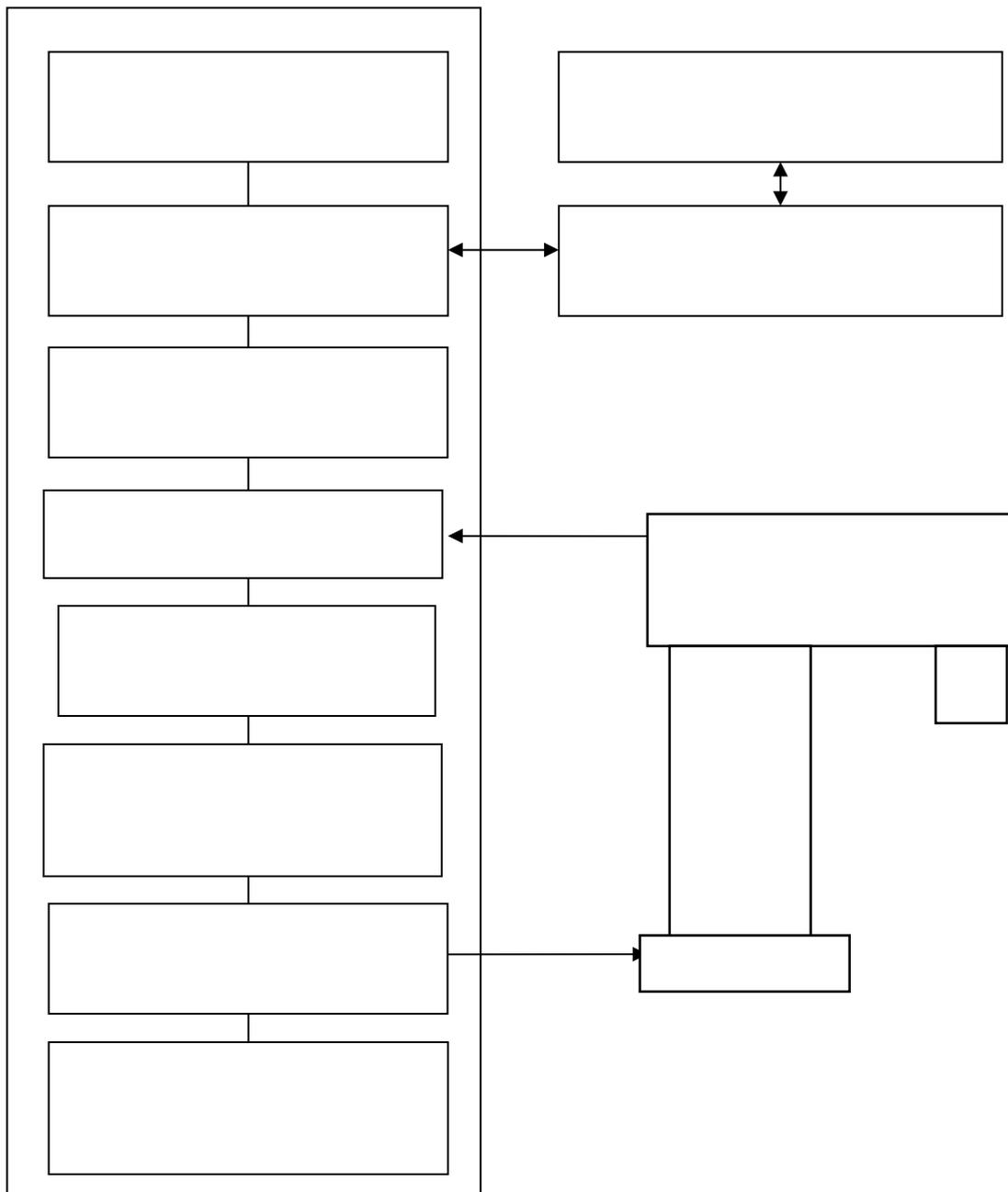


Рис. 2.22. Функции нижнего уровня управления.

Помимо функции приведения манипулятора в заданную точку система управления нижнего уровня осуществляет функцию контроля достижимости программной точки. Эта функция безопасности перемещений манипулятора позволяет экстренно останавливать его, при наталкивании на непредвиденное препятствие. В этом случае достичь программную точку невозможно. Если координата не меняется некоторый заданный промежуток времени, то следящая система формирует признак о наступлении аварийной ситуации и

передает его верхнему уровню управления. При поступлении указанного сигнала система управления верхнего уровня отключает питание приводов, тем самым принудительно останавливая робот.

Описанный способ обеспечения безопасности движения используется во всех промышленных манипуляторах. Существенный недостаток его состоит в том, что он неявно предполагает исправность всех элементов системы управления движениями степеней подвижности. Например, при некоторых неисправностях датчика положения реальное движение манипулятора будет отличаться от программного.

В работе [3] проанализированы последствия различных отказов электронных компонентов системы управления движением степенью подвижности. Было показано, что для контроля исправности датчика положения можно использовать информационную избыточность способа измерений положения звена манипулятора.

Например, для энкодера проверка на изменение значений датчиков положения состоит в том, что при перемещении в новую целевую точку в течение любого промежутка времени заранее заданной длительности показания датчика изменятся хотя бы один раз. Это позволяет исключить неисправности, при которых показания датчика постоянны, что имеет место, например, при неподключенном разъёме.

Если установлено, что показания датчика изменяются, то проводится их анализ на отсутствие ошибочных значений. Осуществляется он следующим образом.

Каждый канал инкрементного датчика принимает значения, которые можно обозначить как 0 и 1. Составим пару  $x_1x_2$  из показаний первого и второго каналов, где  $x_1$  и  $x_2$  равны 0 либо 1.

У исправного датчика при очередном измерении код либо не изменится, либо изменится значение только одного разряда, т.е. допустимыми являются коды

$$x_1x_2, \bar{x}_1x_2, x_1\bar{x}_2.$$

Здесь через  $\bar{x}_i$  обозначено противоположное к  $x_i$  значение. Например, для пары (0,1) допустимыми являются пары (0,1), (1,1), (0,0).

Появление кода  $\bar{x}_1\bar{x}_2$  свидетельствует об ошибке, так как он отличается от исходного  $x_1x_2$  сразу в двух разрядах. В приведённом примере ошибочным является код (1,0). При появлении ошибочного кода манипулятор должен быть остановлен, так как в этот и все последующие моменты времени вычисляемое положение, которое манипулятор занимает в пространстве, не соответствует его истинному положению.

При обнаружении ошибок в измерении координат степеней подвижности система управления должна немедленно остановить манипулятор.

Если датчики положения исправны, то их показания могут использоваться для выявления неисправностей электронных компонент

системы управления движением степеней подвижности. Алгоритмы таких проверок описаны в процитированной работе.

Контроль исправности работоспособности электронных компонент системы управления существенно увеличивает надежность работы манипуляционного робота по сравнению со штатной системой безопасности, так как по мере износа возрастает вероятность отказов оборудования. Отказы представляют опасность для обслуживающего персонала. Неприятные последствия возникают также от вызванных ими поломок производственного оборудования. Восстановление его после аварии требует не только дополнительных трудозатрат персонала, но и препятствует выпуску продукции, так как останавливает производственный процесс.

#### **2.4.2. Унифицированные алгоритмы функций верхнего уровня управления и их развитие**

Верхний уровень управления выполняет следующие функции: поддерживает язык управления роботом или систему заданий движений на основе кадров, выполняет рабочую программу, вычисляет целевые и промежуточные точки траекторий перемещения схвата, обеспечивает перемещение манипулятора от ручного пульта и запоминание целевых точек движения, поддерживает обмен информацией с нижним уровнем управления, реализует функции безопасности перемещения манипулятора, взаимодействие с пользователем при обучении робота и исполнении рабочей программы (рис. 2.23). Рассмотрим особенности реализации некоторых из перечисленных функций.

**Функции безопасности.** Функция безопасности обеспечивает экстренную остановку манипулятора по команде пользователя или автоматически при возникновении аварийной ситуации. Пользователю доступны несколько вариантов торможения.

1. Аварийный останов. Манипулятор тормозится путем отключения питания приводов. При этом срабатывают тормоза тех степеней подвижности, которые оснащены ими. Выбег степеней подвижности при торможении с помощью тормозов в несколько раз меньше, чем обеспечивают двигатели.

В результате аварийного торможения двигатели обесточиваются. Дальнейшее исполнение программы невозможно, так как требуется перезапуск системы управления. Основным преимуществом аварийного останова является быстрое и безопасное торможение, так как манипулятор обесточен. Траектория движения манипулятора при аварийном и управляемом останове может значительно отклоняться от программной. Например, наблюдался случай, при котором в точке останова схват манипулятора РМ-01 оказался повернутым относительно программной траектории на 30°. При длине схвата 200 мм линейное смещение составило 100 мм.

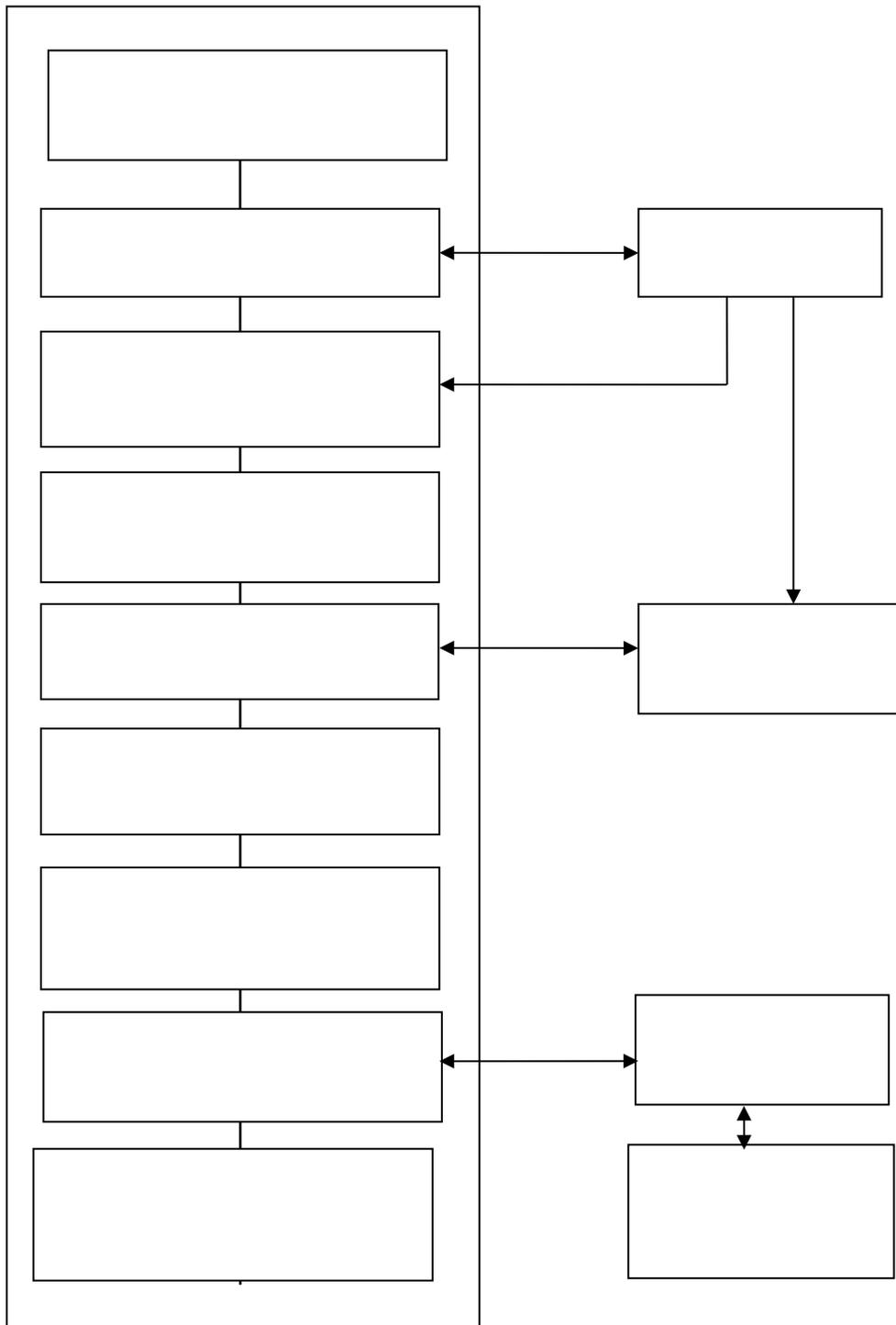


Рис. 2.23. Функции верхнего уровня управления.

2. Экстренный останов. При нажатии на пульте управления соответствующей кнопки система управления тормозит манипулятор с помощью двигателей. Торможение каждой степени подвижности осуществляется с максимальной интенсивностью, т.е. также как и при позиционном управлении. Питание приводов не отключается. Возможно продолжение исполнения рабочей программы. Недостатком экстренного

останова является то, что траектория торможения манипулятора может значительно отклоняться от программной.

3. Пауза в исполнении рабочей программы. После нажатия соответствующей кнопки на пульте ручного управления выполняется текущая команда. В результате робот останавливается. Торможение манипулятора осуществляется вдоль программной траектории. Для продолжения исполнения рабочей программы требуется нажатие соответствующей кнопки на пульте. Этот вариант останова полезен при пошаговой отладке.

Важной характеристикой экстренного торможения является выбег манипулятора до полного останова. Если манипулятор не оснащен тормозами путь аварийного торможения с отключением питания приводов больше, чем при экстренном торможении, ввиду того, что моменты, создаваемые приводами, в несколько раз больше трения в редукторах.

Например, манипулятор PRAGMA A-3000 останавливается за счет сил трения в редукторе менее, чем за 1 с. Разгон до максимальной скорости выполняется приводами за 0,5 сек. Из этого можно сделать вывод о том, что путь торможения приводами в два раза короче, чем за счет трения в редукторах. Для сравнения манипулятор PM-01 останавливается тормозами за 0.25 с. Соответственно отклонение от программной траектории у последнего кратно меньше.

Следует заметить, что движение манипулятора при штатной реализации аварийного и экстренного торможения является неуправляемым. Возникающее вследствие этого отклонение от программной траектории может привести к опасным столкновениям с объектами, находящимися в рабочей зоне.

Проблема отклонений при аварийном торможении известна давно. К настоящему времени предложено достаточно много способов ее решения. Условно их можно разделить на два направления.

Усилия части исследователей направлены на использовании в торможении помимо тормозов также двигателей манипулятора. Этот прием позволяет увеличить интенсивность торможения, что приводит к уменьшению выбега манипулятора и отклонений от программной траектории [4]. Результативность такого решения невелика, так как тормозные усилия, создаваемые тормозами существенно выше.

Вторая группа исследователей предлагает изменить конструкцию приводом с тем, чтобы не выключать двигатели, а использовать их для коррекции траектории торможения [5]. Недостатком этого способа является то, что он реализуем только, если все элементы системы управления исправны в том числе, например, датчиков положения степеней подвижности. В тоже время обесточивание приводов является необходимой мерой безопасности в ситуации, когда выявлена неисправность оборудования манипулятора.

Величину отклонения от программной траектории можно существенно уменьшить, если воспользоваться методом формирования траектории торможения манипулятора с помощью задержек отключения приводов [6].

В этом методе отключение питания привода каждой степени подвижности происходит не одновременно со всеми, а с заранее рассчитанной задержкой. Величины задержек вычисляются системой управления движением после поступления сигнала об аварийной остановке с таким расчетом, чтобы траектория торможения манипулятора оказалась близка к программной.

Техническая реализация предлагаемого способа управления траекторией торможения предполагает, что система управления может отключить привод любой степени подвижности независимо от других. Для этого достаточно оснастить систему управления робота разработанным устройством, которое описано в патентах РФ [7,8].

**Алгоритм вычисления задержек.** В момент поступления сигнала об аварийном торможении контроллер рассматриваемого устройства оценивает длительности интервалов времени  $T_i$ , в течение которых каждая степень подвижности остановится, используя формулу

$$T_i = \frac{v_i}{w_i},$$

где  $i$  – номер степени подвижности,  $v_i$  – скорость степени подвижности в начале торможения,  $w_i$  – ускорение торможения. В этой формуле скорость  $v_i$  является известной величиной. Она вычисляется контроллером по показаниям датчиков положения степеней подвижности.

После вычисления длительностей торможения всех степеней подвижности контроллер находит из них такую, которая тормозится дольше всех. Номер этой степени обозначим через  $i_0$ , длительность торможения – через  $T_{\max}$ .

Торможение остальных степеней подвижности осуществляется с некоторой задержкой, величина которой вычисляется с учетом условия минимизации отклонения расчетной траектории манипулятора от программной. Задача минимизации может быть решена численно с помощью известных методов вычислительной математики.

В работе [6] приведено решение этой задачи при допущении, что сила торможения при выключенном питании приводов постоянна:

$$T_{i \text{ торм}} = \frac{T_{\max} - \frac{v_i}{w_i}}{1 + \sqrt{2}}.$$

Здесь  $T_{i \text{ торм}}$  промежуток времени, через который должна быть выключена степень подвижности с номером  $i$ .

Аналитические оценки, которые получены в работе [6], показывают, что использование рассматриваемого способа формирования траектории торможения в большинстве случаев обеспечивает шестикратное уменьшение отклонение от программной траектории по сравнению с одновременным отключением приводов. При этом длительность торможения по сравнению с исходной остается прежней и равна  $T_{\max}$ , т.е. длительности торможения

степени подвижности  $i0$ . Любопытно, что степени подвижности прекращают движение в той же последовательности, в какой они останавливались бы при аварийном торможении с одновременным отключением питания.

**Средства программирования технологических операций.** Создание программ движений сложных технологических процессов с помощью языка управления роботом требует достаточно больших трудозатрат. Действительно, в многооперационной сборке редуктора [9], состоящего из 28 деталей, сборочный алгоритм состоял из около 1000 операторов языка, из которых более 100 были командами перемещений в целевую точку. Перепрограммирование целевых точек, к которому приводят непредвиденные столкновения с объектами, расположенными в рабочей зоне, при таком количестве программных точек занимают 1 рабочий день. Поэтому понятно желание разработчиков программных средств для манипуляционных роботов упростить работу технолога-программиста путем максимальной автоматизации процесса создания алгоритмов технологических операций, способов их верификации, отладки и настройки. Рассмотрим два пути, которые позволяют решить эти задачи.

Первый из них состоит в максимальном упрощении технологической задачи робота. Этот путь состоит в представлении технологического процесса последовательностью простых действий, каждое из которых выполняется одним манипулятором с помощью нескольких простых движений. При этом используются только транспортные перемещения манипулятора, выполняемые без контакта с технологическим оборудованием или в контакте, реализуемым простым движением. В настоящее время этот подход наиболее распространен среди разработчиков. Он реализован во всех современных роботизированных производственных линиях. В этом нетрудно убедиться по имеющимся в Интернете видеороликам.

Манипулятор заменяет автоматическую производственную ячейку. При этом отпадает необходимость проектирования, конструирования, изготовления и наладки специализированного оборудования для автоматизированного рабочего места. Программируемость траекторий движения манипулятора делает возможным удобное размещение обрабатываемых деталей и их ориентации перед и после обработки, упрощение конструкции подающего оборудования.

Конструкция деталей изменяется с целью упрощения выполнения технологической операции, ужесточаются зазоры и допуски деталей, оборудуются специальные места их хранения в ориентированном положении. Можно с уверенностью сказать, что страны, способные разработать роботизированные технологии массового выпуска продукции достигли нового технологического уровня производства. Список таких стран включает США, Германию и Японию.

Распространение рассматриваемой технологии ограничивает медленная окупаемость (до 10 лет), связанная с высокой стоимостью манипуляторов по отношению к высвобождаемому оборудованию и зарплатой человека, низкой

производительностью манипуляционного робота по сравнению с автоматическим оборудованием, необходимостью высококвалифицированного обслуживания манипулятора и роботизированного процесса.

Альтернативная технология предполагает использование адаптивных технологических движений и выполнение роботом на одной позиции сразу нескольких технологических операций. Реализуемость такого подхода в лабораторных условиях продемонстрирована достаточно давно (например, в работах [9]). Широкому его использованию в промышленности в настоящее время препятствует сложность разработки рабочей программы манипулятора, так как помимо сложности реализации отдельной технологической операции без существенной доработки поступающих деталей (т.е. как при ручном производственном процессе), надежность производственного процесса может быть обеспечена также контролем правильности всех технологических переходов. Следует, однако, иметь в виду, что отмеченные трудности вполне преодолимы по мере совершенствования программного обеспечения манипулятора и средств разработки технологических операций, которое происходит по мере накопления опыта их лабораторной и промышленной реализации.

Развитию рассматриваемого направления исследований мешает устоявшийся подход к разработке программного обеспечения манипуляторов, в котором нижний уровень управления обеспечивает исключительно позиционное или контурное перемещения схвата. Следствием такого подхода является то, что выполнение сложного адаптивного технологического движения возможно только путем оснащения манипулятора датчиками, например, силоизмерительными. Такое решение ведет к удорожанию робототехнического комплекса и усложнению задачи управления, так как приходится учитывать упругие свойства датчика и точность его измерений и, кроме того, позиционное и контурное управление движением манипулятора изначально не рассчитано на разработку мелких перемещений в рабочем диапазоне допустимой податливости датчика.

Альтернативой рассмотренному решению является разработка таких способов управления движениями манипулятора, которые насколько это возможно компенсируют особенности управления нижнего уровня. В качестве примера такого подхода рассмотрим реализацию движения схвата манипулятора ПРАГМА А-3000 (имеющего ортогональную кинематическую схему) по верхней полуокружности при переворачивании паллеты с зажатой в ней базовой деталью [10]. Паллета представляет собой прямоугольную конструкцию, к которой крепится корпус редуктора на время сборки. Она имеет ось, относительно которой паллета с зажатой в ней базовой деталью, переворачивается с целью продолжения сборки на противоположной стороне. Перед переворотом манипулятор оттягивает рукоятку, фиксирующую паллету, и перемещает ее по верхней полуокружности. Полуокружность имеет радиус 215 мм.

Манипулятор ПРАГМА А-3000 оснащен только позиционной следящей системой, поэтому реализация достаточно точного движения по

полуокружности представляет для него неочевидную задачу. Ее успешному решению способствовала возможность задавать отклонение от целевой точки, при котором начинается движение в следующую программную точку [10].

Эксперименты показали, что программирование траектории в виде последовательности промежуточных точек приводит к достаточно большим отклонениям от полуокружности, так как между целевыми точками схват движется по прямой. Если задать достаточно большое количество промежуточных точек с тем, чтобы уменьшить до приемлемой величины (10 мм) отклонение от полуокружности, то скорость переворота оказывается слишком большой (более 10 с).

В работе [11] реализован следующий способ задания траектории (рис. 2.24). На рисунке полуокружность отмечена пунктиром. Точка  $O$  – ее центр полуокружности.  $A$  и  $B$  – начальная и конечная точки движения.  $C$  и  $K$  – вершины квадрата, в который вписана полуокружность. Цифрами 1 и 2 обозначены направления перемещения степеней подвижности манипулятора. Каждой из них может быть задано перемещение в целевую точку независимо от другой.

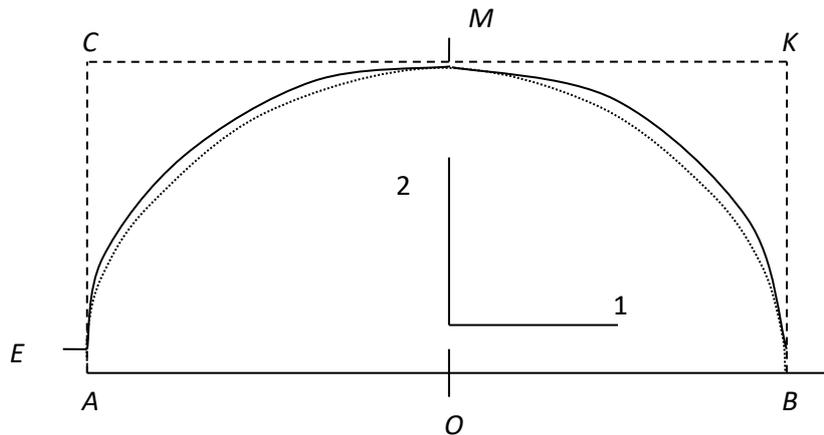


Рис. 2.24. Формирование движения по верхней полуокружности при позиционном управлении. Пунктиром показана верхняя полуокружность, сплошной линией – фактическая траектория движения схвата.

Ось вращения паллеты расположена перпендикулярно плоскости первой и второй степеней подвижности, т.е. переворот осуществляется в плоскости указанных степеней подвижности.  $AB$  лежит вдоль линии перемещения первой степени. Эти обстоятельства существенно упростили разработку алгоритма выполнения операции.

Сначала выдается команда переместить в точку  $C$  вторую степень подвижности. По достижении ею уровня точки  $E$  запускается движение по направлению к точке  $B$  первой степени подвижности со скоростью, величина которой подбирается при настройке алгоритма. Как только первая степень достигнет положения с горизонтальной координатой точки  $M$ , (ее величина

лежит в окрестности точки  $O$ ), выдается команда переместить вторую степень подвижности в направлении точки  $B$ .

Кривая, которая получается в результате работы алгоритма показана сплошной линией. Ее отклонение от полуокружности определяется тремя величинами: вертикальной координатой точки  $E$ , горизонтальной координатой точки  $M$  и скоростью перемещения горизонтальной степенью подвижности  $V$ .

Алгоритм переворота настраивается путем подбора перечисленных величин и многократного опробования получающейся траектории. Процесс настройки упрощает достаточно высокая повторяемость движения манипулятора и то обстоятельство, что вертикальная координата точки  $E$  определяет отклонение в первой четверти окружности, горизонтальная координата точки  $M$  и скорость  $V$  – отклонение во второй четверти.

Эксперименты показали, что при тщательном выборе параметров траектория движения манипулятора ПРАГМА А-3000 отклоняется от полуокружности не более, чем на 10 мм. Аналогичный алгоритм использовался для управления переворотом паллеты манипулятором Универсал-5 (рис. 2.25) [12], причем этот манипулятор мог управляться в режиме позиционного, так и в режиме контурного управления. Это сделало возможным проведение эксперимента по сравнению точности и скорости исполнения движения в разных режимах. При контурном управлении отклонение не превышало 2 мм, т.е. было в несколько раз меньше, чем при позиционном. Длительность переворота при контурном управлении была на 25 % больше – 5 с против 4 с. Ввиду того, что конструкция схвата допускала отклонение рукоятки от полуокружности на 20 мм, в сборочном процессе и использовался режим позиционного управления.

Приведенный пример формирования траектории при позиционном управлении показывает, что творческое использование возможностей верхнего уровня управления позволяет расширить множество движений, которые могут быть реализованы манипуляционным роботом. К сожалению, успех применение найденных возможностей зависит от конструкции конкретной модели манипулятора и его системы управления, т.е. они не являются универсальным способом преодоления ограничений, которые вносит реализация управления нижнего уровня. Например, описанный выше алгоритм не удалось применить для робота РМ-01, так как его система управления не допускает возможности переназначения пользователем целевой точки в произвольный момент времени и делает это только автоматически.



*Рис. 2.25. Манипулятор УНИВЕРСАЛ-5*

Это позволяет сделать вывод о целесообразности введения в систему управления нижнего уровня параметров, определяющих особенности движения в целевую точку, которые доступны разработчику.

### **Литература**

1. Шевченко Ю.Л., Карпов О.Э., Ветшев П.С., Степанюк И.В. Вестник Национального медико-хирургического центра им. Н.И. Пирогова 2 т.3, 2008 года.
2. [http://fano.gov.ru/ru/press-center/card/?id\\_4=38034](http://fano.gov.ru/ru/press-center/card/?id_4=38034).
3. В.А. Карташев, В.В. Сафронов Система автоматического контроля безопасности перемещений манипуляционного робота. Известия РАН. Теория и системы управления. № 4 2017.
4. J. Hofmann, R. Schwarz, S. Marx. Industrial Robot Having Redundant Emergency Brake Circuit. Pat. US 8736219 B2. 2014.
5. S.J. Ho, K. S. Rak. Emergency stop method of industrial robot. Pat. KR20120045766 (A), 2012-05-09.
6. В.А. Карташев, В.В. Карташев. Способ формирования траектории аварийного торможения манипулятора. Известия РАН. Теория и системы управления. №. 4. 2018.
7. В.А. Карташев, В.В. Карташев. Способ управления аварийным торможением манипулятора. Патент РФ № 2657266. 2018.
8. В.А. Карташев, В.В. Карташев. Устройство управления аварийным торможением манипулятора. Патент РФ № 2658589. 2018.

9. Д.Е.Охоцимский, С.С.Камынин, В.А. Карташев, Е.И.Кугушев. Автоматическая многооперационная сборка с помощью промышленных роботов. Роботизация сборочных процессов М., Наука, 1985.
10. Д.Е.Охоцимский, С.С.Камынин, В.А. Карташев, Е.И.Кугушев. Реализация сборочных операций на работе ПРАГМА А-3000. Сборник трудов ИПМ РАН. № 142. 1985.
11. В.А. Карташев. Оптимизация транспортных перемещений сборочного робота. ТЕХНОЛОГИЯ. Гибкие производственные системы, Вып.1, М.,ВНИИМИ,1993.
12. В.А. Карташев. Программное обеспечение робототехнического сборочного комплекса. Программное обеспечение роботов. М., Наука, 1991.
13. Шевченко Ю.Л., Карпов О.Э., Ветшев П.С., Степанюк И.В. Вестник Национального медико-хирургического центра им. Н.И. Пирогова 2 т.3, 2008 года.
14. [http://fano.gov.ru/ru/press-center/card/?id\\_4=38034](http://fano.gov.ru/ru/press-center/card/?id_4=38034).
15. В.А. Карташев, В.В. Сафронов Система автоматического контроля безопасности перемещений манипуляционного робота. Известия РАН. Теория и системы управления. № 4 2017.
16. J. Hofmann, R. Schwarz, S. Marx. Industrial Robot Having Redundant Emergency Brake Circuit. Pat. US 8736219 B2. 2014.
17. S.J. Но, К. S. Rak. Emergency stop method of industrial robot. Pat. KR20120045766 (A), 2012-05-09.
18. В.А. Карташев, В.В. Карташев. Способ формирования траектории аварийного торможения манипулятора. Известия РАН. Теория и системы управления. №. 4. 2018.
19. В.А. Карташев, В.В. Карташев. Способ управления аварийным торможением манипулятора. Патент РФ № 2657266. 2018.
20. В.А. Карташев, В.В. Карташев. Устройство управления аварийным торможением манипулятора. Патент РФ № 2658589. 2018.
21. Д.Е.Охоцимский, С.С.Камынин, В.А. Карташев, Е.И.Кугушев. Автоматическая многооперационная сборка с помощью промышленных роботов. Роботизация сборочных процессов М., Наука, 1985.
22. Д.Е.Охоцимский, С.С.Камынин, В.А. Карташев, Е.И.Кугушев. Реализация сборочных операций на работе ПРАГМА А-3000. Сборник трудов ИПМ РАН. № 142. 1985.
23. В.А. Карташев. Оптимизация транспортных перемещений сборочного робота. ТЕХНОЛОГИЯ. Гибкие производственные системы, Вып.1, М., ВНИИМИ, 1993.
24. В.А. Карташев. Программное обеспечение робототехнического сборочного комплекса. Программное обеспечение роботов. М., Наука, 1991.

