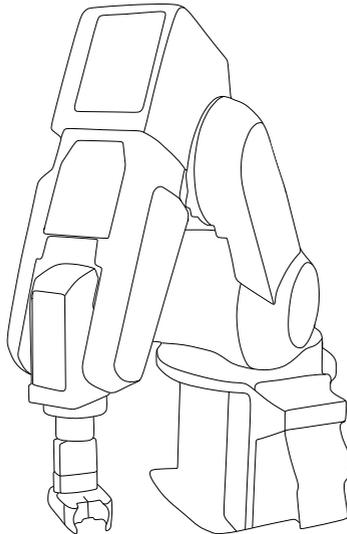


# **ІТМО**

**В.С. Громов, О.И. Борисов, М.М. Синетова**

## **ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ РОБОТОВ СЕРИИ MITSUBISHI MELFA**



**Санкт-Петербург**

**2023**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
УНИВЕРСИТЕТ ИТМО

**В.С. Громов, О.И. Борисов, М.М. Синетова**

**ПРОГРАММИРОВАНИЕ  
ПРОМЫШЛЕННЫХ РОБОТОВ СЕРИИ  
MITSUBISHI MELFA**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

**РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ  
В УНИВЕРСИТЕТЕ ИТМО**

**по направлению подготовки 15.03.06, 27.03.04 в качестве  
учебно-методического пособия для реализации основных  
профессиональных образовательных программ высшего образования  
бакалавриата**

**ИТМО**

**Санкт-Петербург**

**2023**

В.С. Громов, О.И. Борисов, М.М. Синетова. Программирование промышленных роботов серии Mitsubishi Melfa — СПб: Университет ИТМО, 2023. — 80 с.

**Рецензент:**

А.А. Ведяков, к.т.н., доцент, доцент (квалификационная категория «ординарный доцент») факультета систем управления и робототехники, Университет ИТМО, Санкт-Петербург, Россия.

В учебном пособии рассматриваются основные органы управления, алгоритмические конструкции, команды управления движением, принципы работы с внешними устройствами промышленных роботов.



**Университет ИТМО** (Санкт-Петербург) — ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО — участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО — становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2023

© В.С. Громов, О.И. Борисов, М.М. Синетова, 2023

# Оглавление

<b>Введение</b>	<b>6</b>
<b>1 Понимание промышленных роботов</b>	<b>7</b>
1.1 Конструкция промышленного робота . . . . .	7
1.1.1 Типовые составные части промышленных роботов	7
1.1.2 Характеристики промышленных роботов . . . . .	12
1.1.3 Непромышленная робототехника . . . . .	15
1.1.4 Требования к подготовке помещений . . . . .	15
1.2 Классификация промышленных роботов . . . . .	18
1.3 Типы промышленных роботов . . . . .	20
1.3.1 Артикулированные роботы . . . . .	20
1.3.2 Дельта роботы . . . . .	21
1.3.3 Параллельные роботы . . . . .	22
1.3.4 SCARA . . . . .	23
1.3.5 Коллаборативные роботы . . . . .	23
1.3.6 Мобильные роботы . . . . .	24
1.3.7 Экзоскелеты . . . . .	24
1.4 Захватные устройства . . . . .	25
1.5 Области применения . . . . .	28
1.5.1 Классические области применения . . . . .	28
1.5.2 Новые области применения промышленных роботов . . . . .	29
1.5.3 В каких процессах используют роботов . . . . .	30
1.5.4 Влияние роботизации . . . . .	32
1.6 Контрольные вопросы . . . . .	34