### 3. БИОЛОГИЧЕСКИ ИНСПИРИРОВАННЫЕ НЕЙРО-НАВИГАЦИОННЫЕ СИСТЕМЫ ДЛЯ МОБИЛЬНЫХ ШАГАЮЩИХ РОБОТОВ

Решение навигационной задачи является важным этапом построения управления подвижным роботом. Рассмотрим решение задач локализации для шестиногого и андроидного (двуногого) роботов. Для шестиногого робота решается задача, аналогичная задаче, решаемой по показаниям одометрических датчиков для колесных аппаратов. определение положения робота в окружающем пространстве по измерениям углов в шарнирах ног. Для андроидного робота рассмотрены две задачи. Первая состоит в аналогичном определении положения двуногого робота в окружающем пространстве по измерениям углов в шарнирах ног. Вторая – определение позы (положения) головы робота относительно корпуса по измерениям ускорений в двух характерных точках головы, эта задача моделирует вестибулярный аппарат человека. Все задачи решаются в нейросетевом базисе [1-4], в предположении, что роботу известна местная вертикаль.

Применение нейросетей для решения поставленных задач вызвано следующим обстоятельством. В рассматриваемых задачах часть определяемых навигационных параметров не может быть измерена непосредственно, а их прямое вычисление затруднительно, если измерения проведены с ошибками. Затруднения возникают из-за несовместности соотношений, связывающих определяемые параметры с измерениями при наличии ошибок. Нейросети предлагают эффективное решение этой проблемы. Кроме этого, нейросетевое решение задач ориентации робота в пространстве имеет биологический прототип в биологических живых системах, что делает применение нейросетей актуальным.

Отметим, что предлагаемое применение нейросетей является примером использования средств Искусственного Интеллекта (ИИ) в задачах робототехнической навигации.

#### 3.1. Нейросети. Краткое описание

Искусственная нейронная сеть (ИНС) – это абстрактная математическая модель, представляющая собой сеть (структурное объединение) искусственных нейронов. В естественном биологическом нейроне обычно выделяют элементы, схематично показанные на рис. 3.1 (слева). На том же рисунке справа условно показан фрагмент нейросети.

Элементы нейрона таковы: *входы нейрона*, или *дендриты*, к ним «подключены» *аксоны* (выходы) других нейронов, это подключение обычно выполнено через специальные сложные «устройства», называемые *синапсами* (их сложность связана со сложным характером передачи информации через синапс), *тело нейрона*, в котором преобразуется полученная информация (взвешенное суммирование сигналов от входов и преобразование полученной суммы специальной функцией активации), *выход нейрона*, называемый

аксоном. В общей нервной системе множество нейронов объединены в сеть согласно указанной схеме дендриты-аксоны, такая сеть условно показана на рис. 3.1. справа. Информация в биологической нейронной сети передается электрическими импульсами, причем процесс передачи является электрохимическим. Считается по современным данным, что в мозге человека до 86 млрд. нейронов, причем они не все идентичны, а специфицированы по меньшей мере по функциям.

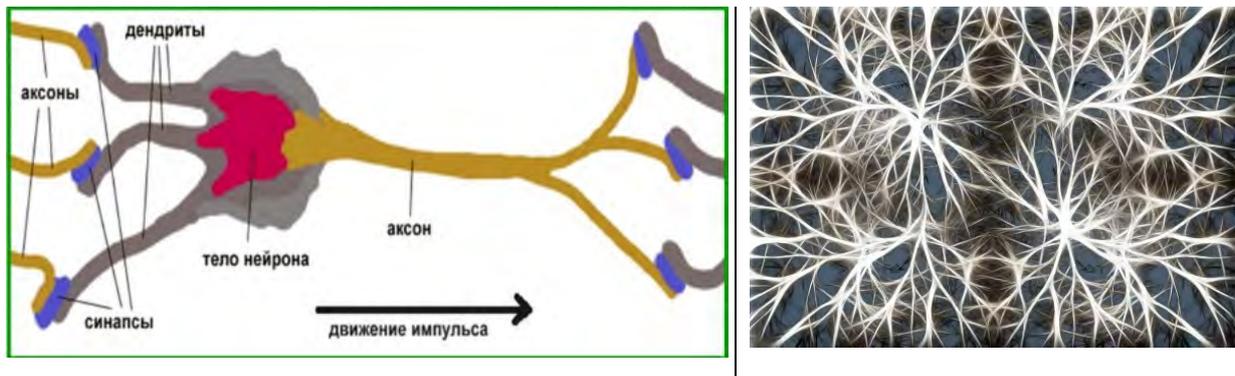


Рис. 3.1. Схема биологического нейрона (слева), условная схема биологической нейросети (справа) (по материалам ресурсов [1-3]).

Ниже на рис. 3.2 приведена схема одного из базовых искусственных нейронов, которая также является абстрактной математической моделью. Это так называемый нейрон МакКаллока-Питтса [1]. Показанная на рис. 3.2 модель искусственного или формального нейрона в настоящее время широко используется в ИНС.

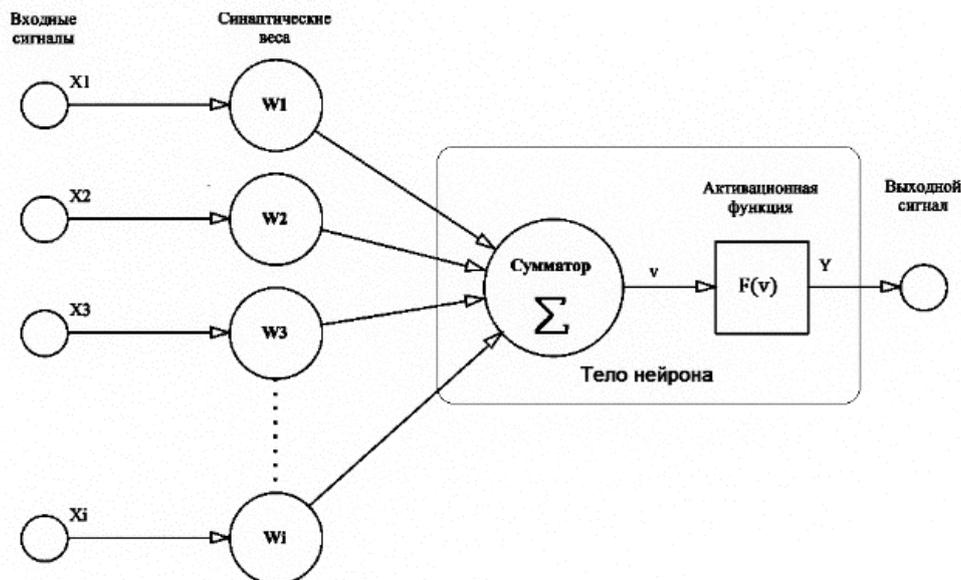


Рис. 3.2. Схема искусственного нейрона

Нейрон на рис. 3.2 имеет  $j$  входов (дендритов) от входных сигналов  $x_1, x_2, \dots, x_j$ , образующих  $j$  связей «вход-тело нейрона», каждой такой связи  $k$  приписан весовой коэффициент (синаптический вес)  $W_k, k=1, 2, \dots, j$ , коэффициенты  $W_k$  поначалу при построении сети неизвестны и подлежат определению. Задача определения  $W_k$  называется тренировкой или обучением сети (когда нейроны объединены в сеть) и решается различными алгоритмами, среди них популярен так называемый алгоритм обратного распространения ошибки [4, 5].

Нейрон имеет тело, в котором имеются сумматор и преобразователь. Соответственно, в теле нейрона выполняются две математические операции

- 1) взвешенное суммирование входов с весами ( $f_b$  – смещение, часто полагают  $f_b = 0$ )

$$v = (\sum W_1 \times x_1 + W_2 \times x_2 + \dots + W_j \times x_j) + f_b ,$$

- 2) нелинейное преобразование полученной суммы с помощью функции  $F$ , называемой функцией активации нейрона (активационной функцией),  $Y$  – выход нейрона

$$Y = F(v)$$

В качестве функции активации обычно используют одну из следующих четырех:

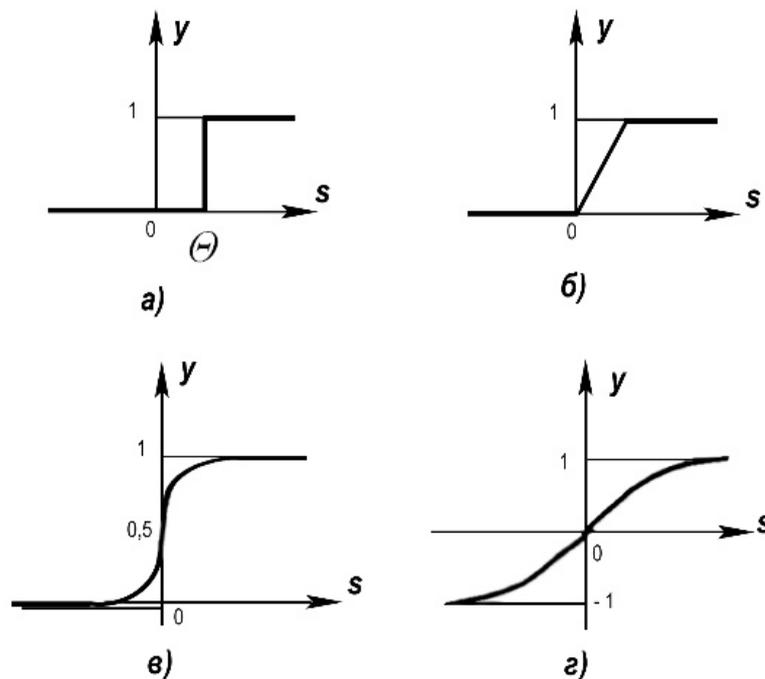


Рис. 3.3. Функции активации формального нейрона

Эти функции таковы [4, 5]: а) функция единичного скачка, б) гистерезис, в) сигмоид (логистическая функция), г) сигмоид (гиперболический тангенс). Существуют и другие функции активации, но перечисленные четыре используются чаще. Тот или иной тип функции активации

Далее, построение ИНС для конкретной задачи из описанных нейронов заключается в выборе архитектуры сети (определении типа связей нейронов), выборе типа функции активации и обучении построенной сети, т.е. вычислении весовых коэффициентов, приписанных связям в сети. Архитектур ИНС существует уже достаточно много [5, 6], в том числе существенно сложных и изощренных. Среди них одной из простых является многослойная ИНС, пример которой показан на рис. 3.4 – так называемый многослойный перцептрон (Multi Layered Perceptron MLP).

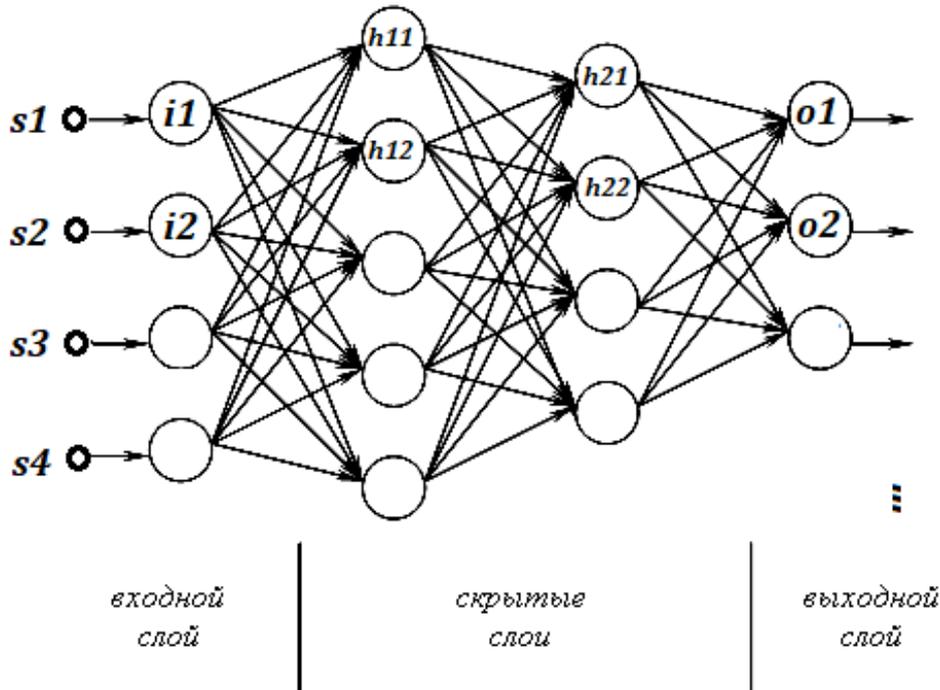


Рис. 3.4. ИНС многослойный перцептрон (MLP)

В MLP все нейроны одинаковы, они организованы в несколько слоев: входной слой (нейроны  $i1, i2, \dots$ ), один или несколько скрытых слоев (нейроны  $h11, h12, \dots, h21, h22, \dots$ ) и выходной слой (нейроны  $o1, o2, \dots$ ). К нейронам входного слоя подключены датчики и источники входных сигналов  $s1, s2, \dots$ , естественно, число входных нейронов равно числу датчиков и источников информации, вводимой в сеть. Выходные нейроны формируют выходные сигналы сети, их число равно числу требуемых выходов. Число нейронов скрытых слоев обычно подбирают экспериментально, их число выбирают больше числа входных и выходных нейронов. Топология связей в перцептроне такова: от каждого нейрона каждого слоя связи ведутся «вперед» по сети по одной к каждому нейрону следующего слоя. Связи обратного направления отсутствуют. В результате число выходных аксонов каждого нейрона обычно получается больше одного (или он разветвляется) и в этом заключается расширение формальной модели, показанной на рис. 3.3.

Именно MLP в основном используются для решения названных задач настоящей работы.

### 3.2. Шестиногий робот: локализация

Рассмотрим задачу локализации для шестиногого аппарата. Цель такой задачи – определение положения робота в пространстве. Измерения (входные сигналы) доставляются датчиками углов в шарнирах ног аппарата. Также известными считаются абсолютные координаты опорных точек опорных ног.

#### 3.2.1. Описание модели

Будем предполагать, что шагающий аппарат состоит из абсолютно твердого корпуса-параллелепипеда и шести ног. С корпусом жестко свяжем ортогональную декартовую систему координат  $Oxyz$ . Пусть  $O_1\xi\eta\zeta$  неподвижная абсолютная система координат;  $O\xi\eta\zeta$  система координат, движущаяся поступательно вместе с точкой  $O$ , причем ее оси в каждый момент времени параллельны соответствующим абсолютным осям (система осей Кенига). Шагающий аппарат рассматривается как шарнирно соединенная совокупность абсолютно твердых тел, корпуса и звеньев ног. Считается что двигатели, стоящие в шарнирах, реализуют заданные управляющие программы (моменты). Модель шагающего аппарата приведена на рис. 3.5.

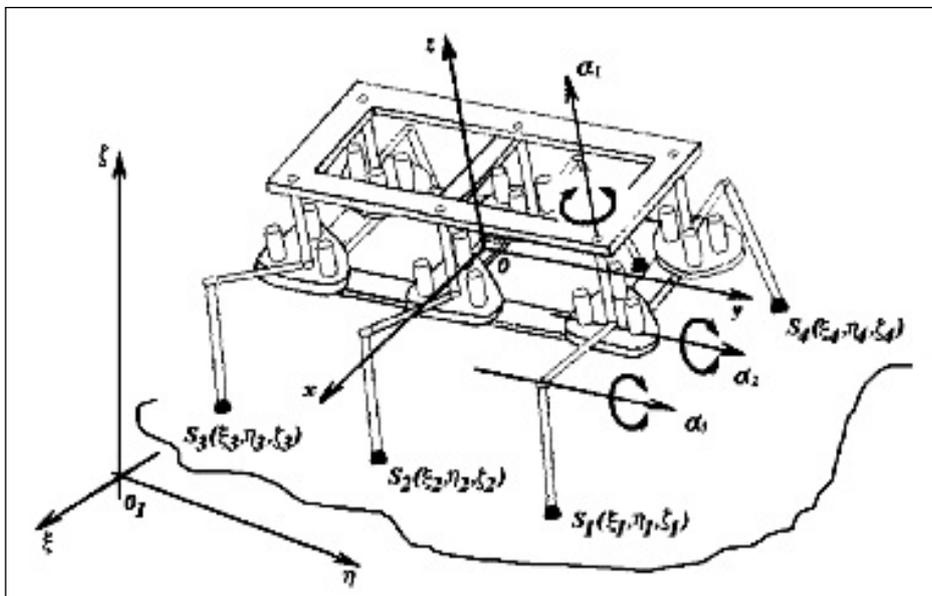


Рис. 3.5. Модель шагающего аппарата

Все ноги аппарата предполагаются одинаковыми. Отдельная нога состоит из двух звеньев – бедра и голени. Бедро соединено с корпусом посредством двухстепенного шарнира. Связь между бедром и голенью осуществляется одностепенным шарниром. Примем, что нога может иметь с несущей поверхностью лишь одну точку контакта (стопу). Нога называется

опорной, если она касается поверхности и возникающая в точке касания реакция опоры отлична от нуля. Полагаем, что в системе координат  $Oxyz$ , жестко связанной с корпусом шагающего аппарата, оси координат направлены по главным осям симметрии корпуса: ось  $Oy$  направлена по направлению движения шагающего аппарата, ось  $Oz$  – «вверх» перпендикулярно плоскости корпуса, а ось  $Ox$  так, чтобы образовывалась правая тройка векторов.

Положение корпуса будем описывать шестью координатами – тремя абсолютными координатами  $(\xi, \eta, \varsigma)$  точки  $O$  и тремя углами  $\psi, \vartheta, \gamma$ . Выбранные углы определяют ориентацию осей системы  $Oxyz$  относительно неподвижных осей  $O_1\xi\eta\varsigma$ :  $\psi$  – угол между проекцией оси  $Oy$  на плоскость  $O_1\xi\eta$  и осью  $O_1\eta$  (угол рыскания),  $\vartheta$  – угол между осью  $Oy$  и плоскостью  $O_1\xi\eta$  (угол тангажа),  $\gamma$  – угол поворота корпуса вокруг оси  $Oy$  против часовой стрелки относительно направления оси (угол крена).

Схема модели ноги робота приведена на рис. 3.6.

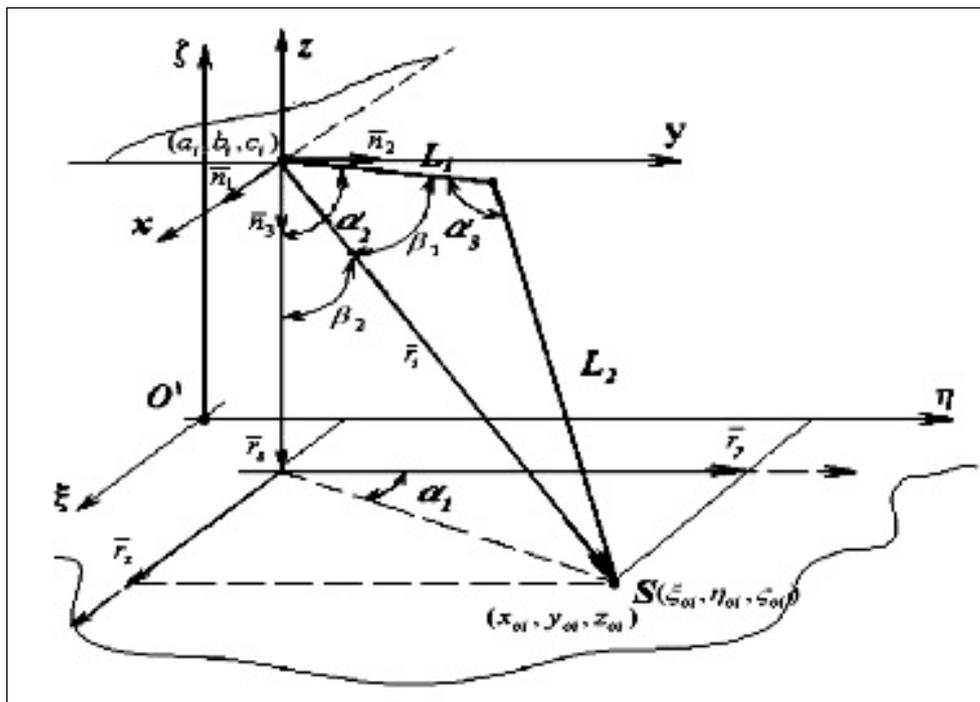


Рис. 3.6. Модель ноги шагающего аппарата.

Все динамические и кинематические характеристики ног предполагаются известными. Считается, что положение ног относительно корпуса может быть полностью определено в каждый текущий момент времени с использованием измерений, проводимых в процессе движения. Если нога с номером  $i$  находится в состоянии опоры, то радиус-вектор ее точки контакта с поверхностью  $nf_i$  в системе координат  $Oxyz$  и скорость этой точки относительно корпуса  $vf_i$  являются известными величинами.

### 3.2.2. Методика расчетов

Воспользуемся нейросетевым методом. В этой схеме данные с датчиков углов в шарнирах ног робота и рассчитанные на предыдущих циклах координаты опорных точек поступают на входы нейронной сети, выходом сети являются текущие значения координат и навигационных углов, определяющих подвижную систему в системе  $O_1 \xi \eta \zeta$ .

Нейросеть, решающая навигационную задачу, должна решать систему нелинейных уравнений:

$$(\xi, \eta, \zeta, \varphi, \psi, \gamma) = F(\alpha_{i,j}, \vec{a}) \quad (3.1)$$

где  $\alpha_{i,j}$  – измеренные углы в шарнирах ног, возможно, неточные,  $(\xi, \eta, \zeta)$  – координаты центральной точки корпуса робота в системе  $O_1 \xi \eta \zeta$ ,  $\psi, \varphi, \gamma$  – навигационные углы корпуса робота,  $\vec{a}$  – вектор координат точек опоры,  $F$  – нелинейное отображение измеряемых (определяющих) параметров в определяемые.

Для решения поставленной задачи была выбрана искусственная нейронная сеть прямого распространения с одним скрытым слоем.

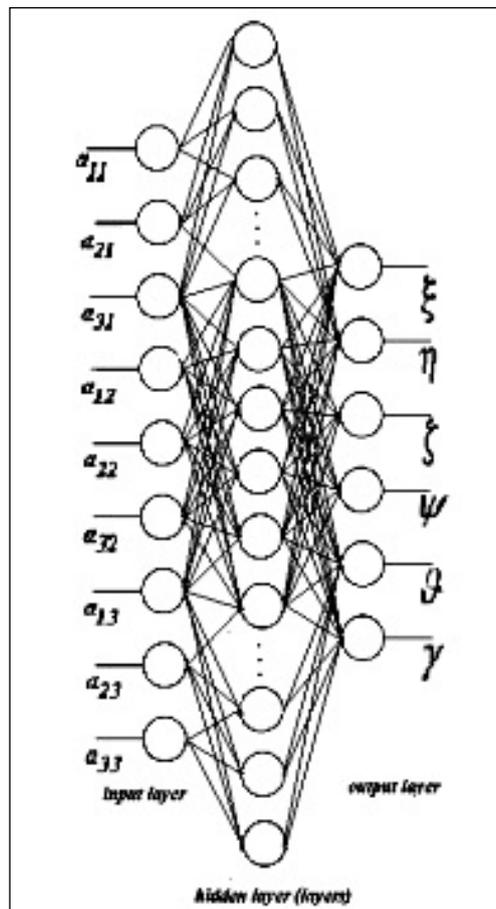


Рис. 3.7. Навигационная нейросеть для походки «трешками».

На рис. 3.7 показана такая нейросеть для одного из вариантов расчетов. Показана сеть для варианта походки «трешками», это статически устойчивая походка, когда аппарат в каждый момент времени опирается на три ноги, или на все шесть ног. При опоре на три ноги (две крайние с одной стороны и средняя с другой стороны) обеспечивается минимальный устойчивый вариант движения, и именно соответствующая этому сеть показана на рис. 3.7. Поэтому во входном слое сети 9 переменных, соответствующих девяти измеряемым углам опорных ног, а на выходе сети 6 навигационных параметров.

Для корректной работы НС ее необходимо предварительно настроить. Для этого используется метод обучения сети «обучение с учителем» и строится специальная обучающая выборка, состоящая из некоторого набора входов и соответствующих им (в смысле соотношения (3.1)) выходов. Так как опорные точки неподвижны и длины звеньев ног фиксированы, положение корпуса не может быть произвольным – расстояния от точек подвеса ног до опорных точек не могут превосходить длины ног. Для построения обучающего множества использовалось обучающее пространство, соответствующее этому условию при различных значениях параметров  $a$ . Навигационные углы для обучения брались в пределах от  $-\pi/4$  до  $\pi/4$ .

Обучающая выборка – множество обучающих точек. Точки обучающего множества выбирались двумя способами – регулярным и случайным образом. В последнем случае обеспечивается лучшее качество работы и скорость обучения сети по сравнению с равномерной расчетной сеткой. Однако, построение регулярного набора проще. Пример частичного набора для регулярного случая показан на рис. 3.8.

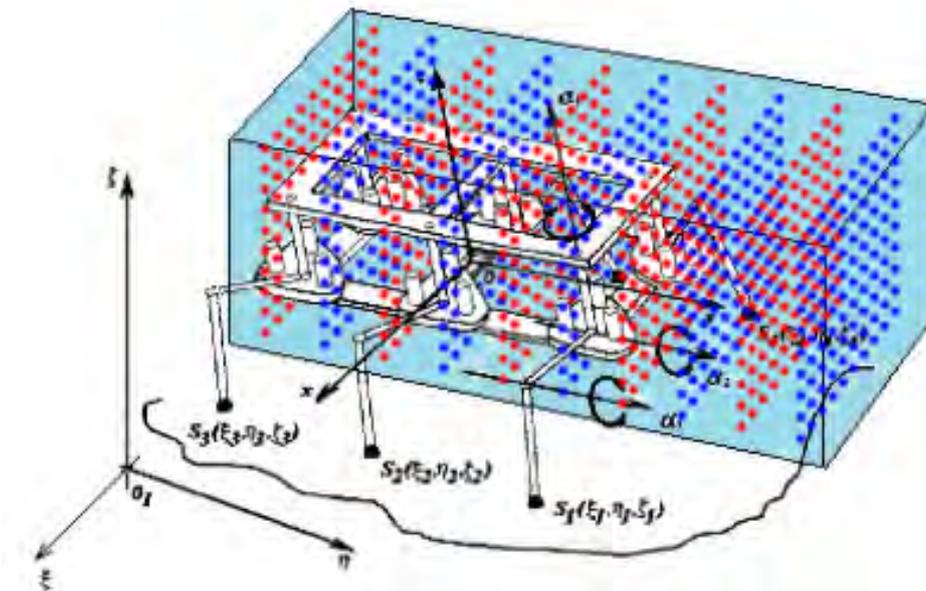


Рис. 3.8. Пример регулярного обучающего подмножества.

На рис. 3.8 показано множество для обучения сети по трем линейным координатам, такой же параллелепипед строится в пространстве угловых координат. В пространстве построенного параллелепипеда показано подмножество регулярных обучающих точек, оно равномерно заполняет весь параллелепипед. Шаг построения точек по параллелепипеду можно изменять.

### 3.2.3. Определение архитектуры нейросети

Для определения конкретной структуры сети был проведен ряд численных экспериментов. Рассматривались сети, содержащие различное число нейронов на скрытом слое в пределах от 10 до 40. Для каждой сети проводилась настройка на обучающих множествах разной мощности от 5000 точек до 20000 точек. По результатам тестов оценивались 2 критерия качества: точность и обобщающая способность полученной сети.

Использовались тестовые множества трех видов:

1. Входные данные содержат малые ошибки (до 0.3 градуса)
2. Входные данные содержат большие ошибки (от 0.3 до 1 град.)
3. Входные данные сильно отличаются от обучающих.

Помимо чувствительности модели к шуму интересна также ее чувствительность к изменениям величин координат опор ног  $a$ . Оказалось, что вдали от границ обучающего пространства точность зависит от  $a$  слабо.

### 3.2.4. Результаты моделирования

При моделировании прежде всего рассматривались базовые движения отдельно по каждой координате. Вектор состояния робота определялся следующим образом:

- 1 тип движения – движение по координате  $\xi$ ,  $\eta, \zeta = const$ ,  $\psi, \vartheta, \gamma = 0$ .
- 2 тип движения – движение по координате  $\eta$ ,  $\xi, \zeta = const$ ,  $\psi, \vartheta, \gamma = 0$ .
- 3 тип движения – движение по координате  $\zeta$ ,  $\xi, \eta = const$ ,  $\psi, \vartheta, \gamma = 0$ .
- 4 тип движения – движение по координате  $\psi$ ,  $\xi, \eta, \zeta = const$ ,  $\vartheta, \gamma = 0$ .
- 5 тип движения – движение по координате  $\vartheta$ ,  $\xi, \eta, \zeta = const$ ,  $\psi, \gamma = 0$ .
- 6 тип движения – движение по координате  $\gamma$ ,  $\xi, \eta, \zeta = const$ ,  $\psi, \vartheta = 0$ .

При формировании измерений углов в шарнирах ног шагающего аппарата вводились соответствующие случайные последовательности (шумы в измерениях), распределенные по нормальному закону с математическим ожиданием равным нулю. Среднеквадратические значения шума в измерениях принимались разными и были приняты соответственно равными  $\frac{\pi}{3000}$ ;  $\frac{\pi}{600}$ ;  $\frac{\pi}{300}$ ;  $\frac{\pi}{60}$ ;  $\frac{\pi}{30}$ , что соответствует относительным погрешностям в измерениях углов в 0.1%; 0.5%; 1%; 5%; 10%.