<b>2</b>	<b>Основы взаимодействия с промышленным роботом</b>	<b>35</b>
2.1	Основные органы управления . . . . .	36
2.1.1	Кнопки аварийной остановки . . . . .	36
2.1.2	Элементы управления в ручном режиме . . . . .	36
2.1.3	Элементы управления в автоматическом режиме	38
2.2	Работа с интерфейсом программирования на компьютере	39
2.3	Контрольные вопросы . . . . .	44
<b>3</b>	<b>Синтаксис языка программирования Melfa Basic</b>	<b>46</b>
3.1	Типы переменных . . . . .	46
3.2	Алгоритмические конструкции . . . . .	47
3.2.1	Команды начала и завершения работы . . . . .	47
3.2.2	Условия . . . . .	48
3.2.3	Циклы . . . . .	49
3.2.4	Процедуры . . . . .	50
3.2.5	Функции . . . . .	51
3.2.6	Прерывания и таймеры . . . . .	52
3.2.7	Паузы . . . . .	53
3.3	Команды управления движением . . . . .	54
3.3.1	Приближённое движение в точку . . . . .	54
3.3.2	Точное движение в точку . . . . .	56
3.3.3	Движение по дуге . . . . .	57
3.3.4	Формирование комплексных траекторий . . . . .	59
3.4	Работа с внешними устройствами . . . . .	60
3.4.1	Управление захватными устройствами . . . . .	60
3.4.2	Порты ввода-вывода . . . . .	62
3.4.3	Сетевые интерфейсы . . . . .	63
3.5	Контрольные вопросы . . . . .	64
<b>4</b>	<b>Лабораторные работы по курсу «Программирование промышленных роботов»</b>	<b>65</b>
4.1	Лабораторная работа 1 «Преобразования координат и циклы» . . . . .	65
4.1.1	Задание на лабораторную работу . . . . .	65
4.1.2	Основные команды программирования . . . . .	65
4.1.3	Порядок выполнения . . . . .	67
4.1.4	Содержание отчета . . . . .	68
4.2	Лабораторная работа 2 «Порты ввода-вывода и условия»	68

---

4.2.1	Задание на лабораторную работу . . . . .	68
4.2.2	Основные команды программирования . . . . .	69
4.2.3	Порядок выполнения . . . . .	70
4.2.4	Содержание отчета . . . . .	70
4.3	Лабораторная работа 3 «Формирование комплексных траекторий» . . . . .	71
4.3.1	Задание на лабораторную работу . . . . .	71
4.3.2	Основные команды программирования . . . . .	72
4.3.3	Порядок выполнения . . . . .	72
4.3.4	Содержание отчета . . . . .	72
4.4	Лабораторная работа 4 «Сетевые сообщения» . . . . .	73
4.4.1	Задание на лабораторную работу . . . . .	73
4.4.2	Основные команды программирования . . . . .	73
4.4.3	Необходимые команды программирования на Python . . . . .	74
4.4.4	Порядок выполнения . . . . .	75
4.4.5	Содержание отчета . . . . .	75
	<b>Заключение</b>	<b>77</b>
	<b>Список литературы</b>	<b>79</b>

# Введение

Данное учебно-методическое пособие посвящено вопросам взаимодействия и управления, а также программирования промышленных роботов серии Mitsubishi Melfa, применяющихся в рамках лабораторных работ курсов «Программирование промышленных роботов» и «Программное обеспечение промышленных роботов», реализуемых на факультете систем управления и робототехники Университета ИТМО. Пособие содержит информацию об основных компонентах промышленных роботов, примерах областей применения, порядке действий для безопасного взаимодействия, основных командах управления. Завершается учебно-методическое пособие циклом лабораторных работ.

В теоретической части учебного пособия рассматриваются основные органы управления, алгоритмические конструкции, команды управления движением, принципы работы с внешними устройствами промышленных роботов. Данные теоретические материалы рекомендуются использовать как один из источников информации в дополнение к материалам лекций и как теоретическая подготовка перед выполнением лабораторных работ.

В данном учебно-методическом пособии представлены указания по выполнению четыре лабораторных работы, охватывающих следующие разделы: преобразования координат и циклы, порты ввода-вывода и условия, формирование комплексных траекторий, сетевые сообщения.

Пособие предназначено для студентов факультета систем управления и робототехники Университета ИТМО, обучающихся по направлению подготовки 15.03.06 «Мехатроника и робототехника», 27.03.04 «Управление в технических системах» в качестве учебно-методического пособия для реализации основных профессиональных образовательных программ высшего образования бакалавриата

# Глава 1

## Понимание промышленных роботов

### 1.1 Конструкция промышленного робота

#### 1.1.1 Типовые составные части промышленных роботов

Робот — автоматическое устройство, предназначенное для взаимодействия с окружающей средой с целью выполнения различных механических операций по заложенной программе [1]. Первые примеры роботов, предназначенные для автоматизации перемещений грузов на производствах по сложной траектории, появились в середине 20 века [4]. Программирование таких устройств осуществлялось за счёт разработки определенных механизмов, которые обеспечивали необходимые перемещения. Для изменения программы требовалось вносить изменения в существующие механизмы либо разрабатывать полностью новое устройство.

По мере развития технологий произошла цифровизация робототехнических устройств [3]. Теперь программу с последовательностью действий технологических операций можно было представлять в виде алгоритма, который будет исполняться микроконтроллером робота. Внесение существенных изменений в конструкцию робототехнического устройства при корректировке программы в этом случае не требуется. Цифровизация позволила существенно расширить области применения роботов. Сегодня они используются не только на крупных предприятиях, но и доступны среднему и малому бизнесу. Кроме того они нашли применение в быту для некоторых формализуемых задач.

Согласно теории механизмов и машин [5], роботы состоят из:

- звеньев — деталей, либо групп деталей, жестко связанных между собой;
- шарниров — соединений звеньев, допускающих их относительное движение.

Взаимное расположение этих элементов показано на рисунке 1.1. Число подвижных соединений в робототехнической системе определяет число степеней свободы, являющееся одним из основных параметров при выборе робота.

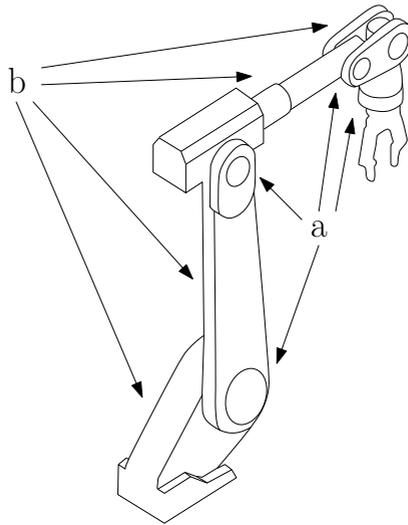


Рис. 1.1. а) шарниры робота, б) звенья робота

Тип применяющихся шарниров в робототехнической системе определяет кинематическую пару. Кинематические пары бывают следующих видов:

- вращательные;
- призматические;
- комбинированные.

Вращательные кинематические пары обеспечивают поворот двух звеньев друг относительно друга на угол поворота вокруг выбранной оси вращения. Схематичное изображение вращательной кинематической пары показано на рисунке 1.2.

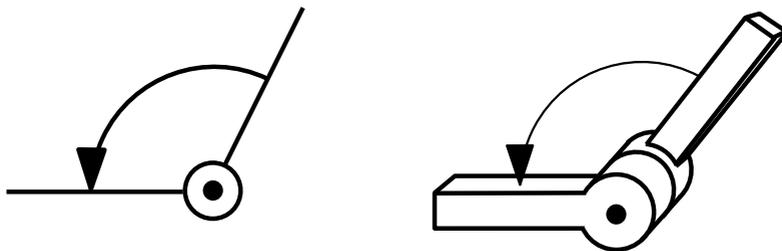


Рис. 1.2. Вращательные кинематические пары

Призматические кинематические пары обеспечивают смещение двух звеньев друг относительно друга на линейное смещение вдоль выбранной оси. Схематическое изображение призматической кинематической пары показано на рисунке 1.3.

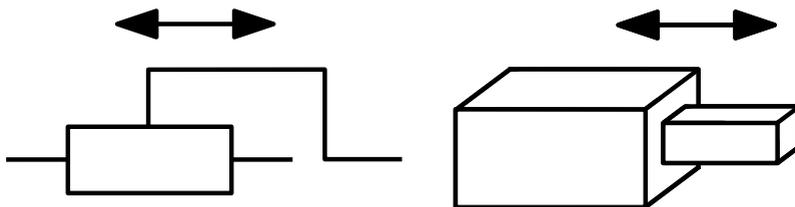


Рис. 1.3. Призматические кинематические пары

Обеспечить перемещение звена на заданное линейное расстояние в окружающем пространстве можно как с помощью призматической, так и вращательной кинематической пары. В первом случае перемещение осуществляется за счёт задания требуемого смещения призматического шарнира. Во втором случае перемещение осуществляется за счёт пово-

рота вращательного шарнира на такой угол, чтобы конечное положение звена оказалось в требуемом положении окружающего пространства. Примеры перемещения звеньев с использованием разных кинематических пар показаны на рисунке 1.4.