В моделировании рассматривались и сравнивались различные навигационные фильтры, нейросетевой (НСМ), и построенный по схеме метода наименьших квадратов (МНК), в строго одинаковых условиях, при разном значении относительных погрешностей в измерениях углов.

Информация для расчета строилась по кинематическим схемам для походок класса «трешки». Удовлетворительные результаты в НСМ получаются для нейронной сети следующей конфигурации: нейронная сеть состоит из одного входного слоя с девятью нейронами на входе, одного выходного слоя с шестью нейронами, и одного внутреннего слоя с 36 нейронами. Такую нейросеть оказалось возможным обучить с погрешностью 1% (время обучения оказалось весьма значительным).

Оказалось, далее, что и метод наименьших квадратов и нейросетевой метод оба обрабатывают ошибки достаточно хорошо. Однако, необходимо отметить, что МНК является лучшим в сравнении с любым другим рассмотренным методом решения поставленной задачи, поскольку он оптимален в смысле точности даже при больших отклонениях в значениях измеряемых величин. Нейросетевой метод имеет близкие показатели. Но помимо этого НСМ устойчив к увеличению ошибки измерений, и требует меньше времени для получения определенной точности вычислений, что является его положительной характеристикой.

Затем в качестве тестового движения было выбрано прямолинейное движение с поворотом по углу тангажа  $\vartheta$  на  $15^\circ$  по углу крена  $\gamma$  на  $10^\circ$ . Изучалась позиционная ошибка НСМ и МНК методов. МНК оказался более точным, но более медленным в вычислительном плане методом. Моделирование тем самым показывает, что возможно найти нейронный алгоритм для эффективного вычисления положения шагающего аппарата и его ориентации, особенно если область обучения сокращается до одной плоскости, как это бывает, когда описывается постоянное регулярное движение шагающей машины.

Полученные результаты планируется применить на шестиногом роботе (рис. 3.9), создаваемом в ИПМ им. М.В. Келдыша РАН.



Рис. 3.9. Шестиногий шагающий робот.

Сказанное позволяет сделать общий вывод о достаточной точности нейросетевой локализационной системы и эффективном решении навигационной задачи в нейросетевом базисе.

### 3.3. Метод локализации двуногого аппарата в пространстве

Решается локальная задача проприоцептивной (по датчикам внутренней информации) навигации для двуногого шагающего аппарата: определение геометрии позы и ориентации в окружающем пространстве. Задача рассматривается в следующих предположениях:

1. Робот находится в фазе стояния, его опорные точки находятся в известных положениях и неподвижны.
2. Координаты опорных точек известны.
3. Прямым измерениям доступны углы в сочленениях ног и корпуса. Измерения проводятся с некоторыми ошибками.

Для решения задачи будем применять аппарат искусственных нейронных сетей [5, 6].

#### 3.3.1. Постановка задачи

Построим решение задач локализации и ориентации в пространстве в сформулированных выше предположениях. Оценим точность метода и его чувствительность к изменению параметров задачи.

#### 3.3.2. Описание модели

Опишем модель формально. На рис. 3.10 приведена ее геометрическая схема.

Пусть  $A, B$  – опорные точки ног робота. Точки  $R$  и  $L$  будем условно называть «коленями», а точку  $P$  – «тазом» робота,  $AR = BL = l_1$  и  $RP = LP = l_2$  – длины звеньев ног робота,  $l = l_1 + l_2$  – общая длина ноги. В расчетной модели  $l_1 = 60$  см и  $l_2 = 40$  см,  $l = 100$  см.

Введем абсолютную систему координат  $OXYZ$ , где точка  $O$  – середина отрезка, соединяющего опорные точки  $A$  и  $B$ . Ось  $OZ$  направлена вертикально вверх, ось  $OY$  сонаправлена с вектором  $AB$  (рис. 3.10).

Относительную систему координат  $P\xi\eta\zeta$  свяжем с корпусом робота. Ось  $P\xi$  направлена вдоль корпуса, как это показано на рис. 3.10. Примем, что корпус робота не может поворачиваться вокруг оси  $P\xi$  и, таким образом, имеет две степени свободы. Тогда ось  $P\xi$  расположена в плоскости, параллельной  $OXZ$ , а ось  $P\eta$  образует с ними правую тройку.

Координаты опорных точек  $A, B$  в системе координат  $OXYZ$  известны и равны соответственно  $(0, -a, 0)$  и  $(0, a, 0)$ . Будем также считать, что шарниры в «коленях» имеют одну степень свободы, т.е. каждая «нога» лежит в одной плоскости с точкой  $P_l$  – ортогональной проекцией «таза» на плоскость  $OXY$ .

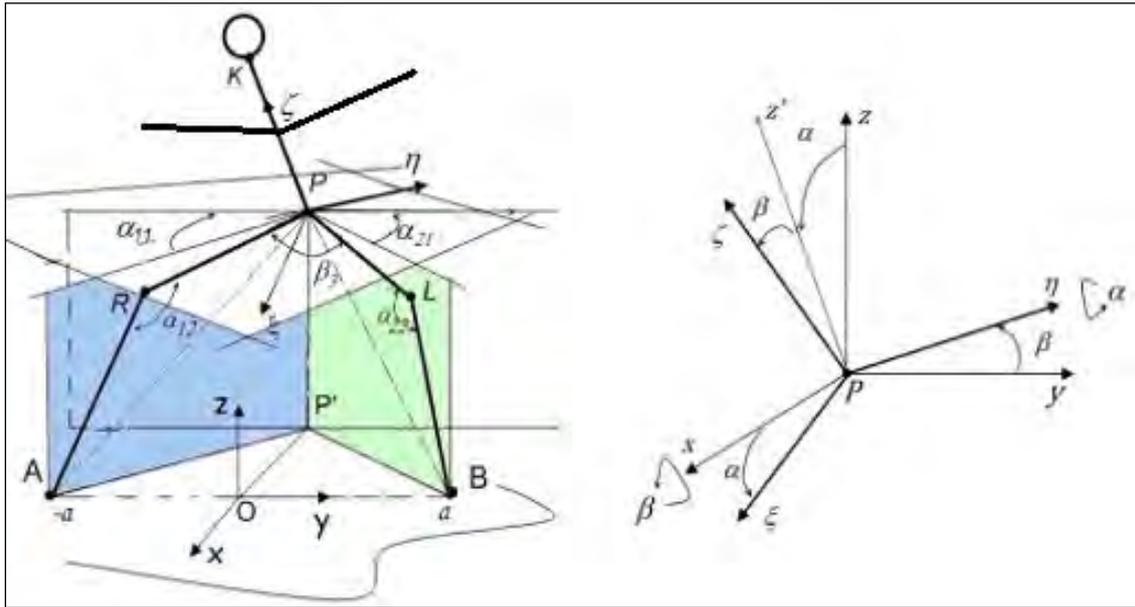


Рис. 3.10. Модель аппарата и параметры геометрии корпуса.

Опишем угловые параметры. Углы  $\alpha_{11}$  и  $\alpha_{21}$  характеризуют углы поворота плоскостей каждой ноги в абсолютной системе координат (так как вертикаль известна, их измерение не составляет труда),  $\alpha_{12}$  и  $\alpha_{22}$  углы в сочленениях ног (коленках). Углы  $\theta_1, \theta_2, \theta_3$  – углы между  $Pz$  и  $PR$ ,  $Pz$  и  $PL$ ,  $PR$  и  $PL$  соответственно. В качестве навигационных углов рассматриваются углы Крылова: крен, тангаж и рыскание (см. рис. 3.10 справа). Угол тангажа соответствует наклону корпуса вперед и назад (углу  $\alpha$  поворота вокруг оси  $PY$ ), угол крена – наклону корпуса вправо и влево (углу  $\beta$  поворота вокруг оси  $Pz$ ), угол рыскания (угол поворота корпуса вокруг оси  $Pz$ ) в рассматриваемом случае равен нулю, так как вращения вокруг оси  $Pz$  нет.

Будем считать, что, исходя из углов  $\alpha_{ij}, \theta_i$  и координат точек  $A$  и  $B$  необходимо определить набор  $(x, y, z, \alpha, \beta)$ , в который входят координаты  $(x, y, z)$  точки  $P$  и навигационные углы  $\alpha, \beta$ , определяющие ориентацию относительной системы отсчета.

### 3.3.3. Методика расчетов

Возможным методом определения навигационных параметров для шагающего аппарата может служить нейросетевой метод. В этой схеме, как и в предыдущей задаче, данные с датчиков углов в шарнирах робота и координаты опорных точек поступают на входы нейронной сети, выходом сети являются текущие значения координат и навигационных углов, определяющих систему  $Pz\eta z$  в системе  $OXYZ$ . Нейросеть, решающая навигационную задачу, должна решать систему нелинейных уравнений

$$(x, y, z, \alpha, \beta) = F(\alpha_{i,j}, \theta_i, a) \quad (3.2)$$

где  $\alpha_{ij}$ ,  $\theta_i$  – измеренные углы в шарнирах, вообще, неточные,  $x$ ,  $y$ ,  $z$  – координаты точки  $P$  корпуса робота в системе  $OXYZ$ ,  $\alpha$ ,  $\beta$  – навигационные углы корпуса робота,  $2a$  – расстояние между точками опоры,  $F$  – нелинейное отображение измеряемых (определяющих) параметров в определяемые.

Для решения поставленной задачи была выбрана искусственная нейронная сеть прямого распространения с одним скрытым слоем. Такие сети хорошо зарекомендовали себя в задачах аппроксимации функций [6].

Для корректной работы НС ее необходимо предварительно настроить. Для этого строится специальная обучающая выборка, состоящая из некоторого набора входов и соответствующих им (в смысле (3.1)) выходов. Сравнивая полученный отклик с желаемым на одном и том же наборе входных данных, сеть корректирует синаптические веса, настраиваясь на аппроксимацию конкретной функции. Остановимся подробнее на построении обучающей выборки для настройки рассматриваемой НС, так как от нее кардинально зависит качество работы полученной сети.

Решить обратную задачу, т.е. определить входы по выходам, проще, чем прямую, поэтому при построении обучающей выборки будем исходить из известных выходов и соответствующих им входов. Обучающую выборку будем получать прямым расчетом.

Определяемые и определяющие параметры связаны следующими нелинейными соотношениями (полужирным шрифтом отмечены векторы, обычным – скалярные величины):

$$\begin{aligned}
 AP^2 &= x^2 + (y + a)^2 + z^2, \\
 BP^2 &= x^2 + (y - a)^2 + z^2, \\
 \mathbf{P}_1\mathbf{A} &= (-x, -a - y, 0), \quad \mathbf{P}_1\mathbf{B} = (-x, a - y, 0), \\
 \cos(\alpha_{11}) &= \frac{(\mathbf{P}_1\mathbf{A}, \mathbf{e}_y)}{P_1A}, \quad \cos(\alpha_{21}) = \frac{(\mathbf{P}_1\mathbf{B}, \mathbf{e}_y)}{P_1B} \\
 \cos(\alpha_{12}) &= \frac{l_1^2 + l_2^2 - AP^2}{2l_1l_2}, \quad \cos(\alpha_{22}) = \frac{l_1^2 + l_2^2 - BP^2}{2l_1l_2}, \\
 \cos(\theta_1) &= \frac{(\mathbf{PK}, \mathbf{PR})}{PK \cdot PR}, \quad \cos(\theta_2) = \frac{(\mathbf{PK}, \mathbf{PL})}{PK \cdot PL}, \quad \cos(\theta_3) = \frac{(\mathbf{PR}, \mathbf{PL})}{PR \cdot PL}, \\
 \mathbf{PK} &= \sin(\alpha)\mathbf{e}_x - \sin(\beta)\mathbf{e}_y + \cos(\beta)\cos(\alpha)\mathbf{e}_z.
 \end{aligned} \tag{3.3}$$

Обучающее пространство и обучающие точки показаны на рис. 3.11.

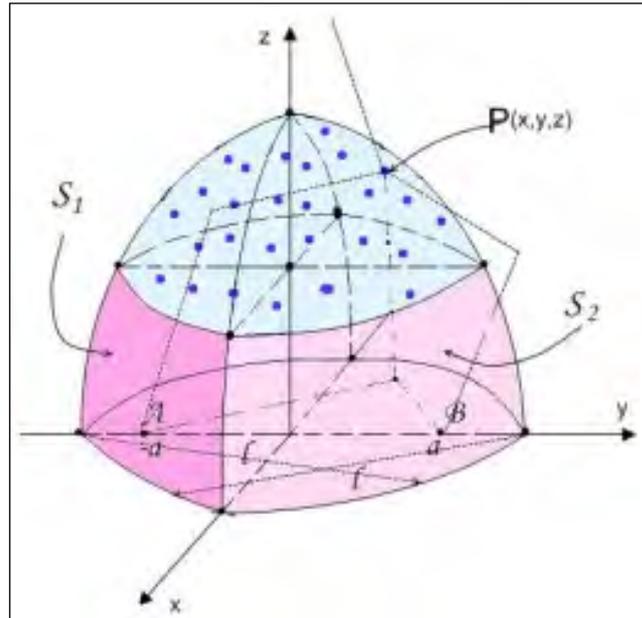


Рис. 3.11. Обучающее пространство.

Так как опорные точки неподвижны и длина ноги фиксирована, положение таза  $P$  не может быть произвольным – расстояния  $AP$  и  $BP$  не должны превосходить  $l$ . Таким образом, множество допустимых положений точки  $P$  представляет собой пересечение внутренних областей двух сфер  $S_1$  и  $S_2$ , как показано на рис. 3.11. Сфера  $S_1$  имеет центр в точке  $B$ , сфера  $S_2$  – в точке  $A$ . Радиусы обеих сфер равны полной длине ноги  $l$ .

Для построения обучающего множества использовалось обучающее пространство, соответствующее верхней области на рис. 3.11 при различных значениях параметра  $a$ . Это положения точки  $P$ . Навигационные углы для обучения брались в пределах от  $-\pi/4$  до  $\pi/4$ .

Точки обучающего множества выбирались случайным образом, что обеспечивает лучшее качество работы и скорость обучения сети по сравнению с равномерной сеткой.

### 3.3.4. Определение архитектуры сети

Для определения конкретной структуры сети был проведен ряд численных экспериментов. Рассматривались сети, содержащие различное число нейронов на скрытом слое в пределах от 10 до 20. Для каждой сети проводилась настройка на обучающих множествах разной мощности от 7000 точек до 20000 точек. По результатам тестов оценивались 2 критерия качества: точность и обобщающая способность полученной сети. Точность характеризует ошибку, с которой сеть аппроксимирует функцию в случае, когда тестовое множество мало отличается от обучающего, а способность к обобщению – качество аппроксимации на входных данных, сильно отличающихся от тех, на которых сеть обучалась.

Использовались тестовые множества трех видов:

1. Входные данные содержат малые ошибки (до 0.3 градуса)
2. Входные данные содержат большие ошибки (от 0.3 до 1 градуса)
3. Входные данные сильно отличаются от обучающих.

Для оценки точностей определения координат таза и навигационных углов использовался стандартный аппарат математической статистики. Матрицу входов будем называть  $I_{np}$ , матрицу выходов сети –  $O_{ut}$ , матрицу истинных значений выходных параметров –  $T_{ar}$ . Тогда  $E_{rr} = T_{ar} - O_{ut}$  – матрица отклонений. Эксперименты показывают, что для выборки в 10000 точек отклонения по каждой компоненте выходного вектора имеют почти симметричное распределение, близкое к нормальному с нулевым средним, как показано на рис. 3.12.

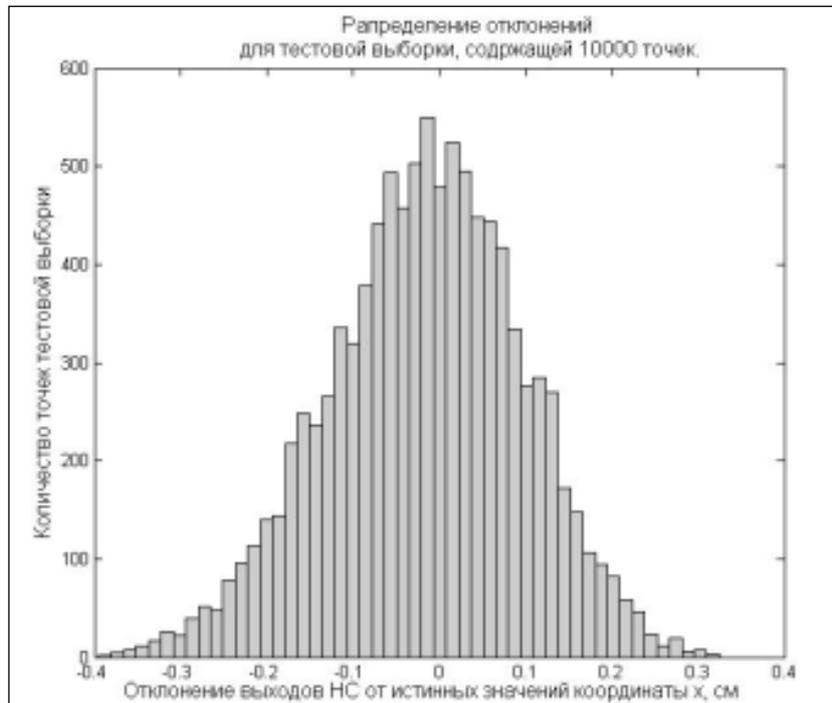


Рис. 3.12. Распределение точности по уровням погрешности датчиков.

Для оценки дисперсии этого распределения использовалась несмещенная выборочная дисперсия:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2 \quad (3.4)$$

В качестве оценки точности были выбраны границы 99% симметричного доверительного интервала. По итогам тестов было выбрано две сети следующих конфигураций:

1. 17 нейронов на скрытом слое, обучающая выборка содержит 9000 точек.
2. 15 нейронов на скрытом слое, обучающая выборка содержит 7000 точек.

Первая сеть имеет лучшие показатели точности при малых ошибках измерений (по результатам тестов первого типа), вторая менее чувствительна к росту ошибок и выходу за границы обучающего пространства (по результатам тестов второго и третьего типа). Графики зависимости точности от различных параметров приведены ниже на рис. 3.13- 3.15. Так как в задаче предполагается, что ошибки датчиков не превосходят 0.2 градуса, то для этой ситуации более интересна первая сеть. В результате был сделан выбор: нейронная сеть для решения задачи локальной навигации корпуса робота представляет собой сеть прямого распространения с одним скрытым слоем, содержащим 17 нейронов, с 8 входами и 5 выходами, всего в сети 30 нейронов.

### 3.3.5. Результаты численного моделирования и анализ чувствительности

В этом разделе представлены результаты численного моделирования построенной нейронной сети в системе MATLAB.

На рис. 3.13 наглядно представлен способ определения точности сети. На основе допущения, что раствор ног равен  $a = 30$  см, и соображений, высказанных в предыдущем разделе о характере поведения отклонений, построены доверительные интервалы для отклонений с доверительными вероятностями от 70% до 100%. В качестве точности взяты границы 99%-го доверительного интервала. Как видно на графиках, точность по координате  $y$  несколько хуже, чем по  $x$  и  $z$ . Это связано со свойствами обучающего пространства: по этой координате оно является более протяженным, чем по двум другим.

Эксперименты показывают, что зависимость точности от уровня шума при постоянном растворе ног практически линейная (рис. 3.14). Линейность сохраняется при относительно небольших погрешностях, дальше точность ухудшается быстрее. Зависимости точностей по угловым координатам также линейны.

Помимо чувствительности модели к шуму интересна также ее чувствительность к изменениям величины раствора ног  $a$  (рис. 3.15). Оказывается, что вдали от границ обучающего пространства ( $a = 20$  см и  $a = 60$  см) точность практически не зависит от  $a$ . На рис. 3.6 видно, что вблизи  $a = 20$  см точность по  $y$  заметно хуже, чем около  $a = 30 \dots 50$  см, что объясняется как близостью границы обучающего пространства, так и геометрическими соображениями, вытекающим из свойств обучающего пространства.

Кроме того, эксперименты показали, что точность практически не зависит от области пространства, в которой ведутся вычисления.

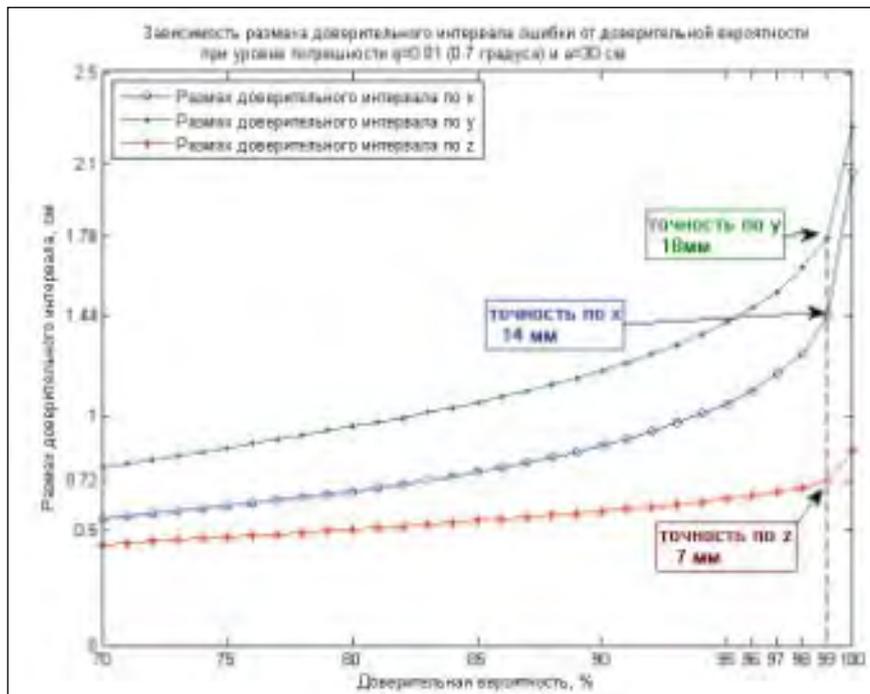
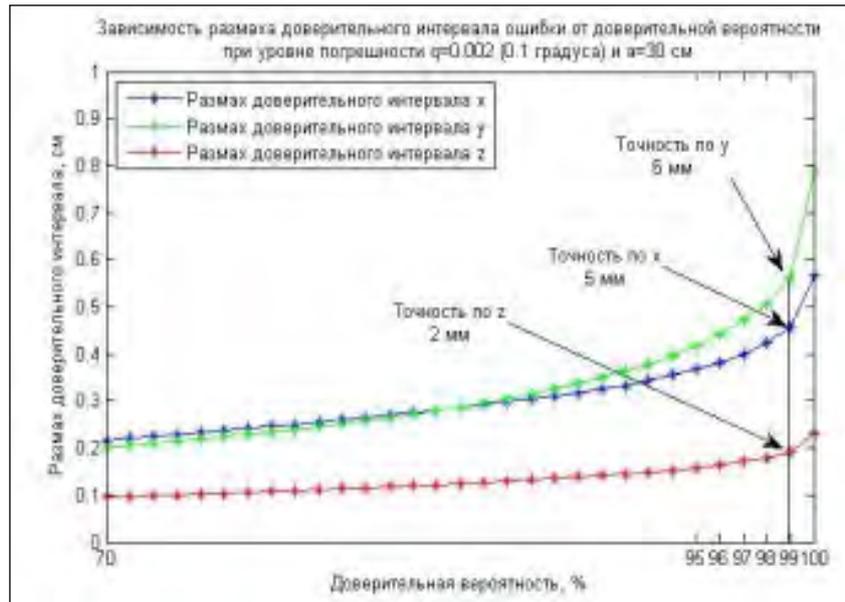


Рис. 3.13. Точность нейронной сети при уровне погрешности 0.1 и 0.7 градуса.

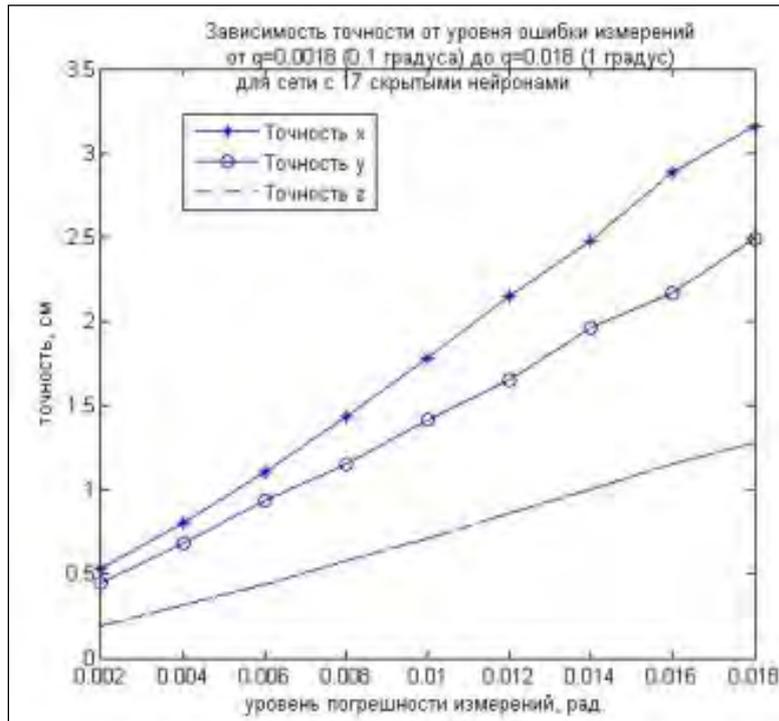


Рис. 3.14. Зависимость точности от уровня измерительных погрешностей (уровня шума).

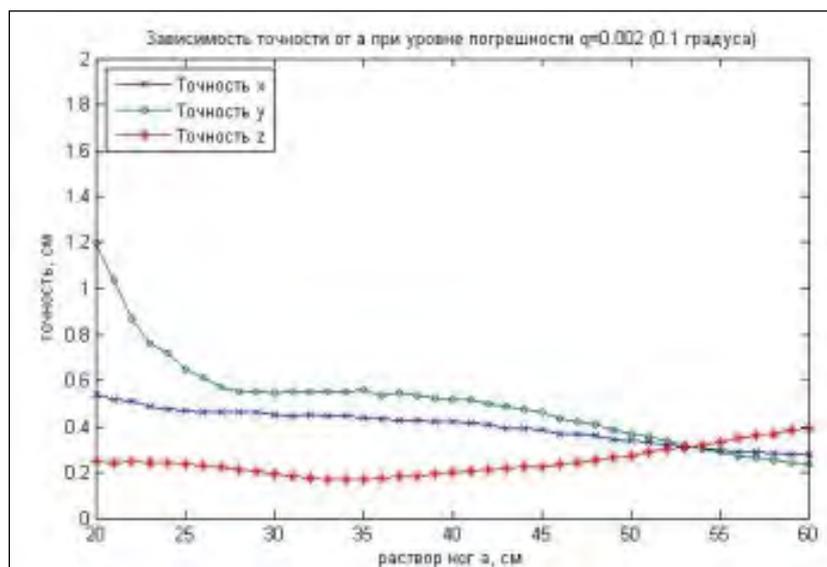


Рис. 3.15. Зависимость точности от раствора ног.

### 3.4. Метод локализации головы относительно корпуса.

Здесь задача состоит в восстановлении позы «головы», а именно углов, определяющих ее ориентацию относительно корпуса. Голова представляет собой твердое тело, закрепленное в одной точке (в точке крепления к корпусу). Положим, что измерениям доступны векторы линейных ускорений в двух точках головы. Такая схема моделирует работу вестибулярного аппарата человека по определению ориентации головы в пространстве (рис. 3.16).

Известно, что вестибулярный аппарат работает именно по измерениям ускорений в областях, расположенных в двух симметричных областях головы рядом с ушным аппаратом (фактически внутри него), но обрабатывает не только линейные, но и угловые (вращательные) ускорения, измеряемые в этих зонах. Далее в модели рассмотрим только обработку линейных ускорений.

Важно следующее. Здесь необходимо кроме обработки измерений с ошибками выполнять двойное интегрирование измерений, чтобы получить преобразование измеренных ускорений в навигационные углы.

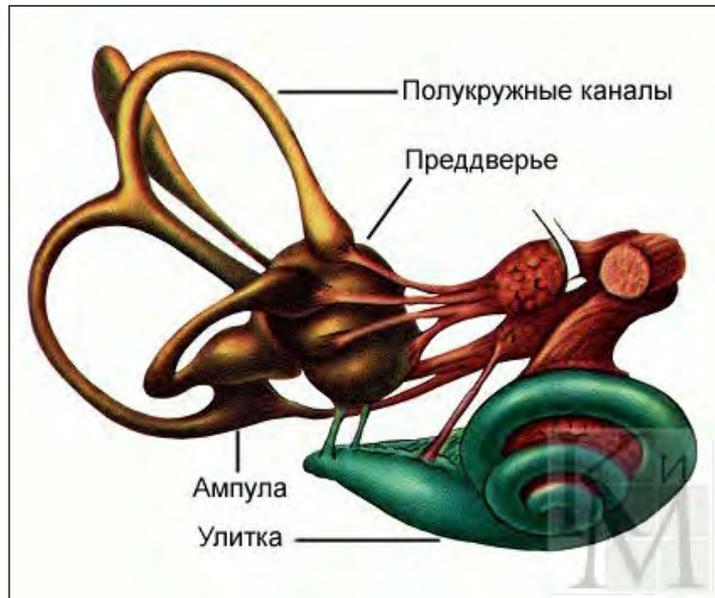


Рис. 3.16. Вестибулярный аппарат (по данным [7]).

### 3.4.1. Постановка задачи

Будем считать, что измерениям доступны ускорения в двух точках головы, измерения имеют некоторую ошибку, которую будем считать некоррелированной во времени.

Схема движения головы показана на рис. 3.17.

$K$  – точка опоры головы. Датчики, измеряющие линейные ускорения, находятся в характерных точках  $M$  и  $N$ . Точка  $C$  – центр головы.

Геометрические параметры:  $R = 15$  см,  $\rho = 7$  см.

$K\zeta\eta\zeta$  – абсолютная система координат, ее оси сонаправлены с осями системы координат, связанными с корпусом робота.

$Kxuz$  – относительная система координат, связанная с головой. Ось  $Kz$  направлена вдоль оси головы  $KC$ , ось  $Kx$  сонаправлена с вектором  $MN$ , а ось  $Ku$  дополняет их до правой тройки. В этой системе координат голова неподвижна.



Требуется восстановить позу головы, то есть найти:

$$\begin{aligned}\alpha(t_k) &= \alpha^*(t_k) + r_1(t_k), \\ \beta(t_k) &= \beta^*(t_k) + r_2(t_k), \\ \gamma(t_k) &= \gamma^*(t_k) + r_3(t_k),\end{aligned}\tag{3.7}$$

где  $\alpha(t_k)$ ,  $\beta(t_k)$ ,  $\gamma(t_k)$  суть истинные значения навигационных углов,  $\alpha^*(t_k)$ ,  $\beta^*(t_k)$ ,  $\gamma^*(t_k)$  – их численные аппроксимации, а также оценить ошибки вычислений  $r_{1,2,3}(t_k)$ .

### 3.4.3. Методика расчетов

Алгоритм вычислений кратко можно описать так:

1 шаг. Интегрируя ускорения в характерных точках, получаем их координаты.

2 шаг. Зная координаты характерных точек, пользуясь соотношениями 3.7), находим навигационные углы.

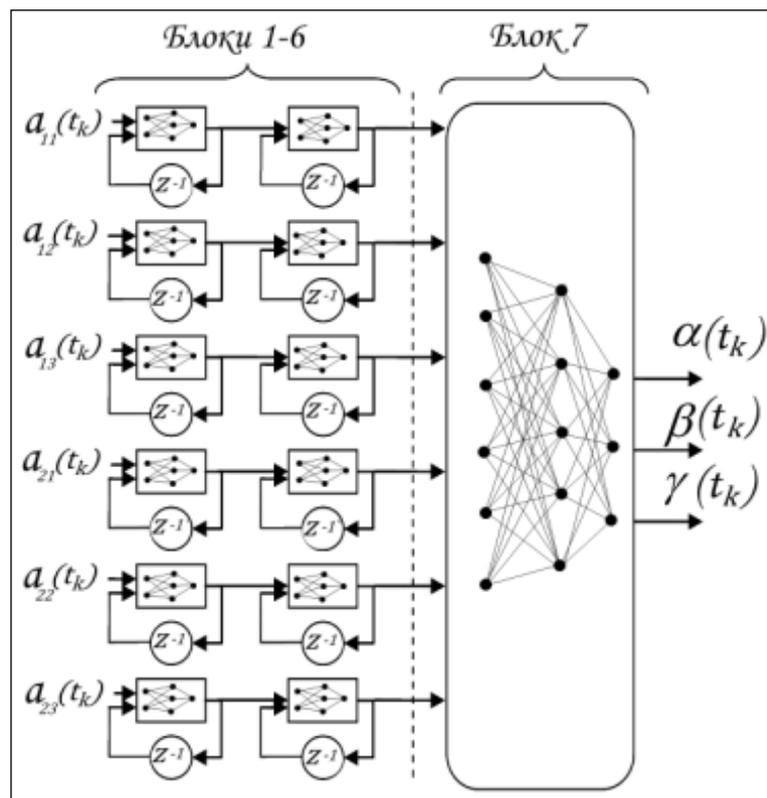


Рис. 3.18. Рекуррентная сеть для двойного интегрирования ускорений.

Для решения поставленной задачи была выбрана рекуррентная нейронная сеть, состоящая из 7 блоков. Общая схема архитектуры сети показана на рис. 3.18. Первые 6 блоков имеют одинаковую структуру и назначение, при помощи специальных обратных связей каждый такой блок преобразует одну из компонент линейных ускорений точек  $M$  и  $N$  в

соответствующую ей координату, т.е. выполняет двойное численное интегрирование. Эти координаты поступают на вход седьмого блока, который преобразует их в навигационные углы.

Рассмотрим подробнее, как устроены интегрирующие блоки. Схема одного из них изображена на рис. 3.19.

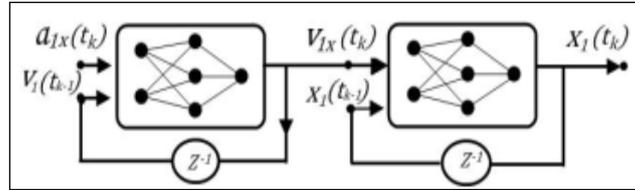


Рис. 3.19. Блок интегрирующей сети.

Блок представляет собой 2 последовательно соединенные сети Джордана, каждая из которых выполняет (однократную) операцию интегрирования. В момент времени  $t_k$  на вход первой из этих сетей поступает  $a_{1l}(t_k)$  – первая координата ускорения, измеренного в точке  $M$ , и значение  $v_{1l}(t_{k-1})$  – приближенное значение первой координаты скорости точки  $M$ , вычисленное сетью на предыдущем шаге и переданное на вход по линии обратной связи. На выходе сети имеем  $v_{1l}(t_k)$  – приближенное значение скорости в момент времени  $t_k$ . Вторая сеть действует аналогично, преобразуя входную скорость  $v_{1l}(t_k)$  в текущий момент времени и приближенное значение координаты  $x_{1l}(t_{k-1})$  в  $x_{1l}(t_k)$  – приближенное значение координаты  $x_{1l}$  точки  $M$  в момент времени  $t_k$ .

Таким образом, первые 6 блоков параллельно покомпонентно интегрируют ускорения точек  $M$  и  $N$ , давая на выходе приближенные значения их координат. Эти координаты поступают на вход второго блока, который выполняет нелинейное преобразование в углы ориентации «головы». Второй блок представляет собой сеть прямого распространения с одним скрытым слоем.

Формально работу сети можно описать следующими уравнениями:

$$\begin{aligned}
 v_{ij}(t_k) &= F_{ij}(a_{ij}(t_k), v_{ij}(t_{k-1})), \\
 x_i(t_k) &= G_{i1}(v_{i1}(t_k), x_i(t_{k-1})), \\
 y_i(t_k) &= G_{i2}(v_{i2}(t_k), y_i(t_{k-1})), \\
 z_i(t_k) &= G_{i3}(v_{i3}(t_k), z_i(t_{k-1})), \\
 (\alpha(t_k), \beta(t_k), \gamma(t_k)) &= H(x_1(t_k), x_2(t_k), y_1(t_k), y_2(t_k), z_1(t_k), z_2(t_k)).
 \end{aligned} \tag{8}$$

Начальные значения  $x_{1,2}(t_0)$ ,  $y_{1,2}(t_0)$ ,  $z_{1,2}(t_0)$  берутся из предварительной калибровки аппарата, а  $v_{i,j}(t_0)$  считаются нулевыми. Благодаря указанной блочной структуре сети каждый блок можно обучать отдельно от остальных.

При обучении сеть можно настроить на подавление определенного (небольшого) уровня ошибок во входных данных. Для этого необходимо во

входные данные обучающей выборки добавить шум, оставив выходные данные без изменений.

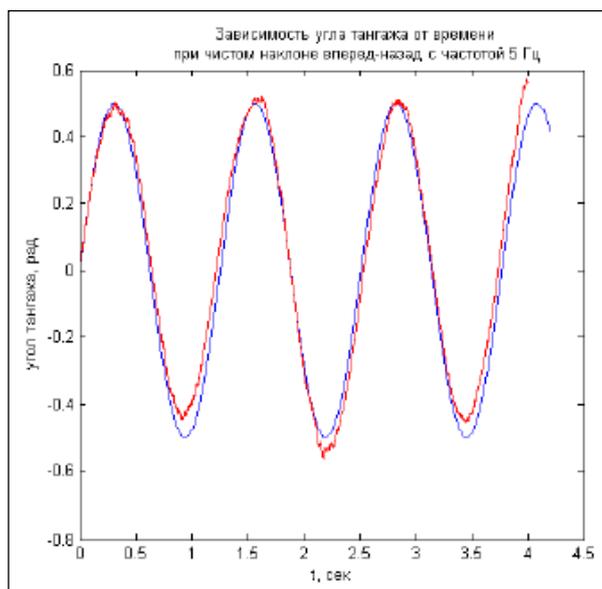


Рис. 3.20. Зависимость угла тангажа от времени при чистом наклоне вперед-назад.

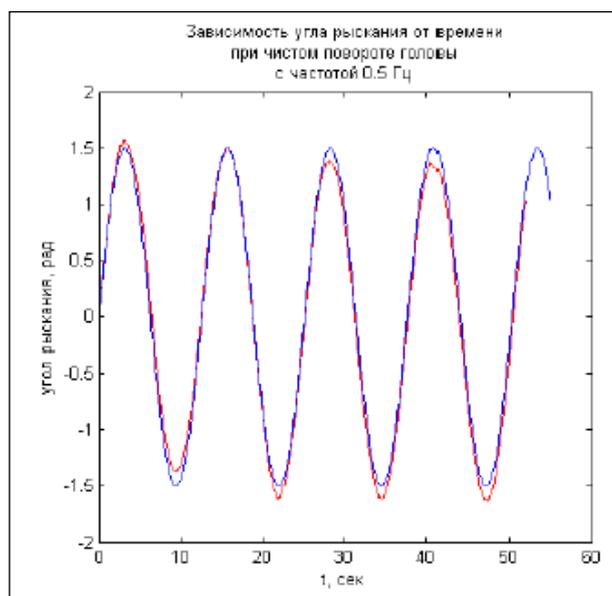


Рис. 3.21. Зависимость угла рыскания от времени при чистом повороте головы.

#### 3.4.4. Результаты численного моделирования и анализ чувствительности

Обучение и тестирование сети проводилось на базовых движениях головы, представляющих собой композиции гармонических колебаний (движений) по одному, двум или трем углам ее ориентации ( $\{8-10\}$ ). Для улучшения характеристик подавления шума на входах каждого блока

установлены так называемые. «шумовые пороги», которые позволяют не учитывать слишком малые входные данные и считать их нулевыми. Примеры работы сети приведены ниже, рис. 3.20-3.22.

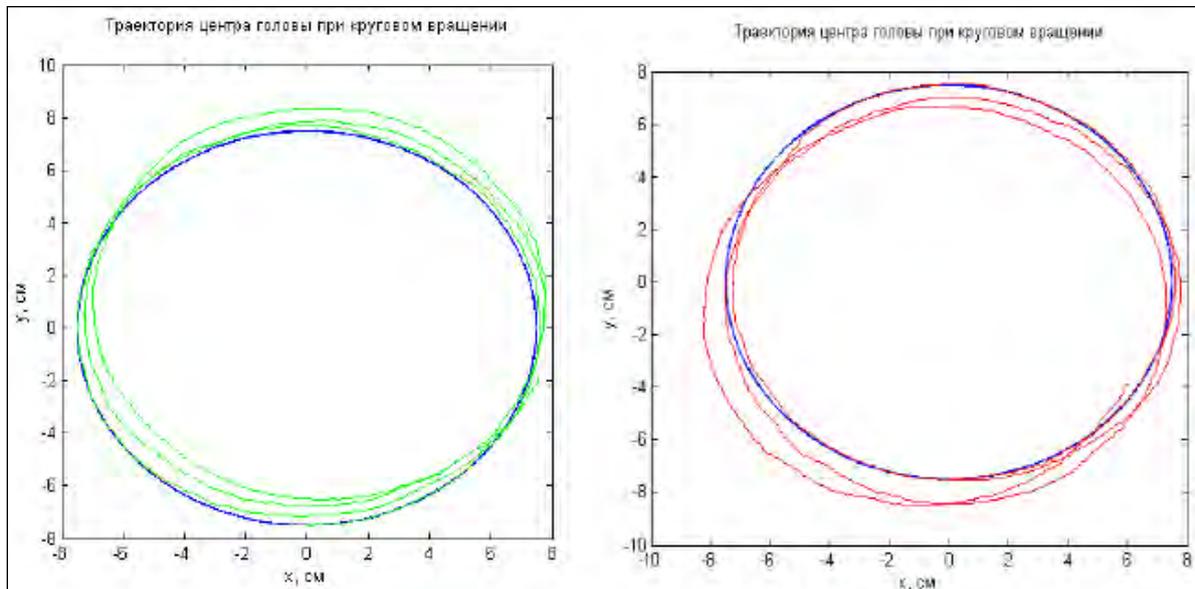


Рис. 3.22. Траектория центра головы и ее оценки при круговом вращении.

Основным недостатком построенной сети является постепенная потеря точности, как это видно на графиках. Накопление ошибки свойственно многим методам численного интегрирования. Однако стоит отметить, что благодаря специальной настройке с подавлением шума, точность сети все же не хуже, чем точность некоторых одношаговых методов аппроксимации решений дифференциальных уравнений. Сравнение проводилось с методом Рунге-Кутты 4 порядка. Этот метод показывает лучшую точность при условии более частых измерений (в зависимости от уровня шума – от 400 до 2000 Гц), однако при ограничении на минимальный шаг в 0.01 сек (что соответствует модельной частоте измерений 100 Гц), накопленная ошибка оказывается больше, чем у нейронной сети, и потеря точности при таком численном интегрировании происходит быстрее.

По результатам численного моделирования описанных трех задач можно уверенно сказать, что нейросетевой метод применим к решению задачи определения положения и позы шагающих аппаратов по проприоцептивным данным.

Для двуногого робота при заданном уровне погрешностей угломерных датчиков  $q = 0.1$  град. нейронная сеть, построенная для локализации двуногого робота, определяет положение и ориентацию корпуса с точностью до 5 мм по координатам и 0.2 градуса по углам. Анализ чувствительности этой сети к изменениям уровня погрешностей и некоторых других параметров выявил линейную зависимость точности от уровня погрешности, а также хорошую обобщающую способность построенной сети.

При заданном уровне погрешностей акселерометров и частоте измерений 100 Гц нейронная сеть двойного интегрирования определяет ориентацию головы относительно корпуса с точностью 0.5-1 град., однако характеризуется постепенным накоплением ошибки.

При заданном уровне погрешностей угломерных сенсоров  $q = 0.1$  град. нейронная сеть, построенная для локализации шестиногого робота, определяет положение и ориентацию корпуса с точностью до 1 см по координатам и 0.2 град. по углам. Анализ чувствительности этой сети к изменениям уровня погрешностей и некоторых других параметров выявил линейную зависимость точности от уровня погрешности, а также хорошую обобщающую способность построенной сети.

Все это позволяет сделать вывод о достаточной точности нейросетевой навигационной системы и эффективном решении навигационной задачи в нейросетевом базисе.

### **Литература**

1. Маккалок Дж., Питтс У. Логические исчисления идей, относящихся к нервной деятельности. // Автоматы. М.: ИЛ, 1956.
2. <https://myslide.ru/presentation/skachat-nechetkie-mnozhestva-operacii-nad-nechetkimi-mnozhestvami>
3. Эл. ресурс: <http://www.machinelearning.ru/wiki/index.php?title=Перцептрон>
4. Эл. рес.: <https://d3d3rmhotmx78.cloudfront.net/ru/photos/нейронная%20сеть/>
5. Голубев Ю.Ф. Нейросетевые методы в мехатронике. – Издательство МГУ, 2007 г., 158 с.
6. Хайкин С. Нейронные сети: Полный курс. - Пер. с англ. Куссуль Н.Н., Шелестова А.Ю. Изд. 2-е, Изд-во Вильямс, 2006 г., 1104 с.
7. Siegelmann H.T., B.G. Home and C.L. Giles. “Computational capabilities of recurrent NARX neural networks”, Systems, Man and Cybernetics, Part B: Cybernetics, 1997, vol. 27, p. 208-215.
8. Электронный ресурс: [http://ru.wikipedia.org/wiki/Вестибулярный\\_аппарат](http://ru.wikipedia.org/wiki/Вестибулярный_аппарат)
9. Павловский В.Е., Шелике А.Г. Нейросетевые навигационные системы для андроида робота // Тр. VI Международной научно-практической конф. «Интегрированные модели и мягкие вычисления в искусственном интеллекте», 16-19 мая 2011, Коломна. М.: Физматлит, т.1, с.427-439.
10. Павловский В.Е., Шелике А.Г. Нейросетевые системы локализации для андроида робота // Препринты ИПМ им.М.В.Келдыша. 2012. № 84. – 16 с.
11. Pavlovsky V.E. Biologically inspired neural-network navigation systems for walking robots // Abstracts of the Int. Conf. “Biologically Inspired Cognitive Architectures (BICA-2013)”. 13-15.09.2013. Киев, Украина. pp.1-2.

## 4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИСУ НА БАЗЕ СТЗ

### 4.1. Функции СТЗ в составе ИСУ

В различных областях человеческой деятельности все более активизируется спрос на автоматизацию сбора и обработки зрительных данных. Особенно эта активизация заметна в робототехнике.

В сфере автоматизированной обработки зрительных данных можно выделить ряд типичных задач обработки зрительной информации, возникающих в рамках исследовательских проектов, а также в промышленных и потребительских приложениях массового применения. Характерно, что в этих задачах требуется обработка зрительных данных именно в реальном или квазиреальном времени (рис. 4.1). Примерами таких задач являются задачи слежения за движущимися объектами, информационные зрительные системы в составе робототехнических комплексов, инспекционный технологический контроль, системы контроля доступа и охраны, стереоскопические системы для трехмерного сканирования в лабораторных условиях и др. В условиях, когда требуется обеспечить надежное и оперативное решение похожих задач, становится актуальным вопрос об использовании типовых решений. Сферой применения типовых решений может быть как этап анализа предметной области в процессе решения конкретной прикладной задачи, так и этап реализации программной и аппаратной частей разрабатываемой системы.



*Рис. 4.1.* Примеры изображений, обрабатываемых СТЗ реального времени в составе системы помощи водителю. СТЗ выполняет обработку изображений ИК-диапазона.

Под системой технического зрения (СТЗ) будем понимать программно-аппаратный комплекс, предназначенный для обработки зрительных данных в режиме реального времени. В ПО СТЗ должно быть предусмотрено решение следующих основных задач:

- сбор зрительных данных с целью формирования цифровых изображений;

- обработка цифровых изображений для извлечения из них информации в соответствии с конкретной решаемой задачей;
- использование результатов обработки изображений.

СТЗ являются системами реального времени и обладают характерными особенностями, присущими этому классу программно-аппаратных систем, а также имеют ряд особенностей, связанных со спецификой обработки зрительных данных. Хотя в состав СТЗ входят и программные, и аппаратные компоненты, дальнейшее изложение в основном относится к рассмотрению только программного обеспечения СТЗ.

Общие свойства систем реального времени (СРВ) анализируются в работах, посвященных разработке ПО СРВ [78, 84, 87, 97]. Характеристика «работа в режиме реального времени» обычно применяется для программных систем, которые должны реагировать на внешние входные воздействия, причем задержка и длительность реакции должны быть конечными и не превышать заданного значения. В некоторых задачах реального времени получение результата за время, превышающее требуемое, может быть приравнено к вычислению неверного результата [97].

Различные процессы во внешней для СРВ среде, информация о которых поступает на вход СРВ, могут протекать параллельно. В ПО, взаимодействующем с таким процессом, должна учитываться эта параллельная природа, а в некоторых ситуациях функционирование СРВ должно выполняться в соответствии с ней [87]. Таким образом, системы реального времени в целом можно охарактеризовать как параллельные программные системы с временными ограничениями. В состав типичной СРВ входит программа, функционирующая в режиме реального времени, операционная система реального времени и подсистема ввода/вывода реального времени. В состав этой подсистемы входят драйверы специальных устройств, управляющие работой различных датчиков и исполнительных устройств. Сама СРВ может быть частью более крупного комплекса [78].

Особенностью СРВ является то, что в них редко удается точно предсказать частоту и последовательность поступления сигналов о внешних событиях, но программная система должна достаточно быстро реагировать на них, чтобы соблюсти временные ограничения, сформулированные в требованиях к системе. Кроме того, входная нагрузка может с течением времени значительно и неожиданно изменяться [78]. Спецификой СТЗ является то, что информация о внешних событиях часто не может быть получена за гарантированное время (например, непосредственным считыванием показаний датчиков каких-либо физических величин), а поступает в СТЗ в результате программной обработки зрительных данных, получаемых от зрительного датчика.

По критерию длительности реакции среди СРВ могут быть выделены категории систем с жесткими и слабыми временными ограничениями. Система с жесткими ограничениями должна отреагировать на событие в пределах установленного временного интервала, в противном случае возможен аварийный отказ. Для систем со слабыми временными

ограничениями выход за пределы допустимого интервала считается нежелательным, но не катастрофическим явлением [78].

К СРВ с жесткими временными ограничениями, например, относятся СРВ, связанные с управлением каким-либо техническим устройством или физическим процессом (лабораторные измерения, телекоммуникационное обслуживание, управление процессами, аэронавигация, управление промышленными роботами и др.) [97] Среди СТЗ также можно выделить системы с жесткими и слабыми временными ограничениями. Например, к первой категории относятся СТЗ в составе систем управления робототехническими комплексами и системы контроля доступа и охраны, а ко второй – системы человеко-машинного интерфейса на основе обработки зрительных данных, системы инспекционного контроля с управляемой от СТЗ подачей контролируемых объектов в рабочее пространство СТЗ и т.п. Для СТЗ обеих категорий время реакции на внешние события зависит от двух факторов: способности ПО СТЗ обеспечить обработку поступающих зрительных данных в темпе их поступления от зрительного датчика, а также от надежности алгоритмов обработки изображений в плане обнаружения на изображениях образов объектов интереса.

Система реального времени должна достаточно быстро реагировать на внешние события и постоянно обрабатывать поток входных данных, чаще всего не имея возможности изменить их количество или скорость поступления. Одновременно может потребоваться и выполнение других операций, например, обмен информацией с оператором, вывод данных на экран и реакция на определенные сигналы [87]. Несмотря на необходимость обработки входных данных в реальном времени, в СРВ могут быть некритичные по времени компоненты. Например, сбор данных может быть критичной операцией, а их анализ и отображение может быть отложен до момента снижения загрузки процессора базового компьютера. [78] Наличие процессов с разными приоритетами характерно и для СТЗ, особенно в связи с тем, что в ПО СТЗ могут функционировать алгоритмы с неоднородным распределением временных затрат при обработке последовательности изображений. Например, в процессе слежения за движущейся целью первоначальное обнаружение цели без априорной информации о ее местоположении на изображении может потребовать времени, превышающем ввод следующего кадра последовательности. Но для обработки кадров в режиме сопровождения обнаруженной ранее цели время обработки кадра может быть существенно меньшим за счет применения алгоритма обработки, отличающегося от алгоритма обработки изображений на этапе первоначального обнаружения цели. В таких условиях на этапе первоначального обнаружения в СТЗ слежения ресурсы процессора должны в основном выделяться потоку обработки зрительных данных, а на этапе сопровождения могут быть также выделены потоку, реализующему интерфейс пользователя (оператора СТЗ).

Базовый компьютер СРВ должен работать со скоростью, соответствующей масштабу реального времени системы. Ход исполнения ПО

СРВ обычно не удается определить заранее. Внешние сигналы могут прерывать или изменять последовательность исполнения операторов программы, причем для каждого нового сеанса работы по-разному. Кроме того, существует проблема эффективного использования ресурсов компьютерной системы с учетом временных ограничений [87]. Все перечисленные особенности ПО СРВ характерны и для ПО СТЗ. Дополнительную сложность в ПО СТЗ в данном случае составляет то, что зрительные данные образуют потоки данных большого объема, которые обычно не удается целиком сохранять в дисковой памяти (не нарушая при этом функционирования ПО СТЗ в реальном времени) для анализа после завершения работы СТЗ. Это создает сложности на этапе отладки и тестирования ПО СТЗ.

Особенности СТЗ требуют учета при разработке методов программирования, направленных на оперативную разработку ПО СТЗ реального времени. В качестве характерных особенностей ПО СТЗ можно перечислить следующие:

- В состав СТЗ входит зрительный датчик – источник зрительных данных (возможно, наряду с другими датчиками). Свойства источника зрительных данных обычно задают характерный масштаб реального времени СТЗ.
- В ПО СТЗ должны поддерживаться параллельные процессы (обычно это процесс обработки зрительных данных, процесс интерфейса пользователя и процессы для обслуживания устройств ввода-вывода).
- СТЗ должна предоставлять оператору актуальную информацию о текущем состоянии и о ходе обработки зрительных данных.
- Зрительные данные в форме цифровых изображений обычно не могут непосредственно использоваться для получения информации о внешних событиях в форме, пригодной для использования в ПО СТЗ. Для получения такой информации требуется программная (или программно-аппаратная) обработка цифровых изображений.
- Обработка цифровых изображений в ПО СТЗ может выполняться с различными временными затратами в зависимости от текущего состояния СТЗ и от результатов обработки предыдущих изображений.

При разработке современного ПО для упрощения поддержки и модификации широкое распространение получил объектно-ориентированный подход. На него возлагались надежды и в плане улучшения повторной используемости ПО, но в данной области решающих преимуществ пока не было получено. Высокая степень повторной используемости характерна для применения устойчивых библиотек компонент в рамках определенных компонентных технологий – Microsoft COM/COM+ [72], Microsoft .NET [89], Enterprise JavaBeans [86]. Применение этих технологий для систем реального времени пока остается предметом для дальнейших исследований.

В работе [95] отмечается, что для 1990-х гг. характерны первые реальные попытки привнести в разработку программного обеспечения концепции компонентной разработки программного обеспечения (CBSE, component-based software engineering) с применением готовых доступных компонентов (COTS, commercial off-the-shelf). Идея состоит в создании небольших, высококачественных модулей и последующего их объединения. Проблема, безусловно, заключается в том, что объединенные вместе высококачественные модули не обязательно превратятся в высококачественную систему. Успешное решение данной проблемы возможно путем построения компонент на основе анализа ряда конкретных задач.

Под «компонентом» будем понимать объект с четко определенным интерфейсом, который может применяться в различных приложениях, а не только в том, для которого он проектировался. Компонентная система – система, инфраструктура которой специально разработана для использования готовых компонентов [78].

Чтобы полностью описать компонент, необходимо определить операции, которые он предоставляет, и операции, которые он потребляет. При традиционном объектно-ориентированном подходе нужно описать лишь операции, которые объект предоставляет [73]. Однако, если требуется интегрировать готовый компонент в систему, составленную из компонентов, то не менее важно знать (и, значит, явно документировать) операции, которые ему требуются. Развитие компонентного подхода обусловлено желанием удовлетворить фундаментальным мотивам модуляризации программного обеспечения: систематизации и упрощению разработки, многократному использованию и изменямости программы [79].

#### **4.1.1. Особенности реализации ПО ИСУ с использованием СТЗ**

Далее рассматриваются методы программирования, предназначенные для разработки ПО ИСУ с использованием СТЗ реального времени. Подобные СТЗ обычно являются малотиражными или имеют исследовательский характер. Применительно к разработке ПО данного класса методы программирования должны учитывать необходимость решения типичных проблем разработки, в особенности:

- Необходимость быстрого прототипирования.
- Повторное использование программных компонентов.
- Сопровождаемость и удобочитаемость исходного текста.

А также проблем, связанных со спецификой ИСУ РТК:

- Функционирование в режиме реального времени. В качестве одного из ключевых элементов ПО ИСУ можно выделить цикл реального времени. В нем должно обеспечиваться получение данных от различных устройств, обработка этих данных и использование результатов обработки. Представляется возможным сформировать

заготовку такого цикла, которую можно оформить в виде компонента, нуждающегося в модификации для конкретной системы.

- Надежность и расширяемость ПО. Для расширяемости программы необходимо спрогнозировать направления ее будущего развития, облегчить и обезопасить выполнение изменений в этих направлениях [79]. Например, в ПО СТЗ прогнозируемыми точками роста являются цикл реального времени и алгоритмы обработки изображений.
- В алгоритмах обработки изображений должны учитываться особенности конкретной прикладной задачи. Желательно формирование компонентов алгоритмов обработки, которые можно будет использовать при разработке алгоритмов обработки изображений при решении задач определенного класса (например, для обнаружения объектов с прямолинейными краями или для слежение за объектами, образы которых движутся на изображении с постоянной скоростью, и т.п.). Информация о внешних событиях в СТЗ формируется в результате обработки изображений. Поэтому ключевое значение имеет разработка алгоритмов обработки, удовлетворительных для решения конкретной задачи в смысле надежного выделения информации о содержании наблюдаемых сцен по данным цифровых изображений.
- Интерфейс пользователя должен обеспечивать удобное взаимодействие оператора и СТЗ как в режиме реального времени, так и в режиме настройки. Пока полностью отказаться от участия человека в обработке зрительных данных не удастся, поэтому затруднительно обеспечить полностью автоматическую работу СТЗ. Для учета этого факта при разработке ПО СТЗ можно предусмотреть в нем два режима работы – режим настройки системы и режим автоматической обработки в реальном времени.
- Требования к СТЗ в части содержания обработки зрительных данных обычно сложно точно выделить на этапе проектирования СТЗ. В связи с этим требуется предусмотреть средства отладки и тестирования ПО СТЗ, позволяющие, кроме устранения ошибок ПО, выполнять уточнение требований к алгоритмам обработки изображений на этапе разработки СТЗ и на этапе опытной эксплуатации СТЗ.
- Предполагаемый срок разработки типичной СТЗ лежит в пределах 6-10 месяцев при разработке 1-3 программистами.

В качестве основы для разработки ПО ИСУ может быть использован компонентный каркас ПО, построенного с применением объектно-ориентированного подхода. Под программным каркасом понимается многократно используемая ведущая часть программы, которая в соответствии с программной логикой при необходимости обращается к различным модулям

(в т.ч. повторно используемым) для решения частных подзадач. В таком случае при разработке конкретной программы разработчику необходимо заново написать лишь некоторые из модулей, заполняющих гнезда готового каркаса [79].

В описываемом каркасе ПО СТЗ отражена структура СТЗ реального времени, сформированная по результатам решения более 10 различных прикладных задач автоматизированной обработки зрительных данных. После рассмотрения компонентных подходов, использованных при разработке каркаса, и собственно каркаса ПО СТЗ, в работе описывается его применение для разработки нескольких конкретных СТЗ реального времени.