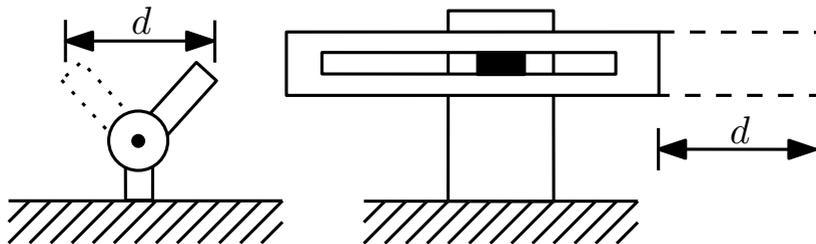


Рис. 1.4. Перемещение на заданное расстояние с использованием различных кинематических пар

Комбинированные кинематические пары представляют собой комбинацию нескольких вращательных или призматических простейших типов пар, пересекающихся в одной точке. Одним примером комбинированных кинематических пар могут быть карданные соединения, которые представляют собой два вращательных шарнира, пересекающихся в одной точке. Другим примером может быть шаровое соединение звеньев, представляющее собой пересечение трёх вращательных шарниров, пересекающихся в одной точке.

За счёт комбинаций различных звеньев и шарниров выстраивается кинематическая схема робота, которая будет определять общие габариты робота, число степеней свободы, общую подвижность робота и тот объем функций, который в итоге можно будет выполнять с его помощью.

В подавляющем большинстве роботов [2] представлены следующие составные части, необходимые для его корректной работоспособности:

- звенья и шарниры;
- приводы;
- датчики;
- носимый инструментарий;

- контроллер;
- блок питания;
- пульт программирования и управления.

Рассмотрим каждый компонент поподробнее. Как уже поминалось ранее, звенья и шарниры определяют кинематическую схему робота, от которой зависят габариты робота, число степеней свободы, рабочая область робота, общая подвижность.

Для приведения шарниров в движение устанавливаются приводы. В подавляющем большинстве современных роботов устанавливают электроприводы. Однако встречаются модели роботов с пневматическими или гидравлическими системами (управляющиеся в свою очередь электрическими клапанами), которые обеспечивают движение звеньев.

Датчики устанавливаются для отслеживания текущего состояния параметров робота. Основная категория датчиков — энкодеры, необходимые для отслеживания текущего положения шарниров робота. Они используются в том числе в обратной связи систем управления по положению для позиционирования звеньев. Кроме того, устанавливаются датчики контроля критичных параметров роботов, например, датчики тока для отслеживания потребления электродвигателей. Этот вид датчиков используется как для обратной связи регуляторов тока, так и для ограничения максимального тока, подаваемого на двигатели.

Робототехнические системы чаще всего взаимодействуют с окружающей средой не посредством звеньев конструкции самого робота, а с помощью носимого инструментария, который закрепляется на специальном установочном фланце. Носимый инструментарий может быть представлен как в виде датчиков, которые собирают данные об окружающей среде, так и представлен в виде инструмента, который напрямую взаимодействует с окружающей средой (например, дрели, сварочные головки и т.д.). Также имеется возможность установить захватные устройства, которые позволяют, будучи установленными на робота, захватывать объекты окружающего пространства и переносить на новое место.

Для хранения и исполнения алгоритмов, отслеживания параметров системы, формирования сигналов управления на исполнительные устройства робототехнической системы устанавливаются контроллеры

системы управления. Кроме основных компонентов современной системы управления, вроде микропроцессора, постоянной и оперативной памяти, контроллер также оснащён мощными драйверами управления двигателями, а также платами цифровых входов и выходов. Контроллер чаще всего помещён в специальный пылевлагозащитный и ударопрочный корпус для работы в возможных тяжелых условиях производства, поэтому общая сборка будет иметь значительные габариты.

Блок питания обеспечивает электроэнергией основные компоненты робототехнической системы. Если робот предполагается к стационарному использованию, то он оснащается преобразователями электроэнергии (блоками питания) для подключения к общественной электросети. Варианты робототехнических систем, предусматривающие мобильное использование, оснащаются блоками батарей, схемами контроля балансировки заряда.

Для возможности взаимодействия человека с роботом, например для контроля текущих параметров, дистанционного управления положением робота напрямую или формирования будущего алгоритма, промышленные робототехнические системы оснащаются пультами программирования и управления. Для обеспечения стабильности связи и надежности функционирования в условиях промышленных предприятий пульта управления изготавливают в специальных ударопрочных корпусах и подключают посредством ударопрочных кабелей. Беспроводное подключение пульта является редкостью, так как на производствах могут иметь место электромагнитные помехи, вызванные результатом работы мощных силовых машин.