#### 4.1.2. Примеры СТЗ реального времени

В качестве основной подсистемы для построения ИСУ РТК будем рассматривать СТЗ РВ. Для построения компонентного каркаса ПО СТЗ был проанализирован ряд прикладных задач обработки зрительных данных в реальном времени. В результате анализа этих и ряда других разработанных СТЗ оказалось возможным построение структуры ПО СТЗ реального времени, на основе которой был сформирован компонентный каркас ПО СТЗ. Реализованные СТЗ по областям применения можно разделить на три категории:

- Мониторинг состояния механических систем. СТЗ этой категории предназначены для определения параметров состояния механических систем в целях последующего анализа поведения этих систем или для непосредственного использования полученных данных в ПО СТЗ. К этим СТЗ, в частности, относятся: СТЗ для определения взаимного расположения частей пантографов в движении (рис. 4.2), СТЗ для определения местоположения контактного провода железной в процессе движения электровоза, СТЗ для определения колебаний контактного провода железной дороги, СТЗ для обнаружения дефектов верхнего строения железнодорожного пути.
- Инспекционный технологический контроль. Выходной контроль продукции на современном автоматизированном производстве однотипных объектов остается актуальной задачей. Из-за дефектов в исходном сырье и погрешностей производственного оборудования сложно обеспечить 100% качество выходной продукции. Сплошной контроль является рутинной задачей, и при больших объемах производства человек не в состоянии обеспечить нужного качества контроля. Цель применения СТЗ данной категории заключается в определении в производственных условиях качества производимой продукции и в обнаружении бракованных изделий (например, СТЗ для контроля качества металлического порошка в порошковой металлургии, рис. 4.3).
- Слежение за подвижными объектами. СТЗ этой категории должны обнаруживать на изображениях образы объектов слежения (целей) и выполнять прослеживание этих целей при условии их видимости в поле

зрения СТЗ. Полученная информация о движении цели может использоваться для передачи в системы управления какими-либо устройствами (если СТЗ применяется в составе системы управления), либо предоставляться оператору в удобном для восприятия виде. К СТЗ данной категории относятся: СТЗ для слежения за низколетящими целями на расстояниях прямой видимости, СТЗ в составе системы «глаз-рука», выполняющей захват подвижного объекта или формирование его траектории, СТЗ для слежения за движущейся малоразмерной целью, СТЗ для слежения за процессом сближения и стыковки космического корабля с Международной космической станцией (рис. 4.4).

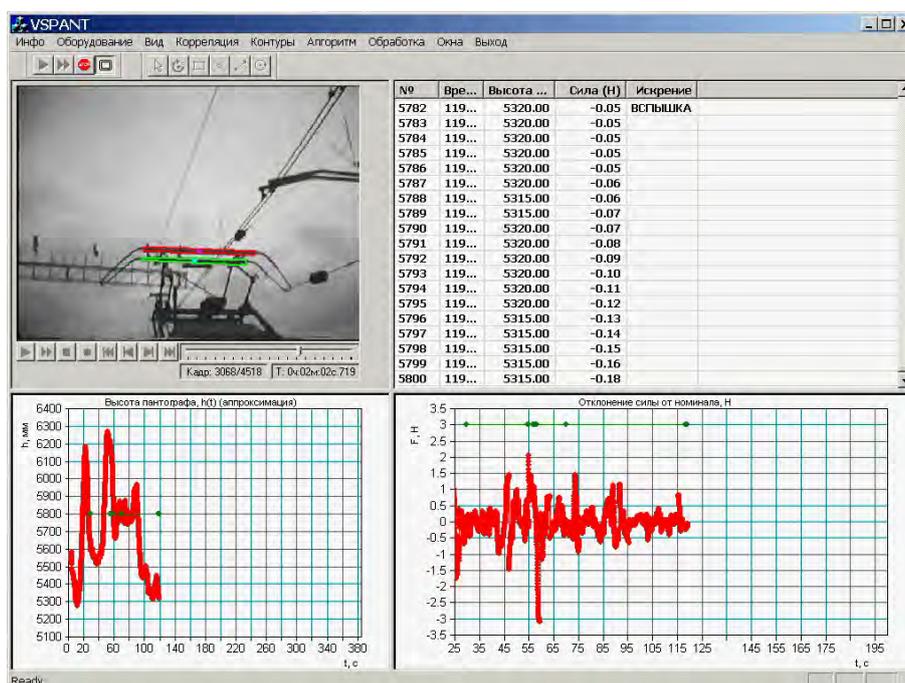


Рис. 4.2. Основной экран СТЗ для определения взаимного расположения частей пантографа в движении.

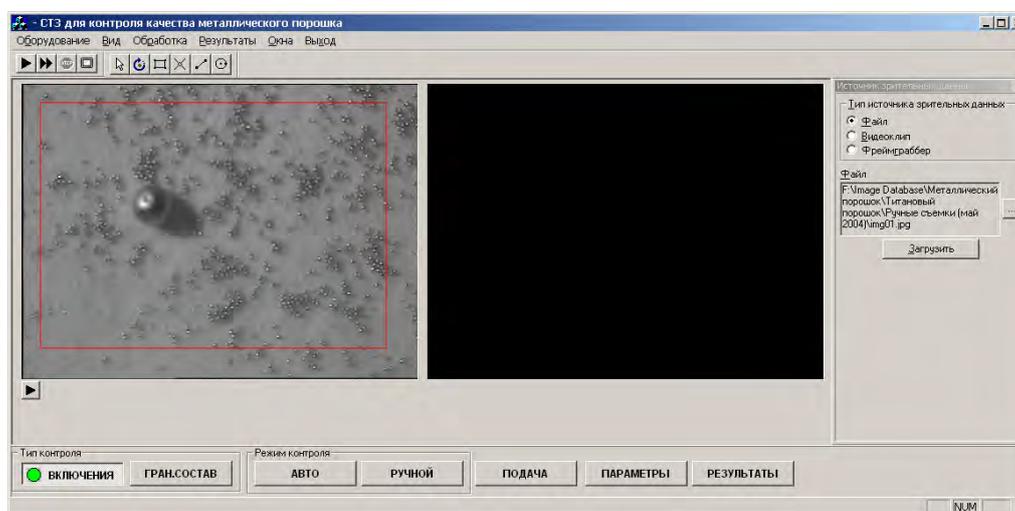


Рис. 4.3. Основной экран СТЗ для контроля качества металлического порошка в порошковой металлургии, рис. 4.3

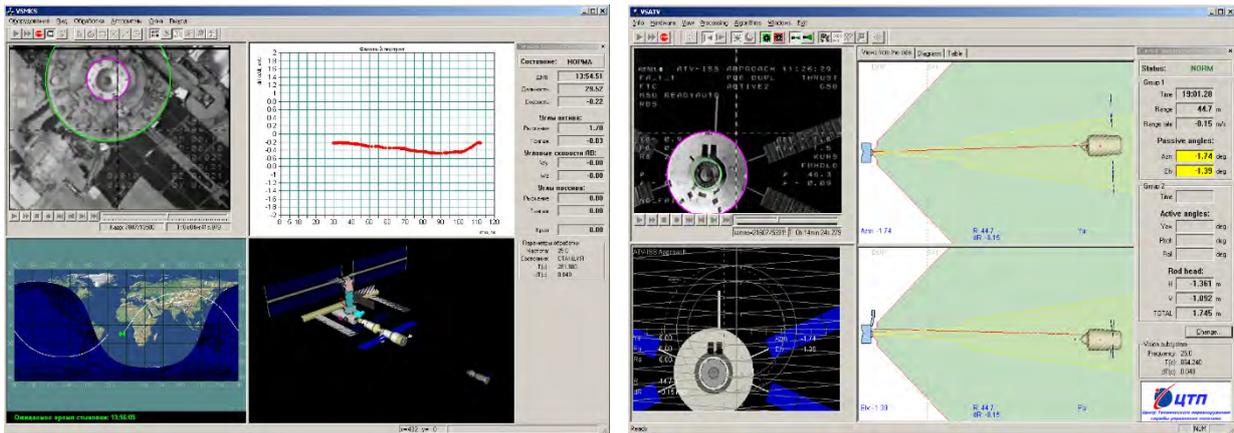


Рис. 4.4. Основной экран СТЗ для слежения за процессом сближения и стыковки космических кораблей с Международной космической станцией.

Специфическую часть каждой СТЗ образует алгоритм обработки изображений, в котором, тем не менее, возможно выделение компонентов для повторного использования. Эти компоненты относятся к процессу построения гипотез, проверяемых с целью обнаружения объектов интереса. Объекты интереса могут быть составными. Образы на изображениях могут быть представлены в виде различных компонентов ПО, например, в виде объектов классов на объектно-ориентированном языке программирования. Высокоуровневые объекты интереса специфичны для разных СТЗ, а их составные части, например, связанные компоненты изображений, могут быть однотипными для нескольких различных СТЗ. Кроме сущностей, связанных с образами на изображениях, в ПО СТЗ можно выделить сущности, связанные с алгоритмической обработкой изображений.

Приведенные соображения были использованы при построении каркаса ПО СТЗ. Основное направление его применения – быстрая разработка прототипа СТЗ для того, чтобы основные усилия разработчика могли быть направлены на проектирование и реализацию алгоритма обработки зрительных данных и цикла реального времени СТЗ.

## 4.2. Объектно-ориентированный подход к разработке ПО

На протяжении всего периода использования вычислительной техники наблюдается постоянное увеличение сложности программных систем. В связи с этим в компьютерных науках в качестве одной из областей исследований можно выделить исследования с целью разработки методов проектирования ПО, позволяющих в предсказуемые сроки разрабатывать программные системы с возможностью их дальнейшей модификации и сопровождения. Под методом проектирования ПО можно понимать последовательный процесс создания моделей, которые описывают различные стороны разрабатываемой программной системы [73]. Методы разработки ПО позволяют упорядочить процесс создания сложных программных систем, выполняя роль общих средств, доступных для всей группы разработчиков ПО.

С распространением языков высокого уровня в 1960-1970-е гг. возможности разработчиков ПО существенно расширились, по сравнению с применением машинных языков и ассемблера, но вместе с тем увеличился уровень сложности программных систем [73]. В связи с этим разрабатывались различные методы проектирования ПО и программирования. Наибольшее распространение среди них получило структурное проектирование методом «сверху-вниз» [75, 80]. Этот метод был непосредственно основан на существующих процедурных языках высокого уровня (например, Фортран и Кобол). Основным типом программных компонентов в этих языках является подпрограмма. Программа в целом принимает форму дерева, в виде которого можно представить вызов одних подпрограмм другими в процессе работы программы. Такой подход используется при структурном проектировании ПО: для разбиения большой задачи на более мелкие применяется алгоритмическая декомпозиция [73].

Еще одним получившим широкую известность методом проектирования ПО является метод потоков данных. Метод потоков данных, как и структурный метод, с успехом применялся при решении ряда сложных задач, например, в системах информационного обеспечения, в которых существуют прямые связи между входными и выходными потоками системы и где не требуется уделять особого внимания быстрдействию [73]. Известны примеры применения этого метода для разработки алгоритмов обработки изображений в интерактивных программных пакетах, например, в пакетах AdOculus [1], MAVIS [44], Pisoft Image Framework [48].

С увеличением объема разрабатываемых программных систем оказалось, что структурный подход не позволяет удовлетворительно справляться с разработкой систем значительного объема (порядка 100000 строк исходного текста и более) [60, 73]. Объектно-ориентированный подход к разработке ПО стал в последнее десятилетие общепринятым инструментом для разработки сложных программных систем, в т.ч. в условиях изменения требований к системе в процессе разработки. В основе объектно-ориентированного подхода к проектированию ПО лежит представление о том, что программную систему необходимо проектировать как совокупность взаимодействующих друг с другом объектов. Каждый объект рассматривается как экземпляр определенного класса. Различные классы могут быть связаны друг с другом. Эти связи отражают отношения между соответствующими понятиями предметной области или структуры ПО. Для разработки программ с применением объектно-ориентированного подхода используются такие языки программирования, как Smalltalk, Объектный Паскаль, Си++, Java, Oberon и др. [73].

В монографии [40] в объектно-ориентированном подходе (ООП) к разработке ПО выделяются четыре основных применяемых метода: метод структурирования ПО, метод обеспечения надежности, метод выделения абстракций, и метод классификации.

Метод структурирования применяется на этапе декомпозиции разрабатываемой программной системы и на этапе повторного использования

ПО. Программная система проектируется таким образом, чтобы все выполняемые системой операции можно было представить в виде наборов сообщений, передаваемых одними объектами другим с целью выполнения действий, предусмотренных классами этих объектов. Предполагается, что использование в качестве базовых программных компонентов объектных типов (классов), а не действий (подпрограмм), позволит повысить уровень повторного использования компонентов ПО. Классы служат в качестве основных компонентов и для модульной структуры ПО, и для системы типов данных.

Метод обеспечения надежности предназначен для решения задачи построения компонентов ПО с замкнутой функциональностью. Идея этого метода состоит в том, чтобы рассматривать любую систему как набор компонентов (например, объектов или подсистем), которые взаимодействуют между собой согласно специфицированным протоколам. Эти протоколы, или, интерфейсы, определяют обязанности, выполняемые компонентами. Для встраивания готового компонента в систему, составленную из компонентов, также необходимо знать операции, которые ему требуются со стороны других компонентов [95].

Метод выделения абстракций относится к вопросу, как можно описать задачу разработки ПО в терминах классов. В ООП объекты описываются классом, в структуре которого содержатся операции и атрибуты. Совокупность операций представляет собой интерфейс класса и определяет поведение объектов данного класса. Все объекты, принадлежащие одному классу, характеризуются одинаковым поведением. Отличие между объектами одного класса заключается в состоянии объектов, которое представляется посредством значений атрибутов класса. Существенно, что объекты соответствуют не только понятиям предметной области, но также моделируют понятия, необходимые только для реализации программной системы (например, фрагменты инфраструктуры программной системы, такие, как организация доступа к базам данных, взаимодействие с операционной системой, устройствами и др.).

Метод классификации позволяет сгруппировать связанные по смыслу понятия и таким образом ввести структуру в программное представление предметной области задачи. Метод классификации в ООП основан на определении отношений между классами. Наиболее часто используемыми отношениями являются отношение «общее/частное», выражаемое с помощью механизма наследования, и отношение «целое/часть», выражаемое с помощью механизма агрегации. Другими допустимыми видами отношений являются ассоциация, использование и параметризация [69, 73].

ООП является общим подходом к проектированию и разработке ПО. Применительно к разработке ПО СТЗ можно отметить необходимость уточнения способов применения методов ООП к программным системам именно этого класса. В качестве цели применения ООП для разработки ПО СТЗ можно указать, в первую очередь, обеспечение быстрой разработки прототипов ПО СТЗ и выработку подходов к классификации понятий

предметной области ПО СТЗ, которые можно использовать для решения однотипных задач, например, задач слежения за движущимися объектами.

Применению ООП в задачах обработки зрительных данных посвящен ряд работ [12, 23, 25, 26, 44, 47]. Распространенным проектным решением является представление с помощью классов изображений, признаков, которые можно обнаружить на изображениях, а также алгоритмов обработки изображений для выделения признаков и для преобразования изображений. Представление алгоритмов обработки в виде иерархии классов позволяет унифицировать интерфейсы алгоритмов для объединения их в цепочки или древовидные структуры, в виде которых можно представить процесс программной обработки зрительных данных.

В работах [46, 47] описывается архитектура системы обработки изображений, реализованная в системе обработки изображений ANIMALS. Задача анализа изображений формулируется как задача построения «наилучшего» описания для входного изображения, соответствующего решаемой задаче. Построение описания выполняется с использованием алгоритмов, создающих т.н. объекты сегментации. Под объектами сегментации понимаются признаки, которые можно выделить на изображениях – отрезки, связные области и т.п. Полученное описание изображения в виде множества объектов сегментации сравнивается с моделями, построенными на основе априорных сведений (например, с использованием автоматической процедуры обучения по эталонным образцам). Сегментация в данной работе рассматривается как задача, решаемая на основе модели потоков данных. Классы используются для представления объектов сегментации. Применение объектов сегментации позволяет представить данные в виде, независимом от алгоритмов сегментации. Распознаваемые модели также предлагается представлять в форме множества объектов сегментации, связанных априорно известными структурными отношениями. Классы алгоритмов обработки изображений в системе ANIMALS образуют иерархию. Эти классы совместимы по параметрам и могут объединяться в цепочки.

В статье [25] рассматривается библиотека классов X Vision, представляющая собой платформно-независимую библиотеку классов для использования в зрительных системах слежения за движущимися объектами. В этой библиотеке содержатся классы для представления признаков и алгоритмов слежения за контурами и областями на изображениях. Основной подход к распознаванию изображений, принятый в системе X Vision, заключается в следующем: на изображении выделяются элементарные признаки, затем они объединяются в вектор состояния, а этот вектор используется для обнаружения моделей, построенных с учетом имеющихся в задаче ограничений. В [25] описывается опыт использования X Vision в нескольких модельных задачах: в системе «глаз-рука», в системе слежения за лицом оператора, в графической программе для рисования на основе распознавания жестов. Основными приемами при разработке приложений с использованием X Vision являются: деформация изображений для

преобразования признаков изображений к канонической ориентации, обработка малых фрагментов изображений, применение априорно известных ограничений и типизация признаков для использования предусмотренных классов признаков.

Библиотека TLIB [23] предназначена для использования в приложениях человеко-машинного интерфейса на основе обработки зрительных данных. Эта библиотека представляет собой набор оптимизированных функций на языке Си, для которых сформирован набор высокоуровневых оболочек в форме классов на языке Си++. Библиотека TLIB использовалась в ряде различных приложений из области человеко-машинного интерфейса, в т.ч. в интеллектуальных средах, бесконтактных медицинских интерфейсах, и в системах управления роботом на основе распознавания жестов. Все высокоуровневые методы TLIB были спроектированы с параметрами по умолчанию, оптимизированными для задач человеко-машинного интерфейса на основе обработки зрительных данных. На практике, это означает, что в большинстве случаев эти методы можно вызывать с несколькими параметрами (или вообще без параметров). В методах каждого класса предусмотрена проверка формата данных и геометрической целостности. Это позволяет использовать оптимизированные низкоуровневые функции, не содержащие средств безопасности (т.е. работающие в предположении корректности всех параметров). Архитектура и интерфейс TLIB обеспечивает простой, согласованный и хорошо документированный интерфейс программирования. Это сделано за счет ориентации библиотеки на использование в узкоспециализированном классе приложений.

В системе обработки изображений MAVIS объектно-ориентированный подход применяется для реализации системы с графическим языком представления алгоритмов обработки изображений в терминах потоков данных [44]. Алгоритмы представляются в виде объектов, которые принимают и генерируют различные скалярные и структурированные значения: числовые значения, строки, массивы и др. Все типы входных и выходных параметров алгоритмов являются специализацией базового класса «Сообщение». Таким образом в MAVIS обеспечивается расширяемость функций используемого графического языка за счет добавления новых алгоритмов обработки изображений.

Еще одним примером применения ООП для решения специализированной задачи обработки зрительных данных служит система для распознавания трехмерных объектов на изображениях, получаемых с помощью стереоскопической зрительной системы [12]. Процедура распознавания в этой системе генерируется на основе дерева интерпретации, которое классифицирует распознаваемый трехмерный объект в одну из групп по ориентации (в «класс видимости» [35, 57]). Каждый признак, выделяемый из изображения, или процедура сопоставления, представляется в виде отдельной сущности, т.е. объекта определенного класса. В системе распознавания используются объекты двух категорий – объекты данных и объекты-события. Объекты данных представляют геометрические объекты,

например, ребра и области. Объекты-события используются для сопоставления признаков и определения ориентации трехмерного объекта. Формирование дерева интерпретации изображений производится на этапе обучения системы. В статье [12] демонстрируется применение данного подхода для распознавания модельного трехмерного объекта.

Анализ опыта применения ООП для задач обработки зрительных данных показывает, что ряд решений является общепринятым: представление в виде классов изображений, признаков и алгоритмов обработки для выделения признаков. В значительной степени это связано с переносом в объектно-ориентированную среду подходов, выработанных при разработке автоматизированных систем обработки изображений, предназначенных для интерактивной обработки изображений. В качестве обоснования выбора ООП для разработки ПО СТЗ реального времени, в первую очередь, можно указать необходимость формирования совокупности повторно используемых компонентов ПО и каркаса ПО, который можно было бы со сравнительно небольшими (по сравнению с полной разработкой новой системы) трудозатратами специализировать для решения конкретных прикладных задач. Задача проектирования повторно используемых компонентов ПО СТЗ подразумевает необходимость как проектирования компонентов каркаса ПО, реализующих инфраструктуру приложения (интерфейс пользователя, интерфейс с устройствами ввода/вывода, взаимодействие с операционной системой и др.), так и компонентов, которые можно использовать для представления образов на изображениях. Решение этой задачи требует учета специфики СТЗ реального времени на всех этапах разработки ПО СТЗ, начиная с уточнения требований, предъявляемых к разрабатываемой программной системе, и заканчивая этапом опытной эксплуатации системы, на котором велика вероятность выявления новых требований к системе и требуются средства для оперативного установления причин ошибочных результатов автоматической обработки.

#### **4.2.1. Итеративные схемы применения объектно-ориентированного подхода**

При применении объектно-ориентированного подхода на начальных стадиях разработки ПО основное внимание уделяется вопросам объектно-ориентированного анализа и проектирования. Одним из ключевых моментов объектно-ориентированного анализа и проектирования является распределение обязанностей между программными компонентами. Этап анализа состоит в исследовании системных требований и решаемой задачи. В процессе проектирования основное внимание уделяется концептуальному решению, обеспечивающему выполнение основных требований, но не вопросам его реализации. Проектные решения впоследствии реализуются на выбранном языке программирования [82].

В процессе объектно-ориентированного проектирования определяются программные объекты и способы их взаимодействия с целью выполнения

системных требований. На этапе реализации выполняется реализация спроектированных компонентов на конкретном языке программирования. Для представления решений на этапе анализа и проектирования, а также при модификации этих решений в процессе разработки и сопровождения ПО могут применяться графические средства представления. Одним из таких средств является язык моделирования Unified Modeling Language – UML [71, 74]. С помощью языка UML можно представить схемы, поясняющие статические и динамические аспекты программной системы. Например, динамическое представление взаимосвязи объектов может представляться с помощью диаграмм последовательностей. На них отображаются потоки передачи сообщений между программными объектами и вызовы методов. На диаграммах последовательностей UML могут отмечаться предполагаемые временные затраты на обмен сообщениями [8], что существенно для систем реального времени. Статическое представление системы может быть построено в виде диаграммы классов, на которой показываются выделенные классы и отношения между ними. Язык UML разработан как общий язык моделирования, и, в частности, он применяется для представления моделей ПО систем реального времени [5, 6, 8, 28, 34, 56, 78, 38].

Для формализации применения ООП к разработке ПО были разработаны несколько различных технологических методов, предлагающих порядок выполнения этапов объектно-ориентированного анализа, проектирования и программирования. Среди таких методов можно назвать унифицированный процесс [30, 37], метод COMET [78] и метод XP (Extreme Programming) [70]. Общей чертой всех этих методов является использование приема итеративной разработки.

В рамках итеративных схем процесс разработки ПО выполняется в виде краткосрочных проектов фиксированной длительности, называемых итерациями [82]. Каждая итерация разработки включает свои собственные этапы анализа требований, проектирования, реализации, и завершается тестированием, интеграцией и созданием работающей системы. Результатом каждой итерации является не экспериментальный прототип, а окончательная версия некоторой части всей системы. На каждой итерации разработки рассматривается небольшое подмножество требований, для удовлетворения которых относительно быстро разрабатывается, реализуется и тестируется небольшая часть системы. То, что требования определяются не до реализации системы, а постепенно уточняются в процессе разработки, является отличительной чертой итеративных схем.

Унифицированный процесс [74, 82] представляет собой один из вариантов схем итеративной разработки ПО. В процессе разработки ПО выделяются четыре основные фазы: начальная фаза, фаза развития, фаза конструирования и фаза передачи в эксплуатацию. На каждой фазе выполняется несколько фиксированных итераций, количество которых зависит от конкретной задачи. На начальной фазе выполняется постановка задачи, формирование основных прецедентов, оценка сложности задачи. На фазе развития производится уточнение постановки задачи, итеративная

реализация базовой архитектуры, создание критичных компонентов, идентификация основных требований, уточнение оценок по срокам разработки. На фазе конструирования выполняется итеративная реализация менее критичных и простых компонентов, производится подготовка к развертыванию системы. На фазе передачи в эксплуатацию происходит опытная эксплуатация и развертывание системы.

Метод СОМЕТ [78] представляет собой метод проектирования параллельных приложений, в т.ч. распределенных и реального времени. Он основан на использовании концепции прецедентов. Каждый прецедент описывает последовательность взаимодействий между несколькими программными компонентами, вызванными некоторым иницирующим воздействием на систему. Для разрабатываемой программной системы в СОМЕТ выполняется несколько вариантов моделирования. На этапе моделирования требований разрабатывается модель, в которой определяются функциональные требования к системе в терминах прецедентов. Формируется текстовое описание каждого прецедента. При этом предполагается активное участие пользователей и экспертов. На этапе аналитического моделирования строятся статическая и динамическая модели системы. Статическая модель задает структурные отношения между классами предметной области. Классы и отношения между ними изображаются на диаграммах классов. На этапе проектного моделирования разрабатывается программная архитектура системы, при этом аналитическая модель отображается на эксплуатационную среду. После этапа проектирования выполняется этап инкрементного конструирования, затем этап инкрементной интеграции и комплексного тестирования ПО. В целом, метод СОМЕТ является специализацией унифицированного процесса для параллельных систем реального времени.

Метод ХР («экстремальное программирование») предполагает реализацию программной системы в течение последовательности коротких итераций. Для уточнения требований в коллектив разработчиков включается представитель заказчика (или постановщик задачи при разработке программной системы массового назначения). Требования к программной системе формулируются в виде набора прецедентов. По окончании каждой итерации проводится модульное тестирование, причем тесты формируются до написания текста программной системы. Разработка исходного текста ведется парами программистов. Как и в других итеративных схемах, подразумевается, что в процессе разработки система совершенствуется, каждый раз удовлетворяя поставленным на данной итерации требованиям.

Перечисленные итеративные методы направлены на организацию процесса анализа, проектирования и реализации ПО группой разработчиков. Эти технологические методы не предоставляют каких-либо средств для непосредственного упрощения реализации ПО СТЗ в плане разработки алгоритмов обработки зрительных данных и цикла реального времени. Во всех рассмотренных методах предполагается, что расширение системы при выполнении итераций разработки производится относительно некоторой базовой архитектуры. Построение этой базовой архитектуры является задачей,

решаемой специально для конкретного приложения, и не определяется каким-либо из технологических методов разработки ПО. Таким образом, применительно к разработке ПО СТЗ, формирование базовой архитектуры ПО представляется актуальной задачей, решение которой необходимо для успешного применения современных итеративных схем разработки ПО.

В качестве специфических особенностей ПО СТЗ, которые необходимо учитывать при применении итеративных схем, также можно отметить:

- изменчивость требований к ПО СТЗ и сложность их формализации;
- сложность планирования сроков разработки алгоритмов обработки зрительных данных (в связи с чем затруднительно применение схем с короткими итерациями наподобие XP);
- длительный срок опытной эксплуатации (например, в [7] приводится оценка, согласно которой 90% затрат на разработку промышленных зрительных систем приходится на настройку системы на этапе опытной эксплуатации).

Рассмотрим несколько примеров итераций в процессе разработки ПО СТЗ. Одним из таких примеров является разработка алгоритма для удаления меток, постоянно присутствующих на изображениях, обрабатываемых СТЗ для слежения за процессом сближения и стыковки космического корабля с Международной космической станцией (рис. 5). Эти метки выглядят на изображениях как горизонтальная и вертикальная штриховые линии, пересекающиеся вблизи центра изображения. Метки, образы которых видны на изображениях, нанесены на объективе камеры, установленной на космическом корабле. На первой итерации разработки алгоритма удаления меток был оформлен в виде отдельной операции, вызываемой в начале алгоритма обработки очередного изображения, полученного в ПО СТЗ. Удаление меток выполнялось в предположении, что их положение фиксировано. Координаты меток были определены посредством анализа тестовых видеосъемок. После выполнения первой итерации на этапе опытной эксплуатации обнаружилось, что положение меток не является постоянным и зависит от конкретного экземпляра камеры. На второй итерации разработки алгоритма удаления меток был модифицирован таким образом, чтобы местоположение меток могло быть задано оператором СТЗ перед включением автоматического режима обработки зрительных данных. По результатам эксплуатации этого варианта системы на очередной итерации разработки было принято решение доработать алгоритм удаления меток таким образом, чтобы в ПО СТЗ перед включением автоматического режима сначала выполнялась попытка автоматического обнаружения местоположения меток, затем результат поиска меток предъявлялся оператору, и оператор мог либо подтвердить результат автоматического обнаружения меток, либо скорректировать его вручную. Таким образом, модификация требований к ПО СТЗ с учетом результатов опытной эксплуатации привела к существенной модификации алгоритма обработки зрительных данных. В качестве проектного решения, обеспечившего возможность успешной модификации

ПО СТЗ, в данном случае можно указать выделение операции удаления меток в качестве отдельного этапа обработки зрительных данных.

Анализ применения итеративной разработки в жизненном цикле конкретных прикладных задач позволяет заключить, что итеративная разработка требует построения абстракций и программных компонентов, допускающих последующую модификацию в условиях прогнозируемых изменений. Поэтому успешный опыт решения однотипных задач, способы выделения классов в предметных областях этих задач и их оформление в виде компонентов-классов может упростить последующую реализацию прикладных СТЗ.

#### **4.2.2. Выделение классов в ПО СТЗ**

При использовании объектно-ориентированного подхода для разработки ПО СТЗ существенным вопросом является выделение классов для последующей реализации в ПО. Выделение классов при объектно-ориентированной разработке ПО отмечается в качестве важной проблемы [40, 73, 93], для решения которой в настоящее время пока не имеется удовлетворительных формальных методов. Во многих прикладных задачах процесс выделения классов носит субъективный характер, т.к. основывается на опыте разработчиков и требует учета компромиссов, связанных с используемыми программно-аппаратными средствами, ограниченными сроками разработки и др. Тем не менее, для обеспечения повторного использования и ускорения разработки ПО СТЗ, целесообразно отметить приемы выделения классов, которые были успешно применены при решении ряда конкретных прикладных задач обработки зрительных данных.

В монографии [73] отмечается, что сложность проблемы выделения классов при объектно-ориентированном проектировании объясняется сложностью более общей проблемы классификации, применяемой в качестве средства упорядочения знаний в различных областях человеческой деятельности. В объектно-ориентированном проектировании определение общих свойств объектов необходимо для обнаружения общих свойств важных для разрабатываемой системы понятий, представимых в виде классов, и механизмов взаимодействия объектов этих классов. Предполагается, что выделение классов с учетом особенностей конкретной задачи позволит упростить архитектуру программной системы. В процессе классификации, кроме выделения понятий, представляемых в виде классов, также производится определение отношений между этими классами (в частности, отношений обобщения, специализации и агрегации). Установление взаимодействия между объектами предметной области позволяет спроектировать взаимодействие между объектами ПО. Таким образом, можно заключить, что классификация при разработке ПО необходима для формирования модульной структуры ПО.

В [73] перечисляются три подхода к классификации понятий, которые находят применение при проектировании ПО: классическая категоризация, концептуальная кластеризация и метод прототипов.

При классической категоризации (классификация по свойствам) среди множества объектов в отдельные категории объединяются объекты, обладающие общими свойствами. На основе применения данного подхода в [40] предлагается выделить три общих категории классов ПО: классы аналитической модели, классы реализации и классы проектирования. Классы первой категории соответствуют понятиям предметной области задачи. Классы второй категории представляют вспомогательные понятия, необходимые для реализации алгоритмов, например, динамические структуры данных. К третьей категории относятся классы архитектуры ПО, например, классы, входящие в состав компонентного каркаса ПО СТЗ. Три общих категории классов допускают дальнейшее разделение, например, проектные классы компонентного каркаса ПО СТЗ могут быть разделены на классы, относящиеся к интерфейсу пользователя, к вводу зрительных данных, и к организации обработки зрительных данных в реальном времени. Приведенная классификация отражает модульное разбиение ПО СТЗ на параллельные подсистемы.

В подходе концептуальной кластеризации предусматривается возможность принадлежности объекта одновременно к нескольким категориям одновременно. При необходимости отнесения объекта к единственной категории применяется некоторый критерий правдоподобия. Примером использования данного подхода в ПО СТЗ является применение множественного наследования для формирования классов, представляющих операции обработки зрительных данных (подробнее рассматривается в п.8).

При использовании метода прототипов класс объектов задается путем указания одного объекта-прототипа. Новый объект можно отнести к данному классу при условии, что он характеризуется существенным сходством с прототипом в соответствии с заданным критерием сравнения. Данный подход классификации в ПО СТЗ в основном применяется на этапе формирования и анализа тестовых данных с целью выделения совокупностей объектов, которые необходимо обнаруживать в результате автоматической обработки зрительных данных.

Для применения перечисленных подходов к разработке ПО, в том числе и ПО СТЗ, целесообразно сформировать информационный материал, который можно рассматривать в качестве исходных данных для процедуры классификации. В нем должны быть представлены предполагаемые свойства и поведение разрабатываемой программной системы в виде, пригодном для выполнения объектно-ориентированного проектирования. Фактически, это означает необходимость спецификации требований к программной системе. В работах в области ООП [40, 73] демонстрируется тесная взаимосвязь между формулировкой требований и выделением классов. Эта связь лежит в основе итеративных схем разработки ПО. Применительно к формулировке требований в последние годы получила распространение концепция

прецедентов [14, 32, 41, 54, 73, 82]. В частности, прецеденты используются для представления требований в ряде известных итеративных технологических методов разработки ПО [30, 70, 78].

Прецедент (use case) представляет собой описание событий, вызванных некоторым инициирующим воздействием на систему. Это воздействие может поступить или из внешней среды (например, в виде команды от оператора), так и от некоторого внутрисистемного аппаратного или программного компонента (например, в системах реального времени инициирующим воздействием часто является сигнал от таймера [78]). Прецедент может быть представлен как в некотором формальном виде, так и в виде неформального текстового описания. Прецедент похож на описание предполагаемого сценария работы программной системы, за исключением того, что в некоторых технологических методах применения ООП в описание прецедента включается описание не одного сценария, а всех возможных или некоторой совокупности сценариев работы системы в ответ на заданное воздействие.

Применительно к ПО СТЗ можно выделить три специфические категории требований к системе: требования, связанные с методикой использования СТЗ; требования, связанные с представительными тестовыми данными, и требования, определяющие реакцию ПО СТЗ на результаты обработки зрительных данных.

Требования, связанные с методикой использования СТЗ, позволяют уточнить структуру интерфейса пользователя и допустимые значения свойств объектов, связанных с обработкой зрительных данных. В качестве примера рассмотрим СТЗ для слежения за движущейся малоразмерной целью. Эта СТЗ применяется для прослеживания в поле зрения неподвижной камеры образа лазерного целеуказателя, отмечающего местоположение апертуры бесконтактного измерителя. В качестве бесконтактного измерительного прибора может использоваться оптический пирометр, перемещаемый человеком вручную, или триангуляционная дальнометрическая система в состав которой входит источник структурированного освещения. В обоих вариантах применения СТЗ описание методики использования содержит спецификацию допустимых способов перемещения устройства, на котором закреплен лазерный целеуказатель. На основе этой спецификации в ПО СТЗ учитываются ограничения на скорость перемещения образа прослеживаемой цели, ограничения на форму наблюдаемой траектории и на размеры образа цели. Требования данной категории позволяют уточнить признаки корректности объектов классов предметной области, например, указать свойства образов цели и фрагментов траектории, соответствующие корректным объектам.

Требования, связанные с представительными тестовыми данными, позволяют указать образцы зрительных данных, в результате обработки которых в ПО СТЗ должны обнаруживаться объекты интереса или, наоборот, образцы данных, в результате обработки которых не должно происходить ложных срабатываний алгоритмов распознавания. Данные требования формулируются на первых итерациях разработки ПО, и обычно

модифицируются и дополняются на последующих итерациях и в процессе опытной эксплуатации [7]. Верификация требований данной категории является одним из критериев корректности разработки ПО СТЗ.

Третья категория требований, специфичных для ПО СТЗ, содержит требования, определяющие порядок действий СТЗ в процессе обработки зрительных данных в результате обнаружения или диагностирования отсутствия объектов интереса. Например, в СТЗ для слежения за процессом сближения и стыковки космического корабля с Международной космической станцией выполняется обнаружение образа станции, имеющего на изображениях в зависимости от расстояния различную структуру (рис. 6). Среди допустимых значений дальности были выделены четыре диапазона расстояний, на которых образ станции может быть описан с использованием одной и той же структурной модели. Для каждого диапазона расстояний в ПО СТЗ реализован отдельный алгоритм обработки зрительных данных. В зависимости от результатов обработки очередного изображения, в ПО СТЗ может быть принято решение о принадлежности изображения множеству, соответствующему одному из допустимых диапазонов дальности. В таком случае на последующих кадрах сразу может быть применен соответствующий алгоритм обработки, без перебора всех возможных алгоритмов. Таким образом, требования данной категории позволяют уточнить структуру цикла реального времени и способы взаимодействия объектов ПО.

Описание прецедентов является распространенным методом представления требований к разрабатываемой программной системе. Важное значение прецеденты имеют как инструмент верификации ПО, т.к. позволяют проконтролировать работоспособность системы согласно сформулированным на этапе проектирования требованиям. При использовании прецедентов для выделения классов у прецедентов отмечается ряд характерных недостатков [40]. В описании прецедента обычно отражается порядок выполнения системой действий, и описание в таком виде более близко соответствует структурному подходу к декомпозиции программной системы на алгоритмы и действия, а не на классы. Также описания прецедентов, особенно в неформализованном текстовом виде, обычно являются субъективными, например, содержат описание действий системы с точки зрения предполагаемого пользователя. В целом, применительно к использованию прецедентов для выделения классов в [40] отмечается необходимость контроля со стороны разработчика, чтобы следование описаниям прецедентов не приводило к переходу от объектно-ориентированного к структурному подходу к разработке ПО. Применительно к ПО СТЗ, как к системам реального времени, можно отметить возможность совместного использования обоих подходов при разработке различных подсистем ПО. В СТЗ реального времени необходимо обеспечить реакцию системы на возникающие внешние события с гарантированным временем отклика. Соответствие заданным временным ограничениям учитывается в структуре цикла реального времени, в реализации которого взаимодействуют различные объекты ПО СТЗ. Выделение классов этих объектов выполняется с

использованием объектно-ориентированного подхода, а построение структуры цикла реального времени – в основном с использованием структурного подхода, при котором детально уточняются порядок выполняемых в цикле действий с контролем имеющихся в задаче временных ограничений.

После формирования требований к системе эти требования могут использоваться для выделения классов с использованием различных методов [40, 69, 73, 82], среди которых особенно часто применяются следующие:

- Синтаксический метод. При этом в документальном описании требований к программной системе выполняется поиск существительных и глаголов. Существительные являются кандидатами на оформление в виде классов, глаголы – функций-членов. Для успешного применения данного метода необходимо построение требований в виде, отражающем понятия, которые могут быть представлены в виде классов. Проблематичным является применение этого метода для выделения классов, не соответствующих понятиям предметной области, например, к классам проектирования, представляющим объекты программной архитектуры.
- Использование предыдущих версий программных систем. Данный метод предполагает применение имеющегося опыта разработки однотипного ПО (например, с применением шаблонов проектирования), а также адаптацию готовых классов и библиотек (например, посредством создания классов-оболочек).

Процесс выделения классов, как и процесс разработки ПО, может выполняться итеративным образом. Итеративное выделение классов важно для построения отношения наследования. В работах [40, 73] отмечается, что отношение наследование обычно используется на ранних стадиях разработки в случаях, если структурные отношения общности/специализации можно явно идентифицировать в предметной области. В иных случаях наследование уточняется на последующих итерациях разработки ПО.

В [40] в процессе выделения классов отмечается двойственность, заключающаяся в том, что при разработке ПО с одной стороны, важно установление понятий, которые можно представить в виде классов, а с другой, выявление понятий, не являющихся классами. Таким образом, применительно к разработке ПО желательно указать критерии, согласно которым понятие может быть или представлено в виде класса или это нецелесообразно.

В [40] отмечаются следующие распространенные признаки, которые часто указывают на «неверное» определение класса (хотя эти признаки не носят категоричного характера):

- Оформление в виде класса компонента, который можно представить в виде процедуры. Косвенным признаком данного дефекта является построение класса, предоставляющего всего одну функцию-член в общедоступном интерфейсе. С точки зрения рассмотрения классов как абстрактных типов данных, класс должен соответствовать некоторой значимой в данной задаче абстракции данных. Возможным исключением являются классы,

представляющие абстрактные действия, в частности, алгоритмы обработки зрительных данных в ПО СТЗ.

- Классы, не содержащие методов, но содержащие данные. Это аналоги структурных типов данных в процедурных языках программирования. Распространенные варианты применения классов такого типа: представление внешних для системы сущностей, например, данных от датчиков; инкапсуляция констант для использования в иерархиях наследования, представление параметров обработки функциями-членами других классов.
- Нарушение согласованности интерфейса класса: все свойства класса должны соответствовать одной абстракции, идентифицированной применительно к разработке конкретной программной системы.

Критерием, специфическим для ПО СТЗ, является выделение классов на основе анализа структуры обрабатываемых изображений. Если на изображении удастся указать специфический образ, выделяемый в результате обработки, то соответствующее понятие может быть рассмотрено в качестве кандидата на представление в виде класса.

### **4.3. Шаблоны проектирования ПО**

Шаблоны проектирования представляют собой способ использования при разработке ПО архитектурных решений, выработанных в процессе разработки других программных систем. Таким образом, шаблоны проектирования можно рассматривать в качестве механизма документирования и повторного использования архитектурных принципов разработки ПО.

Значение проектирования программной архитектуры рассматривается в монографии [58]. Отмечается, что при увеличении размеров и сложности программных систем, проектирование и спецификация общей структуры системы становится даже более значимым вопросом, чем выбор алгоритмов и структур данных для организации вычислений. Структурные вопросы включают в себя организацию системы как совокупности компонентов; глобальные управляющие структуры; протоколы связи, синхронизации и доступа к данным; назначение функциональных обязанностей проектным элементам; композицию проектных элементов; физическое распределение; вопросы масштабирования и производительности; прогнозирование развития программной системы; выбор среди альтернатив проектирования. Все перечисленные вопросы в [58] отнесены к уровню проектирования программной архитектуры.

Под шаблоном проектирования (design pattern) будем понимать описание взаимодействия объектов и классов, адаптированных для решения типичной задачи проектирования в конкретном контексте [40, 77, 82]. Шаблон проектирования именуется, абстрагирует и идентифицирует ключевые аспекты структуры общего решения, которые позволяют применить его для создания повторно используемой структуры программной системы. Он вычленяет

взаимодействующие классы и объекты, их роли, отношения и функциональные обязанности. При описании каждого шаблона внимания акцентируется на конкретной задаче ООП. В общем случае шаблон проектирования состоит из четырех основных элементов: имя, решаемая задача, способ решения и получаемые результаты.

В качестве примера шаблона проектирования можно указать шаблон Модель/Вид/Контроллер (Model/View/Controller, MVC) [49, 77]. Этот шаблон описывает типичное архитектурное решение в области построения пользовательского интерфейса. В нем предполагается разделение классов, относящихся к модели данных, к визуальному представлению этих данных, и к обработке команд от пользователя. Классы последней категории обеспечивают синхронизацию классов, относящихся к модели данных и их визуальному представлению. Этот шаблон часто используется при организации интерфейса пользователя, в т.ч. в ПО СТЗ.

В [77] предлагается деление шаблонов проектирования на три категории: шаблоны создания объектов, структурные шаблоны и шаблоны поведения. Например, к шаблонам поведения относится «Слабо связанный обмен сообщениями» (асинхронный обмен сообщениями) [78]. Этот шаблон применяется в параллельных системах, в т.ч. в системах реального времени. Он может реализовываться на основе очереди сообщений. В данном шаблоне предполагается, что при обмене сообщениями отправитель сообщения не дожидается, пока посланное сообщение будет обработано получателем. К шаблонам поведения также относится «Цепочка ответственности» [74, 77]. Этот шаблон представляет типичный способ организации обработчиков событий. Реализации шаблонов проектирования на конкретном языке программирования (Си++) рассматриваются в [68, 85].

Кроме шаблонов проектирования, продемонстрировавших работоспособность при решении прикладных задач, можно выделить шаблоны, приводящие к проблемам при разработке, модификации и сопровождении программных систем. Эти шаблоны рассматриваются, например, в [11, 96]. Среди них можно назвать: реализация полиморфизма в объектно-ориентированных системах с применением оператора множественного выбора, дублирование кода, повторные вычисления производных величин, неиспользуемый код, встраивание обращения к аппаратуре на уровень приложения и др.

Отдельные шаблоны проектирования представляют типовые решения типичных проблем в заданном контексте [74]. В ООП также выделяется понятие каркаса (framework). Каркас – это, как правило, архитектурный шаблон, предполагающий расширение для разработки приложений в конкретной области применения [74]. Каркас может включать в себя некоторый набор шаблонов проектирования.

Примером каркаса программной системы является трехзвенная архитектура информационных систем, в которой обязанности разделяются между уровнем пользовательского интерфейса, хранением информации и объектами и правилами предметной области. Этот каркас используется для

построения распределенных информационных систем. Типичный каркас в системах реального времени – «циклический исполнитель». В этом каркасе предполагается деление времени на кадры и подкадры, в которых обработка протекает в строгих временных рамках. Выбор такого каркаса вместо архитектуры, управляемой событиями, влияет на всю программную систему [74].

В ряде работ рассматривается вопрос выделения шаблонов проектирования для решения частных задач разработки систем реального времени. Например, в [50] описывается устройство гибкого производственного робототехнического модуля, предназначенного для сборочного производства. В составе этого модуля применяется зрительная система. Она выполняет определение местоположения собираемых деталей и позволяет избежать применения специального подающего оборудования для механического позиционирования деталей. Необходимость быстрой перенастройки модуля на сборку изделий из деталей новых типов требует перенастраиваемой зрительной системы, которая сможет распознавать новые объекты. Повторно используемыми программными компонентами в этой системе являются процедуры выделения образов на бинарных изображениях. При разработке ПО этого робототехнического модуля были выделены совокупности компонент, предназначенные для применения в гибких производственных модулях в качестве шаблонов проектирования ПО. Выделены шаблоны для решения таких подзадач, как общее управление системой, коммуникации между системными компонентами, манипуляции с деталями и сборками, планирование системных задач, обработка ошибок.

В [9] рассматривается вопрос применения шаблонов в области проектирования человеко-машинного интерфейса. Отмечается проблема выбора способа представления шаблонов. Для этого предлагается несколько возможных способов: текстовые описания, диаграммы классов и объектов на каком-либо графическом языке, а также специализированные языки для представления шаблонов проектирования.

Применительно к разработке каркасов систем реального времени, в т.ч. систем обработки зрительных данных, опубликован ряд работ [3, 4, 18-20, 22, 24, 26, 41, 51, 55, 65, 66]. Предлагаемые каркасы приложений обычно направлены на решение задач узкоспециализированного типа, например, для построения распределенных систем [4, 51, 65], систем навигации мобильных роботов [3], промышленных систем управления [26, 50, 55], виртуальных сред [18-20].

В [24] описывается каркас обработки зрительных данных для построения программных систем, предназначенных для обнаружения семантически значимых событий в рамках решения задачи сегментации видеопоследовательностей [57]. Примеры таких событий: автомобильная авария, поворот человека в процессе ходьбы, погоня животных и т.п. Для обнаружения заданного события предлагается выявлять специфические пространственные, временные и пространственно-временные образы, которые можно обнаружить в процессе автоматической обработки. Каркас построен на

основе использования промежуточного представления видеопоследовательностей. Формат данных в этом представлении рассчитан на построение детекторов заданных событий. В описанном каркасе предполагается описание пространственно-временных изменений образов объектов с использованием разнотипных признаков изображений и видеопоследовательностей, в т.ч. цветовых и текстурных признаков, пространственно-временных образов, признаков движения, и признаков смены съемочного плана. В этом каркасе применяются следующие шаблоны проектирования: многоуровневая архитектура, процедуры выделения образов объектов в пространственной, временной и пространственно-временной области.

В [66] описывается каркас системы для автоматического сопоставления полицейского фоторобота с набором фотографий в базе данных. На сравниваемых изображениях и на фотороботе обнаруживаются характерные признаки лица, затем выполняется нормировка по геометрическому расположению и по яркости. Затем выполняется сравнение нормированных изображений.

Статья [22] посвящена специализированной программно-аппаратной архитектуре для слежения за движущимися объектами в реальном времени с использованием цветовой сегментации. Большинство существующих алгоритмов слежения за объектами выделяют признаки, которые составляют образ объекта и прослеживают эти признаки от одного кадра к другому. В данной статье предлагается программно-аппаратная архитектура для решения задачи слежения на уровне образа объекта. Первый этап заключается в выделении объекта на изображении сцены. При этом используются цветовые признаки объекта. На втором этапе вычисляется центр тяжести объекта. На последнем этапе выполняется слежение за центром тяжести на последовательности изображений. Прослеживание на кадровой частоте видеосигнала обеспечивается за счет применения специализированного видеопроцессора.

В [51] представлена архитектура распределенной зрительной системы с несколькими камерами. Синхронизация ввода и обработки достигается за счет использования шаблона «Программный триггер». Три камеры установлены на трех разных ПК под управлением ОС Linux и работают в качестве клиентов, посылающих изображения для обработки на центральный сервер через сеть Ethernet. Сервер управляет и координирует клиентские машины. В протоколе связи сервера с клиентскими ПК принято, что сначала сервер соединяется со всеми клиентами. Затем сервер одновременно выдает всем клиентским ПК сигнал на съем изображений. Клиенты выполняют ввод изображений размерами  $320 \times 240$  пикселей и отправляют их серверу (возможно, после первичной обработки). Сервер выполняет совместную обработку и отображение полученных изображений. Для сопоставления изображений по времени предусматривается синхронизация часов на всех клиентских ПК с часами сервера. «Одновременность» съема зависит от нескольких факторов: задержки сети, задержки диспетчеризации ОС, особенностей камер.

Синхронизация ввода обеспечивается в пределах одного кадра. Предлагаемый шаблон может применяться для некритичных по времени обработки приложений, например, в некоторых приложениях виртуальной реальности.

В [41] описывается каркас построения процедуры распознавания в системах обработки зрительных данных, основанных на проверке гипотез относительно наличия и структуры образов предполагаемых объектов. Однопроходный вычислительный каркас последовательной проверки гипотез расширяется за счет использования стратегий обратной связи, в которых применяются методы генерации дополнительных гипотез и их верификации. В качестве базового архитектурного шаблона в процедуре распознавания объектов рассматривается шаблон «гипотеза-верификация». Предлагаются способы использования обратной связи в алгоритмах распознавания, которые могут применяться в системах реального времени для сокращения времени вычислений за счет: повышения производительности нижних уровней обработки, на которых выделяются характерные признаки; генерации гипотез для сосредоточения внимания на локальных областях интереса; генерации гипотез для обнаружения дополнительной информации об объектах интереса. Описываемый каркас иллюстрируется на примере обработки зашумленных инфракрасных изображений для обнаружения образов мостов на естественных сценах, в т.ч. при частичном перекрытии этих образов другими объектами.

В работе [3] описывается архитектура зрительной системы для мобильных роботов. В этой архитектуре выделены 5 уровней: низкоуровневая обработка, обработка в областях интереса, обработка отдельных признаков, выделение пространственных отношений между признаками и образами объектов, и распознавание местоположения/видимого объекта. На каждом уровне обработки описание изображения расширяется и видоизменяется по сравнению с предыдущим. В отличие от иерархических многоуровневых архитектур, в этой архитектуре информация может передаваться в обоих направлениях.

Анализ перечисленных работ демонстрирует возможность представления архитектурных решений (фактически, возможность выделения классов проектирования [40]), в форме шаблонов проектирования, и целесообразность формирования каркаса приложения, ориентированного на разработку программных систем специализированного типа. В системах реального времени известен набор распространенных шаблонов – циклический исполнитель, многоуровневая архитектура, последовательная проверка гипотез и применение обратной связи в процедурах распознавания, и др.

По результатам разработки ПО нескольких прикладных СТЗ реального времени был выделен набор шаблонов проектирования ПО СТЗ. В качестве кандидатов на оформление в виде шаблонов проектирования рассматривались компоненты ПО СТЗ, применявшиеся в нескольких различных СТЗ. Это компоненты, связанные с вводом зрительных данных, с применением алгоритмов обработки для решения низкоуровневых задач обработки

изображений (например, маркировка, выделение контуров, бинаризация и др.), и компоненты для организации интерфейса пользователя СТЗ.

Выделенные шаблоны проектирования можно разделить на категории, связанные с построением архитектуры ПО СТЗ, с обработкой зрительных данных и с отладкой и тестированием ПО СТЗ. Перечень шаблонов проектирования для использования в ПО СТЗ реального времени с пояснением назначения этих шаблонов приведен в п. 4.3.1. Реализация набора этих шаблонов, обеспечивающих решение типовых операций СТЗ реального времени, выполнена в виде повторно используемого программного каркаса (рис. 4.5, рис. 4.6).

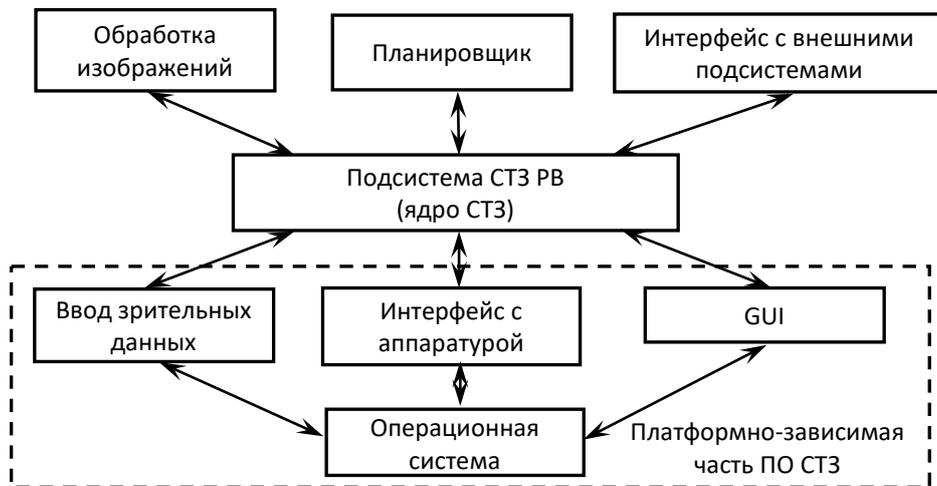


Рис. 4.5. Общая структура каркаса ПО СТЗ реального времени.

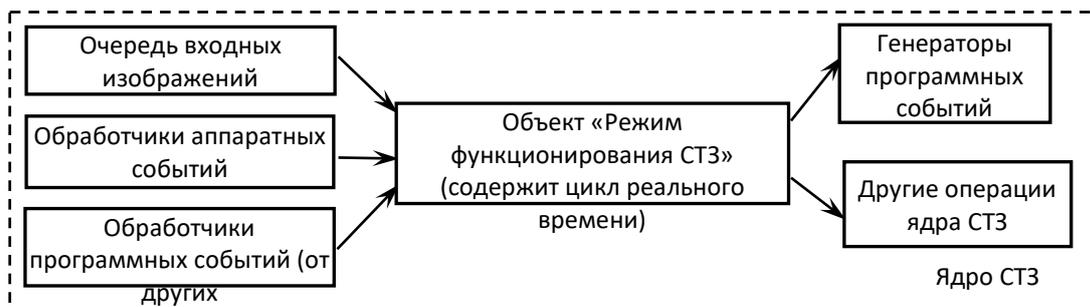


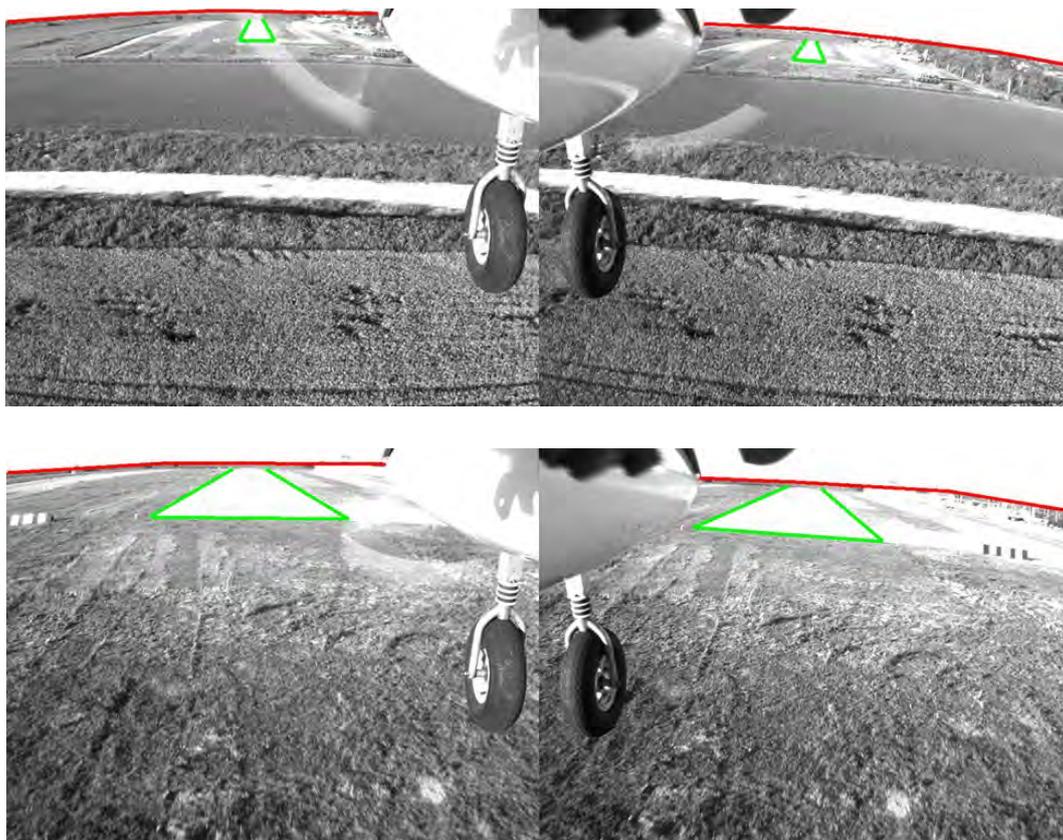
Рис. 4.6. Структура ядра каркаса ПО СТЗ реального времени.

#### 4.3.1. Шаблоны проектирования архитектурного уровня

1) **Монокулярная СТЗ.** Этот шаблон описывает взаимодействие объектов «Источник зрительных данных», «Алгоритм обработки изображений» и «Элемент пользовательского интерфейса «Поле зрения» для обеспечения ввода и обработки зрительных данных в двух режимах работы СТЗ: в режиме настройки и в автоматическом режиме. Шаблон «Монокулярная СТЗ» лежит в основе компонентного каркаса ПО СТЗ. В СТЗ с несколькими

полями зрения содержится соответствующее количество экземпляров реализации данного шаблона (рис. 4.7).

- 2) **Поле зрения.** Данный шаблон описывает взаимодействие объектов пользовательского интерфейса, обеспечивающих визуальное представление и операции с одним полем зрения в режиме настройки СТЗ. Эти операции включают, главным образом, манипуляции с элементами управления в области изображения поля зрения и просмотр содержимого этих областей, при необходимости, с применением различных масок.



*Рис. 4.7.* Примеры изображений, обрабатываемые в стереоскопической СТЗ на борту ЛА для обнаружения наземных объектов интереса.

- 3) **Применение пулов памяти для уменьшения количества операций динамического выделения и фрагментации памяти.** В связи с многократным повторением цикла обработки реального времени актуальна проблема сокращения количества операций динамического выделения памяти. Для буферов изображений проблема может быть решена путем однократного выделения буфера при его первом использовании с последующим контролем, что выделенный буфер достаточен для загрузки обрабатываемого изображения. Применение пулов памяти целесообразно для работы с объектами, количество которых ограничено, но непостоянно на различных итерациях цикла реального времени.
- 4) **Защита общих данных объектами синхронизации многопоточного доступа операционной системы и вычисление времени, проводимого в ожидании захвата этих объектов.** Этот шаблон, как и предыдущий, имеет

общий характер по отношению к проектированию параллельных систем реального времени. В СТЗ защищаемыми данными являются объекты, служащие для связи основных системных компонентов: компонентов ввода зрительных данных, их обработки и интерфейса пользователя. Например, это очередь кадров, заполняемая компонентом ввода изображений; буфер текущего кадра в интерфейсном компоненте. Применительно к разработке СТЗ выполнение оценок временных затрат на ожидание объектов синхронизации позволяет обнаружить дефекты в проектировании многопоточного взаимодействия в ПО СТЗ.