### 1.1.2 Характеристики промышленных роботов

Основными характеристиками промышленных роботов являются:

- число степеней свободы (управляемых и неуправляемых);
- погрешность позиционирования (повторяемость);
- грузоподъемность;
- рабочее пространство;
- максимальная линейная скорость.

Число степеней свободы определяет количество координат, по которым возможно позиционирование носимого инструментария в окружающем пространстве. Выделяют управляемые и неуправляемые степени свободы. Управляемые позволяют напрямую контролировать положение шарниров и звеньев робота посредством приводов. Неуправляемые могут использоваться в специфических задачах, не содержат приводов прямого управления; позиционирование таких степеней свободы осуществляется косвенными средствами, например, с помощью действия гравитационных или инерционных сил. Увеличение степеней свободы позволяет позиционировать объекты по большему числу координат, но это приводит к увеличению количества двигателей, увеличению массы устройства, удорожанию всей системы.

Погрешность позиционирования является важным параметром для определения того, может ли быть задача выполнена с требуемой точностью или нет. Однако, в силу сложной конструкции промышленного робота, производители часто не дают информацию о погрешности позиционирования в техническом паспорте устройства. Вместо этого предоставляется информация о так называемой «повторяемости» устройства. Повторяемость означает, что если робот будет функционировать в одинаковых условиях и выполнять повторяющиеся операции, то позиционирование в этих условиях будет осуществляться с определенным отклонением. Условия эксплуатации должны включать в себя стабильность рабочих температур, отсутствие вибраций, выполнение планового технического обслуживания в установленные сроки. При соблюдении условий величина возможного отклонения будет сохраняться. Для повышения повторяемости робота требуется высокоточное изготовление всех его компонентов, что приводит к удорожанию устройства.

Параметр грузоподъемности определяет максимальную массу переносимых роботом грузов. Важно учитывать, что этот параметр включает в себя как массу переносимых объектов, так и носимого инструментария, который будет закрепляться на роботе. Для переноса тяжелых грузов требуются модели с мощными двигателями и звеньями, выдерживающими большие нагрузки, что приводит к увеличению массы устройства и удорожанию системы.

Размер рабочего пространства робота определяется его кинематической схемой. Рабочее пространство зачастую представляет собой неправильную фигуру. Внутри рабочего пространства требуется разместить

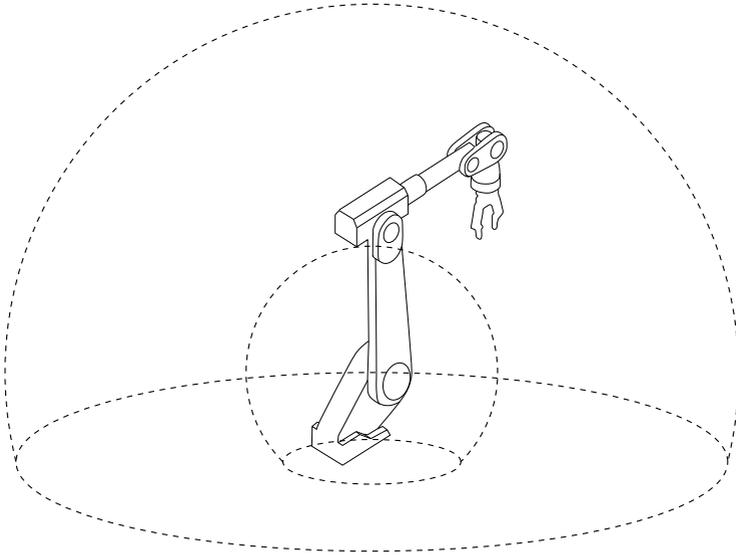


Рис. 1.5. Пример рабочего пространства робота

все объекты и инструменты, с которыми будет взаимодействовать робот. Пример рабочего пространства робота показан на рисунке 1.5. Для увеличения рабочего пространства робота необходимо увеличивать длину звеньев, что приводит к необходимости установки приводов, способных создавать больший крутящий момент. В связи с этим увеличивается масса и стоимость робототехнической системы. Для оптимизации расходов следует пересматривать технологический процесс с целью использования всего доступного рабочего пространства робота (например, хранить детали вокруг робота вертикально, а не горизонтально).

Для автоматизации многих технологических процессов посредством внедрения робототехнических систем важно знать максимальную развиваемую линейную скорость для обеспечения требуемой производительности. Но не все производители предоставляют данные о линейных скоростях, развиваемых роботом. Если робот содержит только вращательные сочленения, то может быть предоставлена информация о максимальных угловых скоростях вращения сочленения. Значения ли-

нейных скоростей могут быть рассчитаны с помощью преобразований. Требование увеличения максимальной линейной скорости также приводит к необходимости установки мощных двигателей, что удорожает систему, по аналогии с другими характеристиками робота.

Таким образом, выбор робота обуславливается определением критичных характеристик, трактуемых технологическим процессом, в рамках доступного бюджета.

### **1.1.3 Непромышленная робототехника**

Важно различать промышленные и непромышленные робототехнические системы, доступные в наши дни на рынке. Непромышленные робототехнические системы предназначены для решения бытовых формализуемых задач, имеют доступную стоимость для рядового потребителя, безопасны для людей и не требуют специальных навыков для запуска системы в работу. Однако, такие системы зачастую не предназначены для непрерывной работы в тяжелых условиях промышленных площадок.

Промышленные роботы, в свою очередь, способны работать без перерыва на долгих промежутках времени. Для обеспечения такой возможности, промышленные робототехнические системы изготавливают в специальных корпусах, защищённых от внешних воздействий окружающей среды. Промышленные роботы обладают характеристиками высокой грузоподъёмности за счёт использования мощных электротехнических решений. Как следствие, они могут быть потенциально опасны для человека и требуют специальной подготовки всего персонала, который будет взаимодействовать с роботом. Для обеспечения возможности непрерывной работы промышленные роботы сопровождаются расширенными возможностями гарантии и обслуживания для снижения времени простоя и снижения издержек в случае поломки робота. Все меры по обеспечению надежности работы и высоких характеристик приводит к удорожанию конечного устройства, делая его недоступным для рядового потребителя и некоторых малых предприятий.

### **1.1.4 Требования к подготовке помещений**

На рынке представлены как коллаборативные (оснащены комплексными системами безопасности, безопасны для человека), так и неколлабо-

ративные роботы (как правило, мощнее, но потенциально опасны для человека). Промышленные неколлаборативные роботы требуют специальной подготовки помещений для обеспечения мер безопасности рабочего персонала. Выполнение данных требований позволит сохранить здоровье и жизни сотрудников в случае возникновения чрезвычайной ситуации.

В первую очередь, требуется установить ограждение вокруг всей рабочей области робота. Ограждение не позволит сотрудникам проникнуть в рабочую область во время выполнения алгоритмов, сохраненных в памяти робота. Физическое отделение рабочей области показало с годами свою эффективность в сравнении с другими подходами, например, с расчерчиванием рабочей области на полу, так как в противном случае сотрудники имели возможность по невнимательности или загруженности попасть в потенциально опасную для здоровья и жизни область.

Тем не менее, для периодического обслуживания робота или внесения корректировок в алгоритмы управления требуется подходить в непосредственную близость к роботу. Для этого в ограждении робота устанавливаются двери или ворота, оснащенные концевыми датчиками открытия. Концевые датчики подключаются в контур безопасности контроллера робота, не давая в случае открытия запустить робота в автоматическом режиме и снижая максимальную скорость в ручном режиме. Наличие такого элемента безопасности позволяет избежать нанесения вреда здоровью при открытии ограждения безопасности.

В случае невозможности установки ограждения по всему периметру рабочей области робота, например, если конвейер доставляет детали к роботу, устанавливаются оптические датчики присутствия человека. Такие датчики не позволят проникнуть сотрудникам в потенциально опасную область через открытую часть ограждения.

Для остановки функционирования робота в случае возникновения чрезвычайной ситуации устанавливаются кнопки аварийной остановки. Данный элемент выполнен в виде большой красной кнопки, которая фиксируется при нажатии. Аварийной бездвижное состояние робота будет сохраняться всё время, пока кнопка находится в нажатом состоянии. Это необходимо для того, чтобы робот случайно не пришёл в движение, пока люди рядом с ним занимаются устранением чрезвычайной ситуации. Для возвращения аварийной кнопки в исходное состояние необходимо повернуть её по часовой стрелке. Кнопки аварийной установки необходимо монтировать в легкодоступном для персонала месте, они

подключаются к контроллеру промышленного робота, их должно быть не менее одной единицы.