#### **4.3.2. Шаблоны проектирования для обработки данных в реальном времени**

- 1) **Временной прогноз.** Данный шаблон предполагает проектирование объектов, выполняющих предсказание местоположения образов распознаваемых объектов на основе ранее полученных данных, возможно, с учетом априорной модели движения объектов.
- 2) **Отложенное конструирование.** Этот шаблон описывает схему построения программных структур, представляющих распознаваемые объекты. Эти объекты должны содержать необходимое количество свойств, которые могут потребоваться для успешного распознавания. Полное заполнение всех свойств корректными значениями производится только в случае успешного распознавания и распределено во времени.
- 3) **Верификация масштаба реального времени.** Во многих СТЗ обработка зрительных данных должна быть синхронизирована с внешними событиями. В подобных случаях применим шаблон «Верификация масштаба реального времени», предназначенный для фиксации средствами СТЗ важных для системы событий, возможно, посредством специальной обработки изображений.
- 4) **Множественное наследование «алгоритм обработки»+«направление поиска».** В реализации алгоритмов обработки изображений выделение элементарных признаков часто производится путем перебора и анализа фрагментов изображения по определенным траекториям. Данный шаблон предполагает распределение действий по выделению признаков и по просмотру изображения по заданной траектории по отдельным классам, пригодным для комбинирования путем множественного наследования. Например, при выделении на изображении низкоуровневых признаков в виде краев может потребоваться применение краевого оператора с локальной апертурой вдоль некоторой траектории изображения, вдоль которой вероятно обнаружение искомого края на основе априорных данных или предыдущих результатов обработки. Для реализации такой операции требуется обеспечение функциональной обработки двух видов: просмотр данных изображения вдоль траектории поиска признака и применение краевого оператора, выделяющего край. Эта функциональность может быть совмещена для формирования класса,

выполняющего поиск краевого признака вдоль требуемой траектории, посредством множественного наследования.

- 5) **Обратная связь между уровнями ввода и обработки зрительных данных.** В структуре операций по вводу и обработке зрительных данных можно выделить несколько уровней: ввод зрительных данных, выделение элементарных признаков изображения, обнаружение образов объектов интереса и определение искомым параметров этих объектов. Данный шаблон может использоваться в ПО СТЗ для повышения надежности обработки зрительных данных в ПО СТЗ посредством связи операций различных уровней. Например, после успешного обнаружения образа искомого объекта может быть выполнена настройка параметров яркости и контраста источника зрительных данных с целью повышения качества цифровых зрительных данных для автоматической обработки.
- 6) **Планирование низкоуровневых операций обработки зрительных данных.** Этот шаблон предназначен для обеспечения функционирования ПО СТЗ в реальном времени. Длительность выполнения отдельных операций обработки зрительных данных (например, трассировка контуров на бинарном изображении) зависит от содержания обрабатываемого изображения. При необходимости обработки каждого поступающего изображения в ПО СТЗ могут использоваться временные ограничения на выполнение отдельных операций, организованные так, чтобы слишком долго обрабатываемое изображение могло быть пропущено до завершения всех предусмотренных операций обработки (такие изображения рассматриваются в качестве особых случаев, например, это могут быть зашумленные изображения). Дополнительно может быть предусмотрено сохранение исключенных из обработки изображений для последующего анализа разработчиком ПО СТЗ. Для организации учета временных ограничений в ПО СТЗ в рамках данного шаблона предлагается предусматривать возможность явного указания предельной длительности выполнения операции, которая должна явно проверяться в алгоритме, реализующем данную операцию. В случае превышения заданного лимита времени алгоритм должен выполнять возврат с передачей информации о возникновении данной особой ситуации.
- 7) **Верификация гипотез распознавания.** При обработке зрительных данных в ПО СТЗ требуется принимать решения относительно наличия на изображении образов объектов интереса или их частей. Эти решения могут приводить к формированию результатов обработки отдельных изображений (например, объект найден/не найден) или к оценке состояния наблюдаемых объектов, влияющей на выбор высокоуровневого алгоритма обработки изображений в структуре цикла реального времени СТЗ. В процессе принятия решений может использоваться шаблон «Верификация гипотез», который предполагает выполнение дополнительных операций, подтверждающих корректность распознавания объекта.

## **4.4. Инструментальные средства**

### **4.4.1. Базовый язык программирования**

Основная задача, решаемая в ПО СТЗ, заключается в обработке зрительных данных в режиме реального времени. Для сокращения сроков разработки и упрощения решения однотипных задач обработки зрительных данных при разработке ПО СТЗ предполагается применение повторно используемых компонентов. По этим причинам при выборе языка программирования для реализации ПО СТЗ учитывались следующие критерии:

- возможность разработки на данном языке программирования ПО для функционирования в режиме реального времени;
- возможности реализации алгоритмов обработки зрительных данных;
- способы оформления компонентов ПО для повторного использования;
- доступность на ПК общего назначения;
- наличие интегрированной среды программирования.

Критерий возможности разработки ПО для работы в режиме реального времени позволяет сформулировать ряд требований к языку программирования, среди которых в [97] были выделены требования надежности, удобочитаемости, гибкости, простоты, переносимости и эффективности. В качестве основного средства для обеспечения надежности языка в [97] отмечается поддержка типов данных и строгой типизации. Требование гибкости языка подразумевает наличие в языке различных языковых возможностей для выражения необходимых в программе операций. Например, применительно к разработке ПО систем реального времени это средства для взаимодействия с нестандартными устройствами ввода/вывода. ПО многих систем реального времени, в т.ч. СТЗ, часто выполняет параллельную обработку, в связи с этим требуется, чтобы язык программирования обеспечивал средства для разработки параллельных программ. К ним относятся средства описания параллельных процессов, их синхронизации и обмена данными между процессами [87].

В качестве распространенных языков для разработки ПО систем реального времени и встроенных систем выделяются языки Си, Си++ и Ада [16, 87, 88]. Системы реального времени рассматривались в качестве одной из основных целевых областей применения при проектировании языка Ада [88]. На ПК общего назначения не распространены средства разработки ПО на этом языке. Языки Си и Си++ не были предназначены непосредственно для программирования в реальном времени, но могут быть расширены для этой цели путем использования в среде соответствующих операционных систем и применения дополнительных библиотек [87]. Переносимость Си и Си++ упрощается благодаря наличию стандартов на эти языки программирования [2, 29]. Наличие стандарта языка, доступность средств разработки на ПК общего назначения, и поддержка средств объектно-ориентированного программирования в сочетании со строгой типизацией позволили выбрать

Си++ [93] в качестве базового языка для реализации ПО СТЗ. Эффективность Си++ в области доступа к памяти и к аппаратным средствам унаследована от его предшественника, языка Си. По применению языка Си для реализации алгоритмов обработки изображений и зрительных данных опубликовано большое количество работ, в т.ч. [23, 47, 83, 98].

Применительно к задаче реализации алгоритмов обработки зрительных данных в качестве достоинств языка Си++ можно выделить низкоуровневые средства, связанные с организацией доступа к памяти и с арифметическими действиями с указателями, а также высокоуровневые средства абстракции данных на основе механизма классов с применением строгой типизации. В комбинации эти средства позволяют представить на языке программирования высокоуровневые абстракции – классы, в функциях-членах которых используются низкоуровневые средства для эффективного выполнения операций обработки зрительных данных, например, попиксельной обработки данных изображений. Механизм строгой типизации в Си++ позволяет указывать функциям-членам классов атрибуты const для спецификации функций-членов, в результате вызова которых не меняются переменные-члены класса, т.е. не меняется состояние объекта класса. Использование атрибутов const для функций-членов классов, их параметров и возвращаемых значений обеспечивается применение строгой типизации для обнаружения на этапе компиляции ошибок проектирования, связанных с ошибочным выполнением функций-членов, изменяющих состояние константных объектов.

Операции над отдельными пикселями составляют значительную часть операций обработки в ПО СТЗ. Поэтому возможность уменьшения количества машинных инструкций при обработке отдельного пикселя является одним из способов повышения эффективности реализации алгоритмов обработки. Типичный пример применения данного подхода заключается в доступе к пикселям, последовательно хранящимся в памяти, путем увеличения значения переменной-указателя. В частности, данный прием в силу своей эффективности применяется даже в среде виртуальной машины .NET [89] посредством перехода к «небезопасному» коду (в котором возможен прямой доступ к памяти в среде виртуальной машины).

В области применения Си++ для представления и обработки изображений также можно отметить следующие частные вопросы, касающиеся использования конкретных механизмов языка:

- применение параметризованных классов для представления изображений;
- перегрузка операторов для оформления операций с данными изображений;
- применение встраиваемых функций для обеспечения удобочитаемости исходного текста наряду с эффективной реализацией алгоритмов обработки изображений.

Параметризованные классы [69, 73, 93] при проектировании ПО позволяют выразить свойства протоколов классов, связанные с

необходимостью сопоставления операций класса с жестко не специфицированным набором классов, которые могут применяться в качестве типов данных параметров функций-членов, возвращаемых значений и переменных параметризованного класса. Обычный (непараметризованный) класс экспортирует операции, которые можно выполнять над его объектами. У параметризованного класса настраиваемый параметр служит для импорта классов и значений, предоставляющих некоторый протокол. Компилятор Си++ проверяет их взаимное соответствие при компиляции, когда фактически и происходит создание конкретного варианта параметризованного класса. С помощью механизма параметризации оказалось возможным построение единого класса «Изображение», который может быть специализирован для работы с бинарными, полутоновыми, цветными изображениями, а также с изображениями с целочисленными и вещественными пиксельными данными (подобные изображения применяются для представления результатов обработки изображений, например, для представления спектра мощности изображения).

Перегрузка операторов в языке Си++ предназначена для переопределения действия операторов языка применительно к описанным в программе классам. В ряде случаев этот механизм позволяет повысить удобочитаемость программ, когда выполняемые переопределенным оператором действия соответствуют его общепринятой семантике. Примером подобного применения перегруженных операторов является доступ к данным пикселей в классе «Изображение». Переопределение для этой цели оператора вызова функции с двумя параметрами (координатами пиксела) обеспечивает запись, напоминающую обращение к двумерному массиву и упрощающую исходный текст алгоритмов обработки.

Для оформления компонентов повторного использования в ПО СТЗ были выбраны классы и их совокупности, образующие реализации шаблонов проектирования и компонентного каркаса. Язык Си++ отмечается в [88] в качестве языка, упрощающего взаимодействие с библиотеками, разработанными сторонними производителями. Причиной этого служит широкое распространение языка в области разработки системного и прикладного ПО, в частности, операционных систем. Это упрощает доступ к сервисам базовой операционной системы. Среди них можно указать технологии для оформления и использования компонентов ПО, реализуемых операционной системой, а не непосредственно языком программирования (например, технологии СОМ для оформления двоичных повторно используемых компонентов в среде ОС Windows). Недостатком применения такого подхода является снижение переносимости ПО, поэтому в рассматриваемых СТЗ в качестве повторно используемых компонентов рассматривались модули, поддерживаемые языком программирования, т.е., классы в Си++, а в качестве предполагаемого пользователя этих компонентов – разработчик ПО СТЗ.

Вопрос поддержки компонентного программирования на уровне языка программирования в настоящее время остается предметом исследований [21,

62]. Например, в качестве средства для поддержки компонентного программирования на уровне языка в [21] предлагается использовать обмен сообщениями, для оформления которого применяются специальные синтаксические конструкции языка. Этот подход используется в экспериментальном языке программирования Lagoon. Отличительными чертами императивного объектно-ориентированного языка Lagoon являются трактовка сообщений и отсутствие наследования. Модульная программная единица в этом языке – модуль, а не класс. Для реализации полиморфизма используются сообщения, которые, в отличие от интерфейсов в Си++ или Java [76], не привязаны статически к собственной реализации. В объектах объявляются методы, «реализующие сообщения», т.е., методы ответственные за обработку этих сообщений. Необрабатываемые сообщения передаются в метод «по умолчанию». Данный подход может быть реализован средствами Си++ в виде слабосвязанного обмена сообщениями [78], когда сообщение (возможно, представленное в виде объекта специального класса) передается в предусмотренный для этой цели функцию-член диспетчеризации сообщений. Эта функция-член выполняет обработку сообщения или передает его в обработчик «по умолчанию». Применительно к ПО СТЗ этот подход обладает тем недостатком, что соответствие сообщений и обрабатываемых их объектов не может быть проверено на этапе компиляции. Поэтому в ПО СТЗ для передачи сообщений используется явный вызов функций-членов классов, реализующих обработку сообщений.

При разработке ПО систем реального времени необходимо обеспечить выполнимость временных ограничений [21, 87, 97]. Спецификация временных ограничений на уровне отдельных компонентов ПО представляется проблематичным, т.к. временные ограничения в ПО систем реального времени обычно являются свойством системы в целом, а не отдельных компонентов. В ПО СТЗ этот вопрос решается посредством учета временных ограничений при проектировании цикла реального времени и путем явного контроля времени выполнения операций обработки зрительных данных, длительность которых может зависеть от содержания обрабатываемого изображения (п. 8.2, шаблон проектирования «Планирование низкоуровневых операций обработки зрительных данных»).

Использование классов в качестве повторно используемых компонентов ПО не является универсальным решением. Ряд недостатков применения классов в качестве базовых программных единиц с точки зрения расширяемости ПО рассматривается в [79]. Например, таким недостатком является отсутствие у классов статических горизонтальных связей. Применительно к задаче разработке ПО СТЗ не ставилось цели разработки универсального способа представления программных компонентов. Основная цель применения классов заключается в формировании повторно используемых компонентов, которые можно использовать в составе каркаса ПО СТЗ реального времени для упрощения решения однотипных задач обработки зрительных данных. При этом объектно-ориентированное программирование является средством для проектирования рациональной

структуры разрабатываемой программной системы [79], в которой исключаются регулярные перестановки модулей с одного места на другое. Хотя обеспечение расширяемости и многократной используемости ПО не обеспечивается автоматически благодаря применению объектно-ориентированного проектирования и языка Си++, формирование набора классов для дополнения компонентного каркаса ПО СТЗ позволяет упростить разработку ПО посредством доработки незначительного количества классов ПО СТЗ, представляющих прогнозируемые точки роста ПО СТЗ.

#### **4.4.2. Базовая операционная система**

В ПО СТЗ в реальном времени могут параллельно выполняться несколько задач: ввод зрительных данных в память компьютера, обработка зрительных данных, управление нестандартными устройствами ввода/вывода и представление информации о состоянии СТЗ и результатов обработки оператору. На однопроцессорных системах многозадачность можно рассматривать как способ структуризации ПО, выполняющего параллельные действия, т.к. фактически процессорное время делится на кванты, и каждый квант в соответствии с логикой используемого планировщика времени предоставляется только одной задаче.

Последовательное планирование выполнения задач сложно предусмотреть в структуре ПО СТЗ. Сложности связаны с тем, что операции обработки отдельных порций зрительных данных (кадров) могут занимать неодинаковые промежутки времени. Обработка некоторого кадра может занять больше времени, чем доступно до получения очередной порции данных. Например, это может быть обработка при первоначальном обнаружении движущегося объекта без априорной информации о его местоположении. Тем не менее, пропуск кадров может быть нежелателен, если в случае успешного обнаружения объекта обработка оставшихся кадров может быть выполнена за меньшее время, чем длительность одного кадра. Поэтому параллельно с обработкой зрительных данных в ПО СТЗ требуется реагировать на очередные поступающие кадры, например, чтобы буферизовать их для последующей обработки. Кроме того, параллельно с действиями по вводу и обработке зрительных данных может потребоваться произвести операции по взаимодействию с устройствами ввода/вывода, причем входные сигналы от этих устройств часто нельзя предсказать заранее, что является характерной особенностью систем реального времени. Организация подобных неоднородных по времени и длительности выполнения программных действий существенно проще производится на уровне программирования параллельных задач, чем в виде последовательной программы, т.к. последний вариант фактически означает необходимость организации некоторого планировщика в структуре ПО СТЗ. Поэтому разработка ПО СТЗ в рамках параллельной архитектуры упрощает проектирование ПО, а также возможный перенос в многозадачную среду на вычислительную систему с несколькими процессорами.

В выбранном языке программирования Си++ параллельное программирование непосредственно не поддерживается, поэтому поддержка параллельности необходима на уровне операционной системы (ОС), с возможностью доступа к этим средствам из программ на Си++.

Кроме требования многозадачности, к ОС для ПО СТЗ можно сформулировать следующие требования:

- доступность на ПК общего назначения, выбранных в качестве базовых для ПО СТЗ;
- наличие программных и аппаратных компонентов для ввода зрительных данных;
- наличие таймеров и часов реального времени;
- возможность написания драйверов нестандартных устройств ввода/вывода;
- обеспечение обработки прерываний с предсказуемой малой задержкой (в сравнении с масштабом реального времени ПО СТЗ);
- наличие интегрированной среды разработки на языке Си++;
- возможность интеграции ПО СТЗ с готовым прикладным ПО для представления результатов обработки ПО СТЗ;
- многоплатформенность ОС для упрощения переноса ПО СТЗ на встроенные и специализированные вычислительные системы.

Выбор ОС, удовлетворяющей всем предъявляемым требованиям, проблематичен. По этой причине возможно выбрать ОС, удовлетворяющую некоторым из этих требований, и спроектировать ПО СТЗ таким образом, чтобы при необходимости ПО СТЗ могло быть перенесено в среду другой ОС с изменениями, не затрагивающими, по крайней мере, подсистему обработки зрительных данных.

Для описания средств многозадачности в современных ОС часто применяются стандарты POSIX (стандарты переносимого интерфейса ОС). Эти стандарты были разработаны Институтом инженеров по электронике и радиоэлектронике (IEEE), и приняты в качестве международных стандартов Международной организацией по стандартизации ISO [92]. Стандарт POSIX основан на ОС UNIX. В терминологии POSIX «тяжеловесные» процессы, владеющие адресным пространством, называются просто «процессами», а облегченные, функционирующие в среде процесса, называются «потокками» [94]. Потокки могут пользоваться общим адресным пространством памяти, принадлежащим их процессу. Каркас ПО СТЗ был спроектирован в расчете на использование потоков. При этом основные действия ПО СТЗ оформляются в виде квазипараллельных (на однопроцессорном ПК) потоков.

В организации многозадачности для систем реального времени ключевым требованием к ОС является требование предсказуемости поведения ОС. Предсказуемость поведения ОС (при выполнении контекстного переключения, синхронизации задач и обработке прерываний) означает, что максимальное время отклика должно быть прогнозируемо при любой ожидаемой нагрузке на систему [78].

Для ПК имеется большое количество различных ОС, среди которых наиболее распространенными являются ОС семейства Windows [90, 91] и семейства UNIX [92, 94]. Эти ОС изначально не разрабатывались для использования в системах реального времени, поэтому их применение для СТЗ с жесткими временными ограничениями проблематично. Существенным препятствием для этого является отсутствие предсказуемости поведения этих ОС при планировании потоков и обработке прерываний. На ПК также доступны несколько ОС реального времени, например, QNX [36, 81], VxWorks [15], Phar Lap ETS Realtime [16].

ОС Unix первоначально была разработана как многопользовательская, а не для приложений реального времени [87]. Подпрограммы ОС работают как пользовательские процессы, но с наивысшим приоритетом, назначенным системой. Поэтому невозможно прерывать те системные вызовы, выполнение которых занимает много времени. Это увеличивает время реакции системы, что является существенным недостатком для задач реального времени, особенно, управляемых прерываниями. В ОС используется динамическая настройка приоритетов. Системные обращения синхронизированы с вызывающим процессом. В настоящее время в UNIX поддерживаются такие функциональные элементы систем реального времени, как семафоры, разделяемая память, обмен сигналами между процессами, приоритетное управление задачами и прямой доступ к внешним устройствам (программный доступ к этим элементам определяется стандартами POSIX).

Как и в UNIX, в ОС семейства Windows NT ядро отделено от системных программ, которые выполняются как прикладные процессы. Машинно-зависимая, аппаратно-ориентированная часть Windows NT называется уровнем аппаратной абстракции (Hardware Abstraction Layer – HAL). Эта часть реализована для нескольких типов аппаратных платформ. В Windows NT поддерживаются потоки, события, семафоры и асинхронный вызов процедур. Непосредственный доступ к аппаратуре из приложений не поддерживается, для этого должны оформляться драйверы устройств [45, 87].

ОС семейства Windows NT не удовлетворяют требованиям, предъявляемым к системам реального времени [52, 63, 64], но с помощью дополнительных подсистем, например, Venturcom RTX [13], эта ОС может быть доведена до уровня ОС реального времени. Расширение Windows NT для использования в системах жесткого реального времени путем применения подсистемы Venturcom RTX описывается в [13]. Эта подсистема заменяет ядро Windows для выполнения предсказуемой диспетчеризации потоков реального времени и межпроцессного взаимодействия между средой реального времени и обычной средой Windows. Подсистема RTX разработана так, чтобы в Windows обеспечивались следующие свойства ОС реального времени:

- многопоточная вытесняющая многозадачность с большим количеством доступных приоритетов;
- предсказуемые механизмы синхронизации потоков;
- система наследования приоритетов;
- таймеры и часы высокого разрешения.

RTX включает в себя набор библиотек (статических и динамических), подсистему реального времени RTSS, оформленную в виде драйвера режима ядра Windows, и расширенную подсистему HAL (для обработки прерываний в реальном времени, реализации высокочастотных таймеров и часов, для диагностирования аварийного останова ОС). RTSS реализует объекты синхронизации реального времени и предсказуемую диспетчеризацию. Библиотеки предоставляют программисту программный интерфейс для вызова функций RTX из прикладных программ.

Приводимые в [13] сведения свидетельствуют о сокращении наихудшего времени выполнения системных операций на 3 порядка (таких, как захват и освобождение объектов синхронизации, смена приоритета потока, задержка диспетчеризации обработчика прерываний). Максимальное время в наихудшем случае для эти операций примерно одинаково и уменьшается до десятков микросекунд (по сравнению с 50 миллисекундами для Windows NT без подсистемы реального времени).

Подобные программные системы, адаптирующие многозадачную ОС общего назначения для использования в системах реального времени, существуют и для ОС Linux [61]. Например, применение расширения RT-Linux рассматривается в [10] для управления двигателями устройства позиционирования камеры с двумя степенями свободы, а в [59] – для ПО управления двигателями мобильного робота. ОС Linux, как и UNIX, непосредственно не предназначалась для систем реального времени. Ядро Linux запрещает прерывания и система не сможет среагировать сразу на поступившее прерывание. Поэтому для применения в жестких системах реального времени требуется модификация ОС Linux. У расширений Linux RT-Linux и RTAI-Linux похожая архитектура, основанная на использовании небольшого ядра, которое обеспечивает функции времени и обработчики прерываний для систем реального времени. Планировщик работает по прерываниям таймера. Задачи реального времени работают в пространстве ядра. С обычным Linux эти подсистемы разделяют одну или более задач. Это позволяет выполнять задачи реального времени наряду с задачами в среде обычной ОС Linux. В [59] продемонстрирована возможность выполнения двух периодических задач в среде Linux с расширением RT-Linux с периодом выполнения 100 и 150 мкс при критической задержке 10 мкс. Эти задачи, оформленные в виде двух потоков, пользуются общим ресурсом, защищенным мутексом.

В [59] рассматривается проблема выбора ОС для разработки систем реального времени на основе метода оценки производительности с учетом трех метрик: пропускной способности, реактивности, и предсказуемости. Пропускная способность – это количество операций, которые может выполнять система в течение заданного промежутка времени. Реактивность характеризует скорость реакции системы на некоторое событие. Предсказуемость описывает, насколько возможно предсказать порядок действий системы в ответ на происходящие события. Для оценки этих метрик предлагается использовать тестовые программы, выполняющие измерение

временных задержек при выполнении определенных системных вызовов, например, операций с семафорами для синхронизации процессов. Пропускную способность для этой операции можно оценить как количество операций с семафорами, которые может выполнить система в течение одной секунды. Реактивность оценивается как промежуток времени между переключением семафора одним процессом и реакции на это другого процесса. Предсказуемость измеряется как дисперсия времени отклика. Для жестких систем реального времени предсказуемость является наиболее важным фактором. Описанная методика позволяет оценить параметры выполнения отдельных операций, но затруднительно распространить полученные оценки для описания функционирования системы реального времени в целом.

Результаты подобных экспериментов по оценке задержек планировщика Windows NT представлены в работе [13]. Отмечается наличие в этой ОС задержек диспетчеризации процессов в десятки миллисекунд. Этот факт делает невозможным применение Windows NT для систем реального времени с временными ограничениями менее десятков миллисекунд. В Windows NT не все процессорное время контролируется планировщиком – он не контролирует время обработки прерываний. ОС Windows NT разработана в расчете на минимизацию латентности прерываний (латентность прерываний – это задержка между аппаратным прерыванием и моментом начала программной обработки этого прерывания.) за счет разделения обработчиков на две части – собственно обработчик прерывания и процедуру отложенной обработки прерываний. Хотя введение процедур отложенной обработки снижает латентность прерываний, выполнение операций с устройствами ввода-вывода (особенно у драйверов видеоадаптеров и сетевых плат) в этих процедурах по результатам [13] выделяется в качестве возможных причин задержек диспетчеризации потоков приложения.

Таким образом, можно заключить, что без применения пакетов расширения, ОС семейства Windows NT могут быть использованы для систем реального времени со слабыми временными ограничениями и с масштабом реального времени не менее 10 мс. Предельная величина масштаба реального времени объясняется периодом срабатывания таймера в подсистеме аппаратной абстракции HAL (в зависимости от платформы, это 10-15 мс [13]).

Применительно к вопросу компонентной разработки ПО СТЗ также может быть рассмотрена возможность использования ПО промежуточного уровня для интеграции ПО СТЗ с прикладным ПО для представления и дальнейшей обработки результатов ПО СТЗ.

ПО промежуточного уровня предназначено для обеспечения взаимодействия двоичных компонентов ПО [17, 39]. В настоящее время распространены три семейства технологий данного класса: JavaBeans/EJB [86], COM/DCOM/COM+/.NET [72, 89], и CORBA/CCM. Эти технологии предлагают различные сервисы времени выполнения для управления активацией компонентов, их взаимодействием и параллельным выполнением, долговременным хранением параметров, распределением и

транзакциями. Такие компонентные каркасы выполняют многие функции, свойственные обычным операционным системам [39]. Они не ориентированы на применение в системах реального времени. Основное назначение этих технологий заключается в построении распределенных и масштабируемых информационных систем. В этом плане они могут использоваться для обеспечения долговременного хранения результатов, получаемых ПО СТЗ, и для интеграции ПО СТЗ с готовым прикладным ПО для передачи и обработки результатов. Например, в СТЗ для слежения за движущейся малоразмерной целью и в СТЗ для определения местоположения контактного провода железной в процессе движения электровоза для формирования отчетных документов использовалось представление данных в виде электронных таблиц для пакета Microsoft Excel. Формирование табличных документов с выходными данными выполнялось после завершения сеанса работы в реальном времени с использованием средств взаимодействия приложений в среде Windows на основе технологии COM.

По результатам анализа возможностей применения ОС для ПК общего назначения в качестве базовой ОС были выбраны ОС семейства Windows NT (Windows 8/10). В качестве основных причин выбора этих ОС можно назвать:

- обеспечение многозадачности с диспетчеризацией потоков и поддержкой параллельности в ядре ОС;
- наличие подсистемы ввода зрительных данных DirectShow, для которой доступно большое количество устройств ввода, в т.ч. передающих зрительные данные в цифровом виде через интерфейс IEEE-1394;
- наличие библиотек для обработки зрительных данных для макетирования алгоритмов обработки зрительных данных (п. 3);
- возможность использования технологии COM для интеграции с прикладными программами для представления и обработки результатов, полученных в ПО СТЗ, а также для их долговременного хранения.

ОС Windows NT использовалась для ПО СТЗ с масштабом реального времени порядка 10 мс (в этих СТЗ выполняется обработка зрительных данных, поступающих от источников телевизионного стандарта или цифровых источников с кадровой частотой 25-30 Гц). При необходимости обеспечения более высокой реактивности ПО СТЗ возможно применение подсистемы реального времени для ОС Windows NT или перенос ПО СТЗ в среду другой ОС.

#### **4.4.3. Использование открытого ПО Robot Operating System**

В последнее десятилетие популярным ПО промежуточного уровня для задач робототехники стала ОС ROS (Robot Operating System) [43]. ROS – это «мета-операционная» система (meta-operating system) с открытыми исходными текстами [53]. Она обеспечивает сервисы, характерные для операционной системы, в т.ч. уровень абстракции аппаратного обеспечения,

низкоуровневое управление устройствами, реализация типичной функциональности, передача сообщений между процессами, управление пакетами. В ROS также есть инструментальные средства и библиотеки для получения, сборки, написания и запуска ПО в многокомпьютерной среде.

Основная цель ROS заключается в обеспечении повторного использования ПО в области исследований и разработки роботов. ROS представляет собой распределенный каркас в виде набора процессов (в терминологии ROS называются узлами). Исполняемые процессы могут разрабатываться по отдельности. Во время выполнения взаимодействие между ними реализуется на основе слабо связанного обмена сообщениями.

Эти процессы можно группировать в пакеты, которые легко распространять и совместно использовать. В ROS есть интегрированная система хранилищ исходных текстов, которая позволяет реализовать распределенную разработку ПО. Такая структура ROS, охватывающая уровни от файловой системы до уровня сообщества разработчиков, позволяет применять различные независимые решения относительно деталей проектирования и реализации, но все полученные результаты можно собрать вместе в рамках инфраструктуры ROS с помощью ее инструментальных средств.

#### **4.5. Выводы**

ИСУ с использованием СТЗ представляют собой системы реального времени, которые наряду с общими для систем реального времени свойствами параллельности и необходимости удовлетворения временных ограничений, характеризуются специфическим свойством – информация о внешних событиях в ПО поступает не в непосредственно пригодном для машинного использования виде, а формируется в результате программной обработки зрительных данных, представленных в форме цифровых изображений.

Типичными примерами СТЗ реального времени, которые использовались в качестве основы для формирования методов программирования ПО СТЗ, являются СТЗ для мониторинга состояния механических систем, инспекционные технологические СТЗ и СТЗ для слежения за подвижными объектами. Специфической частью каждой из реализованных СТЗ является структура цикла реального времени и алгоритм обработки зрительных данных.

Для разработки ПО ИСУ с целью обеспечения дальнейшего сопровождения, модификации и повторного использования ПО предлагается применять объектно-ориентированный подход, в последние годы ставший наиболее распространенным методом разработки ПО с расчетом на повторное использование компонентов ПО.