На промышленной площадке необходимо выделять специальное место для размещения контроллера робота. Как уже упоминалось ранее, контроллер выполнен в специально пылевлагозащищенном ударопрочном корпусе и имеет значительные габариты, которые необходимо учитывать при размещении на площадке. Кроме того, его нельзя размещать в рабочей области, чтобы в случае ошибки в алгоритме робот своим движением случайно не разбил свой контроллер.

Опционально помещение с промышленной робототехнической системой может быть оборудовано оптическими и/или звуковыми средствами оповещения персонала на площадке промышленного предприятия о текущем режиме работы робота.

Пример размещения необходимого оборудования на производственной площадке приведён на рисунке 1.6.

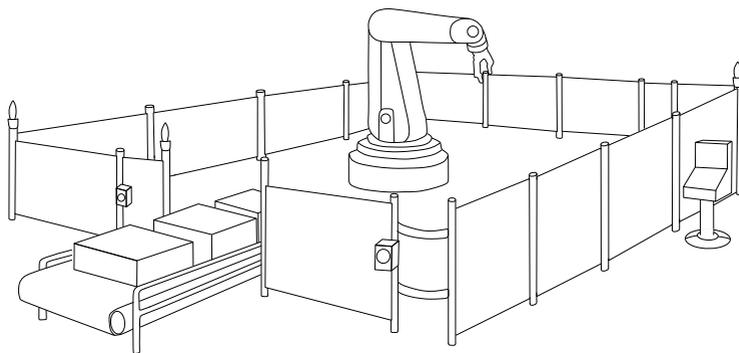


Рис. 1.6. Пример подготовки помещения для промышленного робота

Если на производстве предполагается применение мобильных промышленных роботов, то помещение также необходимо подготовить для обеспечения бесперебойной работы. Для надёжной навигации мобильных роботов по площадке на пол наклеивают магнитные ленты, которые будут показывать допустимые направления движения роботов. Посредством датчиков магнитного поля будет считываться требуемое направление даже в случаях, когда пол производственной площадки загрознён. Это отличает подход наклеивания магнитных лент от нанесения

ния линий направления обычной краской. Если от мобильного робота требуется полное следование траектории без малейших отклонений в сторону, тогда роботизированную систему устанавливают на рельсы, по которым будет осуществляться перемещение. Для надёжной передачи данных на большие расстояния в условиях возможного наличия электромагнитных помех на стенах помещения устанавливают проволочные антенны. В случае перемещения робота между различными помещениями устанавливаются специальные оптические метки (например, qr-коды), содержащие информацию о текущем помещении. В этом случае метки устанавливаются на стенах или потолке, где они не будут случайно захламлены.

## 1.2 Классификация промышленных роботов

В зависимости от того, какой параметр является критическим для автоматизации технологического процесса на производстве, на рынке можно встретить разные классификации промышленных роботов. Данные классификации в большинстве случаев являются условными, но позволяют быстро подбирать необходимые модели под поставленные требования.

Распространенной является классификация роботов по грузоподъемности. Выделяют сверхлёгкие (до 1 кг) и легкие (от 1 до 10 кг). Данные модели применяют для небольших сборочных операций, конструкция роботов позволяет закреплять их на полу, столе или мобильной платформе. Далее идут роботы средней грузоподъемности (от 10 до 200 кг). Такие роботы должны быть надежно закреплены на полу; установка на мобильную платформу допускается с ограничением максимальной скорости. Последние категории в данной классификации — это тяжелые (от 200 до 1000 кг) и сверхтяжелые (более 1000 кг). Для установки таких роботов необходимо внимательно сверяться с паспортом устройства и рассчитывать, какие нагрузки робот будет создавать на междуэтажные перекрытия. В некоторых случаях может потребоваться строительство специального помещения для установки тяжелого робота. Кроме того, сверхтяжелые роботы зачастую содержат в конструкции одновременно электродвигатели и гидравлические приводы для компенсации больших нагрузок, что может потребовать установки на площадке дополнительного оборудования.