Применение объектно-ориентированного подхода для разработки ПО ИСУ подразумевает разработку в рамках итеративного подхода. Каждая итерация разработки сопровождается построением окончательной версии некоторого варианта системы. Спецификой ПО ИСУ является то, что переход

к новой итерации разработки происходит не только в связи с уточнением и расширением сформулированных человеком требований, но и в связи с обнаружением новых типов входных данных, не обрабатываемых на текущей итерации (например, изображений). Для документирования проектных решений предлагается использовать подмножество языка моделирования UML: диаграммы классов, состояний и последовательностей.

Критичным моментом при разработке ПО ИСУ является выделение классов ПО. Выделение классов выполняется с учетом требований к ПО, которые могут быть заданы в форме прецедентов. Спецификой ПО ИСУ с использованием СТЗ является дополнение прецедентов, связанных с обработкой зрительных данных, представительными тестовыми данными. Прецеденты этой категории используются при верификации ПО для установления корректности его функционирования согласно сформулированным требованиям. Выделяемые в ПО классы относятся к трем основным категориям: классы аналитической модели (классы предметной области), классы реализации и классы проектирования. Классы проектирования образуют классы, входящие в состав компонентный каркас ПО. В зависимости от решаемой задачи, перечисленные категории допускают дальнейшее разделение.

Одним из механизмов документирования и повторного использования архитектурных архитектурных принципов разработки ПО являются шаблоны проектирования. Шаблон проектирования представляет собой описания взаимодействия объектов и классов, адаптированных для решения типичной задачи проектирования в конкретном контексте. Применительно к ПО СТЗ выделен ряд шаблонов проектирования: «Монокулярная СТЗ», «Поле зрения», «Временной прогноз», «Отложенное конструирование» и др.

В качестве языка программирования для реализации ПО ИСУ был выбран язык Си++. Этот выбор был сделан с учетом наличия стандарта языка, доступностью средств разработки на ПК общего назначения, и поддержки средств объектно-ориентированного программирования в сочетании со строгой типизацией.

Каркас приложения – это крупномасштабный архитектурный шаблон, предполагающий расширение для разработки приложений в конкретной прикладной области. Этот каркас предназначен для сокращения сроков разработки ПО. В каркасе СТЗ для использования в составе ИСУ предусмотрены подсистемы ввода зрительных данных, интерфейс с оператором, обработка зрительных данных. Эти подсистемы должны функционировать параллельно, и они оформляются в виде отдельных потоков. Компонентный каркас ПО СТЗ состоит из классов, относящихся к категории классов проектирования, и представляет собой реализацию базовой архитектуры, наличие которой упрощает применение итеративных схем разработки ПО.

Компоненты ПО ИСУ с использованием СТЗ допускают интеграцию с ПО в среде системы ROS. Интеграция возможна посредством реализации

компонентов ПО ИСУ в формате узлов ROS, а также на основе обмена данными с узлами ROS посредством сетевых соединений.

### **Литература**

- 1) AdOculus 4.0, DBS GmbH, Germany. 1998. <http://www.dbs-imaging.com>
- 2) ISO International Organization for Standardization. Information technology – Programming languages – C. ISO/IEC 9899:2018, <https://www.iso.org/standard/74528.html>, 2018.
- 3) Balkenius C., Kopp L. The XT-1 Vision Architecture. Proc. of the Symposium on Image Analysis, Lund University, Sweden, 1996. pp. 39-43.
- 4) Barendt N.A., Merat F.L., Podgursky A., Blanchard E. A Distributed, Object-Oriented Software Architecture for Platform-Independent Machine Vision. IASTED Int. Conf. on Robotics and Manufacturing, Banff, Canada, 1998.
- 5) Barros E.A., Franceschetti N.N., Ferreirazy G. Development of self-propelled high speed ship models. Proc. of Mechatronics 2002, University of Twente, 24-26 June 2002, pg. 50-61.
- 6) Bastidas G., Miyagi P.E. Okamoto J. Modelling integration of systems in intelligent buildings through mechatronics approach. Proc. of Mechatronics 2002, University of Twente, 24-26 June 2002, pg. 202-211.
- 7) Batchelor B.G., Whelan P.F. Machine vision systems: Proverbs, principles, prejudices and priorities. Proc. of the SPIE – The International Society for Optical Engineering, Vol. 2347 – Machine Vision Applications, Architectures, and Systems Integration III, Boston, USA, pp. 374-383.
- 8) Bauckhage C., Fritsch J., Rohlfing K.J., Wachsmuth S., Sagerer G. Evaluating Integrated Speech- and Image Understanding. In Proc. IEEE Int. Conf. on Multimodal Interfaces (ICMI'02), pages 9-14, Pittsburgh, PA, 2002.
- 9) Borchers J.O., Thomas J.C. Patterns: What's In It For HCI? Extended abstracts of the CHI 2001 Conference on Human Factors in Computing Systems, Seattle, March 31–April 5, New York: ACM Press, 2001.
- 10) Broenink J.F., Jovanovic D.S., Hilderink G.H. Controlling a Mechatronic Set-Up Using Real-Time Linux and CTC++. Proc. of Mechatronics 2002, University of Twente, 24-26 June 2002, pg. 1323-1331.
- 11) Brown W.J., Malveau R.C., McCormick H.W., Mowbray T. Antipatterns. Refactoring Software, Architectures, and Projects in Crisis. John Wiley & Sons, NY, 1998. 336 pg.
- 12) Chang H., Ikeuchi K., Kanade T. Model-Based Vision System by Object-Oriented Programming. Tech. report CMU-RI-TR-88-03, Robotics Institute, Carnegie Mellon University, February, 1988.
- 13) Cherepov M., Hirst M., Jones C., Zimmerman M. Hard Real-Time with Venturcom RTX on Microsoft Windows XP and Windows XP Embedded, 2002, <http://windowsfordevices.com/articles/AT3618198049.html>.
- 14) Cockburn A. Writing Effective Use Cases. Addison-Wesley, 2000. 304 pg.
- 15) Collier J. An overview tutorial of the VxWorks Real-Time Operating System, 2004, <http://www.cross-comp.com/pages/embedded/index.php>.

- 16) Cyliax I. Software Development for RTOSs. Circuit Cellar INK, the Computer Applications Journal. №93, April 1998, pp.53-59.
- 17) Economopoulos A., Martakos D. Component-based architectures for computer vision systems. In Proc. Of The 9th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision'2001 (WSCG'2001), University of West Bohemia, Czech Republic, February 5-9, 2001.
- 18) François A. Components for Immersion. Proceedings of the IEEE Int. Conf. on Multimedia and Expo, Lausanne, Switzerland, August 2002.
- 19) François A. Software Architecture for Computer Vision: Beyond Pipes and Filters. IRIS Technical Report IRIS-03-420, University of Southern California, Los Angeles, USA, July 2003.
- 20) François A., Medioni G., A Modular Middleware Flow Scheduling Framework. Proc. ACM Multimedia 2000, Los Angeles, USA, Nov. 2000. pp. 371-374.
- 21) Franz M., Fröhlich P.H., Kistler T. In Proc. Of The 5th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'99F, IEEE Press, Monterey, CA, USA, 18-20 November 1999. pp. 125.
- 22) García-Campos R., Battle J., Bischoff R. Architecture of an Object-based Tracking System Using Color Segmentation. In B. G. Mertzios and P. Liatsis (Eds.): 3rd International Workshop on Image and Signal Processing. Manchester, UK, November 1996, Proc. IWISP'96, pp. 299-302.
- 23) Grange S, Fong T.W., Baur C. TLIB: a real-time computer vision library for HCI. Digital Image Computing – Techniques and Applications Conference, IAPR/IEEE, Sydney, Australia, December, 2003, pp. 1017-1026.
- 24) Haering N., Lobo N. A Framework for Designing Event Detectors. Technical Report CS-TR-99-01, Univ. of Central Florida, Orlando, USA, January 1999.
- 25) Hager G.D., Toyama K. X Vision: A Portable Substrate for Real-Time Vision Applications. Computer Vision and Image Understanding. 1998, Vol. 69, No. 1, pp. 23-37.
- 26) Horst J.A. Architecture, Design Methodology, and Component-Based Tools for a Real-Time Inspection System. In Proc. of the 3rd IEEE International Symposium on Object-oriented Real-time Distributed Computing (ISORC 2000), Newport Beach, California, USA, March 15-17, 2000, pp. 296-304.
- 27) Horst J.A. Architecture, Design Methodology, and Component-Based Tools for a Real-Time Inspection System. Third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. Newport Beach, California, USA, March 15-17, 2000. p. 296
- 28) Huang H.M., Messina E., Scott H., Albus J. et al. Open System Architecture for Real-time Control Using a UML Based Approach. In Proc. of the 1st ICSE Workshop on Describing Software Architecture with UML, Toronto, Canada, May 15, 2001.
- 29) ISO International Organization for Standardization. Information technology – Programming languages – C++. ISO/IEC 14882:2017, <https://www.iso.org/standard/68564.html>, 2017.

- 30) Jacobson I., Booch G., Rumbaugh J. The Unified Software Development Process. MA.: Addison-Wesley, 1999.
- 31) Jacobson I., Christerson M., Jonsson P., Övergaard G., Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, Massachusetts, 1992.
- 32) Jacobson I., Ericsson M., Jacobson A., The Object Advantage: Business Process Reengineering With Object Technology, Addison-Wesley, 1995.
- 33) Jones M.B., Regehr J. The Problems You're Having May Not Be the Problems You Think You're Having: Results from a Latency Study of Windows NT. In Proceedings of the 1999 Workshop on Hot Topics in Operating Systems (HotOS VII), Rio Rico, AZ, March 29-30, 1999, pp. 96-102.
- 34) Kersten M., Matthes J., Manga C.F., Zipser S., Keller H.B. Customizing UML for the development of distributed reactive systems and code generation to Ada 95. Ada User Journal, vol. 23, No. 3, September 2002.
- 35) Koenderink J.J., van Doorn A.J. The internal representation of solid shape with respect to vision. Biological Cybernetics, v. 32, 1979, pp. 211–216.
- 36) Kolnick F. The QNX 4 Real-time Operating System. Canada, Basis Computer Systems Inc., 1998. 936 pg.
- 37) Kruchten P. The Rational Unified Process – An Introduction, 2nd edition. MA.: Addison-Wesley, 1999.
- 38) Lange F., Wunsch P., Hirzinger G. Predictive Vision Based Control of High Speed Industrial Robot Paths. // Proc. of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998, p. 2646-2651.
- 39) Makowski P., Ravn A.P. Component Based Development – Where is the Place for Formalization? UNU/IIST Tech. Rep. № 284, UNU Int. Institute for Software Technology, Macau, September 2003, pp. 159-166.
- 40) Meyer B. Object-Oriented Software Construction, Second Edition. Prentice Hall, 1997. 1254 pg.
- 41) Mirmehdi M., Palmer P.L., Kittler J., Dabis H. Complex Feedback Strategies for Hypothesis Generation and Verification. Proceedings of the 7th British Machine Vision Conference, University of Edinburgh, UK, 1996.
- 42) Müller H., Gülch E., Mayr W. A new modelling technique for object-oriented photogrammetric computer vision algorithms. ISPRS 2002 Symposia, September 9-13, 2002, Graz, Austria. p. B-186 ff (4 pg).
- 43) Newman W. A Systematic Approach to Learning Robot Programming with ROS. CRC Press, 2018.
- 44) Olson T.J., Klop N.G., Hyett M.R., Carnell S.M. MAVIS: A Visual Environment for Active Computer Vision. Proc. of the 1992 IEEE Workshop on Visual Languages, September 15-18, 1992, Washington, USA, pp. 170-176.
- 45) Oney W. Programming the Microsoft Windows Driver Model, Second Edition. Microsoft Press, 1999. 800 pg.
- 46) Paulus D., Ahlrichs U., Heigl B., Denzler J., Hornegger J., Zobel M., Niemann H. Active Knowledge-Based Scene Analysis. Videre: Journal of Computer Vision Research, MIT Press, Winter 2000, vol. 1, No. 4.

- 47) Paulus D., Niemann H. Object-Oriented Programming for Image Analysis. In Current Topics of Pattern Recognition Research (editor J. Menon), volume 1 of Research Trends, India, 1996, pp. 185-204.
- 48) Pisoft Image Framework 8.0, Институт информационных технологий, Лаборатория компьютерного зрения, Москва, 2000.  
<http://www.iitvision.ru/pisoft.htm>
- 49) Prorise J. Programming Windows with MFC, 2nd ed. – Redmond, Washington, USA: Microsoft Press, 1999. – 1376 p.
- 50) Quinn R.D., Causey G.C., Merat F.L., Sargent D.M., Barendt N.A. et al. Design of an Agile Manufacturing Workcell for Light Mechanical Applications. Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, USA, April 1996. pp. 858-863.
- 51) Rai P.K., Tiwari K., Guha P., Mukerjee A. A Cost-effective Multiple Camera Vision System using FireWire Cameras and Software Synchronization. 10th International Conference on High Performance Computing (HiPC 2003), Hyderabad, India, December 17-20, 2003.
- 52) Ramamritham K., Shen C., Gonzalez O., Sen S., Shirgurkar S.B. Using Windows NT for Real-Time Applications: Experimental Observations and Recommendations. In Proc. 4<sup>th</sup> IEEE Real-Time Technology and Applications Symposium, (Denver, CO), June 3-5, Denver, Colorado, 1998. pp. 102-111.
- 53) ROS Introduction. URL: <https://wiki.ros.org/ROS/Introduction>
- 54) Rubin D. Uses of Use Cases. <http://www.softstar-inc.com/Download/Uses of Use Case.pdf>
- 55) Schneider S.A., Chen V.W., Pardo-Castellote G. The ControlShell Component-Based Real-Time Programming System. Proc. IEEE Intern. Conf. on Robotics and Automation, Nagoya, Japan, May 21-27, 1995. pp. 2381-2388.
- 56) Selic B., Gullekson G., Ward P. Real-Time Object-Oriented Modeling. John Wiley & Sons, NY, 1994. 525 pg.
- 57) Shapiro L.G., Stockman G.C. Computer Vision. Prentice Hall, 2001, 580 pg.
- 58) Shaw M., Garlan D. Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, NJ, USA, 1996.
- 59) Silly-Chetto M., Garcia T., Lucas G., Orhant Y. On using Real-Time Linux in a mobile robot application. // Proc. 7th Intern. Conf on Systemics, Cybernetics and Informatics (SCI 2003), Orlando, Florida, USA, July 27-30, 2003.
- 60) Stein J. Object-Oriented Programming and Database Design. Dr. Dobb's Journal of Software Tools for the Professional Programmer, March 1988, No. 137, p.18.
- 61) Straumann T. Open Source Real Time Operating Systems Overview. // Proc. 8th International Conference on Accelerator & Large Experimental Physics Control Systems, San Jose, California, USA, 2001, p. 235-237.
- 62) Szyperski C. Component Software: Beyond Object-Oriented Programming. ACM Press/Addison-Wesley Publishing Co., New York, USA, 1998. 411 pg.
- 63) Timmerman M., Beneden B.V., Uhres L. Windows NT Real-Time Extensions: Better or Worse? Real-Time Magazine, № 3, 1998, pp. 11-19.

- 64) Timmerman M., Monfret J. Windows NT as Real-Time OS? Real-Time Magazine, № 2, 1997, pp. 6-14.
- 65) Tomura T., Kanai S., Uehiro K., Yamamoto S. Object-Oriented Design Pattern Approach for Modeling and Simulating Open Distributed Control System. Proc. of the IEEE Intern. Conf. on Robotics and Automation, Seoul, Korea, May 21-26, 2001. pp. 211-216.
- 66) Uhl R.G., Lobo N. A Framework for Recognizing a Facial Image from a Police Sketch. Proc of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Francisco, California, USA, June 18-20, 1996. pp. 586-593.
- 67) West P. High Speed, Real-Time Machine Vision. Imagenation and Automated Vision Systems, Inc. 2001. 42 pg. <http://www.imagenation.com>
- 68) Александреску А. Современное проектирование на Си++. Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 336 с.
- 69) Бадд Т. Объектно-ориентированное программирование в действии. Пер. с англ. СПб.: Питер, 1997. 460 с.
- 70) Бек К. Экстремальное программирование. СПб.: Питер, 2002. 224 с.
- 71) Бок К. UML 2: модель деятельности и модель действий. Открытые системы, № 2, февраль 2004, стр. 38-43.
- 72) Бокс Д. Сущность технологии СОМ. Библиотека программиста. СПб.: Питер, 2001, 400 с.
- 73) Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. 3 изд. Пер. с англ. Вильямс, 2017. 720 с.
- 74) Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. 2 изд. Пер. с англ. М.: ДМК, 2006. 496 с.
- 75) Вирт Н. Алгоритмы и структуры данных. Пер. с англ. СПб.: Невский Диалект, 2001. 352 с.
- 76) Волш А.И. Основы программирования на Java для World Wide Web. – К.: Диалектика, 1996. – 512 с.
- 77) Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2016. – 366 с.
- 78) Гома Х. Проектирование систем реального времени, параллельных и распределенных приложений: Пер. с англ. М.: ДМК Пресс, 2016. 700 с.
- 79) Горбунов-Посадов М.М. Расширяемые программы. – М.: Полиптих, 1999. – 336 с.
- 80) Йодан Э. Структурное проектирование и конструирование программ: Пер. с англ. М.: Мир, 1979. 415 с.
- 81) Кёртен Р. Введение в QNX Neutrino. Руководство для разработчиков приложений реального времени. СПб.: БХВ-Петербург, 2015, 368 с.
- 82) Ларман К. Применение UML 2.0 и шаблонов проектирования. 3 изд. Пер. с англ. – М.: Вильямс, 2013. – 736 с.
- 83) Линдли К. Практическая обработка изображений на языке Си: Пер. с англ. – М.: Мир, 1996. – 512 с.
- 84) Мартин Д. Программирование для вычислительных систем реального времени: Пер. с англ. М.: Наука, 1975. 360 с.

- 85) Мейерс С. Эффективный и современный C++. М.: Вильямс, 2018. 304 с.
- 86) Монсон-Хейфел Р. Enterprise JavaBeans, 3 изд.: Пер. с англ. СПб.: Символ-Плюс, 2002. 672 с.
- 87) Олссон Г., Пиани Дж. Цифровые системы автоматизации и управления. СПб.: Невский Диалект, 2001. 557 с.
- 88) Прагг Т., Зелковиц М. Языки программирования: разработка и реализация. – СПб.: Питер, 2002. – 688 с.
- 89) Просиз Дж. Программирование для Microsoft .NET: Пер. с англ. М.: Издательско-торговый дом «Русская редакция», 2003. 704 с.
- 90) Рихтер Дж. Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows: Пер. с англ. – 4 изд. – СПб.: Питер, 2001. – 752 с.
- 91) Руссинович М., Соломон М., Ионеску А. Внутреннее устройство Windows. 7 изд. – СПб.: Питер, 2018. 944 с.
- 92) Стивенс У. UNIX: взаимодействие процессов. СПб.: Питер, 2002. 576 с.
- 93) Страуструп Б. Язык программирования C++. Специальное издание. Бином, 2015, 1136 с.
- 94) Таненбаум Э., Бос Х. Современные операционные системы. 4 изд. – СПб.: Питер, 2018. 1120 с.
- 95) Уайтеккер Дж., Воас Дж. 50 лет программирования: основные принципы качества. // Открытые системы, № 3, 2003, с. 58-63.
- 96) Фаулер М. и др. Рефакторинг: улучшение существующего кода. – Пер. с англ. – Вильямс, 2017. – 448 с.
- 97) Янг С. Алгоритмические языки реального времени: конструирование и разработка: Пер. с англ. М.: Мир, 1985. 400 с.
- 98) Яншин В.В., Калинин Г.А. Обработка изображений на языке Си для IBM PC: Алгоритмы и программы. – М.: Мир, 1994. – 240 с.

## Используемые термины и сокращения

*Перечень терминов и сокращений в ряде случаев (заслуживающих, по мнению авторов, специального внимания) содержит используемое в книге толкование термина или понятия.*

**АД** – асинхронный двигатель

**Андроидный робот** – то же, что двуногий робот, существеннее «похож» на человека.

**Алгоритм обратного распространения ошибки** – алгоритм обратного распространения ошибки является одним из методов обучения многослойных нейронных сетей прямого распространения, называемых также многослойными персептронами.

Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синаптические веса сети фиксированы. Во время обратного прохода все синаптические веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Отсюда и название – **алгоритм обратного распространения ошибки**. Синаптические веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому.

**Аналитическое обучение** – создание программы функционирования робота с использованием языка управления роботом.

Архитектура нейронной сети – структура и топология ИНС.

**ВУБ** – вычислительно-управляющий блок.

**ГП** – гидравлический привод.

**ГР** – гидрораспределители.

**Двуногий робот** – мобильный робот, перемещающийся с помощью двух ног, обычно копирует и моделирует движения человека.

**ДГР** – дросселирующие гидрораспределители.

**ДОС** – датчик обратной связи.

**ДТ** – датчик тока.

**ДП** – датчик положения.

**ДС** – датчик скорости.

**Искусственный интеллект** – (ИИ; англ. Artificial intelligence, AI): наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ; имеющих свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека. ИИ связан со сходной задачей использования компьютеров для понимания человеческого интеллекта, но не обязательно ограничивается биологически правдоподобными методами. ИИ имитирует интеллектуальную деятельность человека и является помощником в решении сложных неформализованных задач. Современные технологии ИИ предполагают использование современной математической логики, сложные неформальные рассуждения (их выполняет программный модуль Рассуждатель), вычисления по индукции, аналогии, абдукции.

### **Интеллектуальный робот (робот с искусственным интеллектом)**

#### Определение 1 (структурное 1 – технологическое)

если робот имеет в своем составе хотя бы одну интеллектуальную технологию он может считаться интеллектуальным, к интеллектуальным технологиям относят:

- производственные системы (построенные на правилах),
- экспертные системы для управления, принятия решения,
- системы нечеткой логики,
- нейронные сети для управления,
- подсистемы генетических алгоритмов,
- ...

И тем более, если у робота несколько таких систем.

#### Определение 2 (структурное 2)

Интеллектуальный робот, это робот со структурой:

Интеллектуальная система + Сенсоры = Когнитивная система +

Эффекторы = Интеллектуальный робот

#### Определение 3 (функциональное)

Интеллектуальный робот, это робот, которому дают задание в общем смысле – пойдти туда ... и принеси то то ... Смысл: робот должен понять команду (просьбу) и спланировать свое поведение так, чтобы ее выполнить – функциональная интерпретация.

#### Определение 4 (восходящее к бихевиористскому)

Интеллектуальный робот, это робот, успешно решающий интеллектуальную задачу, требующую сложной сенсорной обработки, неформализованных вычислений, требующую реализации сложного поведения. Или решающий задачу, которая СЧИТАЕТСЯ интеллектуальной. Или робот, который с точки

зрения людей реализует адекватное поведение в человеческой среде и обстановке.

**ИСУ** – интеллектуальная система управления.

**КП** – контур положения.

**Манипулятор** – механизм, состоящий из последовательных или параллельных сегментов (звеньев), которые обеспечивают перемещение переносимого объекта относительно основания манипулятора.

**Манипуляционный робот** – манипулятор, снабженный системой управления, которая обеспечивает его работу по программе, созданной в процессе обучения.

**Многослойный перцептрон / Multi Layered Perceptron MLP** – среди разных структур ИНС одной из простых является многослойная ИНС – так называемый многослойный перцептрон (Multi Layered Perceptron MLP). В MLP все нейроны одинаковы, они организованы в несколько слоев: входной слой, один или несколько скрытых слоев и выходной слой. К нейронам входного слоя подключены датчики и источники входных сигналов, естественно, число входных нейронов равно числу датчиков и источников информации, вводимой в сеть. Выходные нейроны формируют выходные сигналы сети, их число равно числу требуемых выходов. Число нейронов скрытых слоев обычно подбирают экспериментально, их число выбирают больше числа входных и выходных нейронов. Топология связей в перцептроне такова: от каждого нейрона каждого слоя связи ведутся «вперед» по сети по одной к каждому нейрону следующего слоя. Связи обратного направления отсутствуют. В результате число выходных аксонов каждого нейрона обычно получается больше одного (или он разветвляется) и в этом заключается расширение формальной модели МакКаллока-Питтса.

**Мобильный робот** – робот, перемещающийся в среде обитания на колесах, гусеницах, на ногах, летающие роботы обычно мобильными не называют.

**Модель нейрона МакКаллока-Питтса** – Искусственный нейрон (математический нейрон Маккаллока-Питтса, формальный нейрон) – узел искусственной нейронной сети, являющийся упрощённой моделью естественного нейрона. Математически, искусственный нейрон обычно представляют как некоторую нелинейную функцию от единственного аргумента – линейной комбинации всех входных сигналов. Данную функцию называют функцией активации<sup>1</sup> или функцией срабатывания, передаточной функцией. Полученный результат посылается на единственный выход. Такие искусственные нейроны объединяют в сети – соединяют выходы одних нейронов с входами других. Искусственные нейроны и сети являются основными элементами идеального нейрокомпьютера. (Википедия)

**Нейрон** – от др.-греч. νεῦρον – волокно, нерв – структурно-функциональная единица нервной системы. Нейрон – электрически возбудимая клетка, которая обрабатывает, хранит и передает информацию с помощью электрических и

химических сигналов. Нейрон имеет сложное строение и узкую специализацию. Клетка содержит ядро, тело клетки и отростки (дендриты и аксоны). В головном мозге человека насчитывается около 85-86 миллиардов нейронов. Нейроны могут соединяться один с другим, формируя биологические нейронные сети. Нейроны разделяют на рецепторные, эффекторные и вставочные (Википедия)

**Обучение нейронной сети** – вычисление весовых коэффициентов ИНС, приписанных связям в ИНС.

**Обучение с учителем** – (англ. Supervised learning) – один из способов машинного обучения, в ходе которого испытуемая система принудительно обучается с помощью примеров «стимул-реакция». Между входами и эталонными выходами (стимул-реакция) может существовать некоторая зависимость, но она неизвестна. Известна только конечная совокупность прецедентов – пар «стимул-реакция», называемая обучающей выборкой. На основе этих данных требуется восстановить зависимость (построить модель отношений стимул-реакция, пригодных для прогнозирования), то есть построить алгоритм, способный для любого объекта выдать достаточно точный ответ. Для измерения точности ответов, так же как и в обучении на примерах, может вводиться функционал качества. Обучение с учителем – распространенный способ обучения ИНС. (Википедия)

**Обучение без учителя** – (самообучение, спонтанное обучение, англ. Unsupervised learning) – один из способов машинного обучения, при котором испытуемая система спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. С точки зрения кибернетики, это является одним из видов кибернетического эксперимента. Как правило, это пригодно только для задач, в которых известны описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами. Обучение без учителя часто противопоставляется обучению с учителем, когда для каждого обучающего объекта принудительно задаётся «правильный ответ», и требуется найти зависимость между стимулами и реакциями системы (Википедия).

**Обучающая выборка** – в технологии обучения с учителем дискретный набор пар вход-выход («правильный ответ»), может быть достаточно большим, содержать несколько тысяч обучающих пар.

**Обучение манипулятора** – процесс создания программы функционирования манипуляционного робота.

**ОУ** – объект управления

**ОДУ** – обыкновенные дифференциальные уравнения

**П** – пропорциональный регулятор

**Персептрон** – (англ. Perceptron от лат. perceptio – восприятие; нем. perceptron) – математическая или компьютерная модель восприятия информации мозгом (кибернетическая модель мозга), предложенная Фрэнком Розенблаттом в 1957 году и реализованная в виде электронной машины «Марк-1» в 1960 году. Персептрон стал одной из первых моделей нейросетей, а «Марк-1» – первым в мире нейрокомпьютером. Несмотря на свою простоту, персептрон способен обучаться и решать довольно сложные задачи. Основная математическая задача, с которой он справляется, – это линейное разделение любых нелинейных множеств, так называемое обеспечение линейной сепарабельности. (Википедия)

**ПИ** – пропорционально-интегральный регулятор

**ПИД** – пропорционально-интегральный-дифференциальный регулятор

**Позиционное обучение** – процесс запоминания координат целевых точек движения манипулятора путем выведения его в заданные позиции.

**Промышленный манипулятор** – манипуляционные роботы конструктивные, элементы которых соответствуют стандартам, предъявляемым к промышленному оборудованию.

**ППП** – пакет прикладных программ

**ПСА** – повышенная степень автономности

**РВ** – реальное время

**РБ** – регистрирующий блок

**Рекуррентная нейронная сеть Джордана** – эта сеть получается из MLP путем введения обратных связей – связей от некоторых нейронов к нейронам предыдущих слоев (не обязательно ко всем).

**РМ** – регулятор момента

**РС** – регулятор скорости

**Робот** – автоматическое устройство, предназначенное для осуществления различного рода механических операций, которое действует по заранее заложенной программе, программа может быть адаптивной (Википедия).

**РТК** – робототехнический комплекс

**РТС** – робототехническая система

**РТ** – регулятор тока

**ТП** – транзисторный преобразователь

**САУ** – система автоматического управления

**Система управления роботом-манипулятором** – устройство, которое обеспечивает движение манипулятора по заданной траектории, создание и исполнение программы функционирования робота, взаимодействие с оператором.

**СИО** – система информационного обеспечения

**СУ** – система управления

**СТЗ** – система технического зрения

**Тренировка нейронной сети** – то же, что Обучение нейронной сети

**Универсальный манипулятор** – манипулятор, конструкция и характеристики которого делают возможным его применение для решения технологических задач в широком круге приложений.

**Шагающий робот** – мобильный робот, перемещающийся с помощью ног, многоногий (например, шестиногий), или четырехногий, или двуногий.

**Шестиногий робот** – мобильный робот, перемещающийся с помощью шести ног, расположение ног на корпусе может быть различным.

**ШИМ** – широтно-импульсная модуляция

**ШИП** – широтно-импульсный преобразователь

**ШМ** – шагающая машина

**ЭГСП** – электрогидравлический следящий привод

**ЭГУ** – электрогидравлический усилитель

**ЭМП** – электромеханический преобразователь

**ЭП** – электрический привод

**ЭД** – электрический двигатель

**Язык управления роботом** – язык программирования, который содержит операторы, обеспечивающие функционирование робота.

**LS-регулятор** – (Load Sensing) – регулятор чувствительный к нагрузке.