По числу степеней свободы на рынке можно встретить модели с числом от трёх до семи. Уменьшение числа степеней свободы приводит к облегчению конструкции и возможности развивать высокие скорости с высокой точностью, но накладывает ограничения на число координат, по которым может быть позиционирован рабочий инструмент. В свою очередь, увеличение числа степеней свободы позволяет позиционировать носимый инструментарий по большому числу координат, но с ограничениями по скорости. Для артикулированных промышленных роботов (в следующем параграфе будет дано определение) в большинстве представлены модели с шестью степенями свободы, так как они позволяют позиционировать объекты по шести координатам прямоугольной системы координат.

По возможности передвижения промышленные роботы бывают стационарные и подвижные. На рынке можно встретить модели роботов, которые изначально предназначены для мобильного использования, так и модели, которые можно адаптировать путем установки на мобильную платформу. Также существуют модели, у которых в паспорте устройства отдельно прописано, что они должны быть установлены как неподвижные, мобильная установка запрещена, так как это приводит к некорректному функционированию механизмов. В случае необходимости мобильного использования промышленного робота необходимо сначала внимательно свериться с его паспортом касаясь возможной такой эксплуатации.

По типу установки можно встретить напольные, настенные, подвесные промышленные роботы. Необходимо сверяться с паспортом устройства о возможности закрепления робота в том или ином положении. Конструкции некоторых моделей не позволяют проводить установку в незапланированном положении в силу некорректного функционирования механизмов.

По переносимому оборудованию иногда выделяют диагностирующие (позиционируют датчик или комплекс датчиков для сбора данных об окружающей среде), инструментальные (позиционируют инструмент для непосредственного взаимодействия с окружающей средой), удерживающие (позиционируют объекты окружающего пространства посредством захватного устройства).

## 1.3 Типы промышленных роботов

К основным типам промышленных роботов, представленных на рынке, относятся:

- артикулированные;
- дельта;
- параллельные;
- SCARA;
- коллаборативные;
- мобильные;
- экзоскелеты.

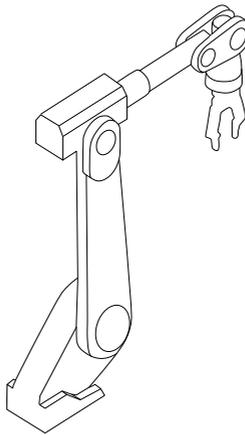


Рис. 1.7. Пример артикулированного робота

### 1.3.1 Артикулированные роботы

Артикулированные роботы являются наиболее распространенным типом промышленных роботизированных систем. Такие роботы содержат последовательные вращательные сочленения и от трёх до шести

степеней свободы, обеспечивающие гибкость в применении. За счёт применения шести степеней свободы позволяют позиционировать инструменты и объекты по шести координатам прямоугольной системы координат. Применяются в широком спектре операций: сборка, покраска, сварка, паллетирование, удержание и т.д. Пример артикулированного робота представлен на рисунке 1.7.

### 1.3.2 Дельта роботы

Дельта роботы содержат три звена, каждый из которых закреплен на базе робота. База робота чаще всего является верхней точкой и включает приводы звеньев (линейные или вращательные), что позволяет сделать звенья очень лёгкими и подвижными. За счёт такой кинематической схемы робот может развивать высокие скорости с высокой точностью в ущерб доступным координатам позиционирования и грузоподъемности. Применения: упаковка, сборка с высокой скоростью и точностью. Пример дельта робота представлен на рисунке 1.8.

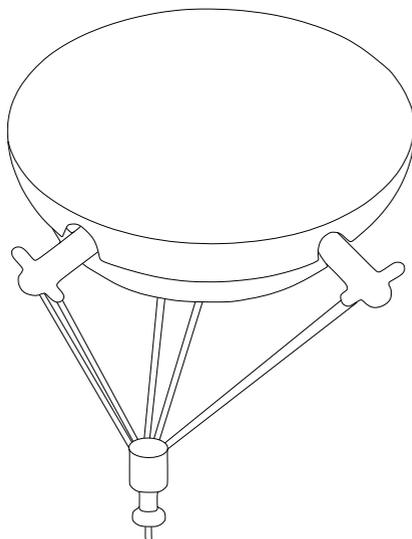


Рис. 1.8. Пример дельта робота

Артикулированные и дельта роботы возможно использовать совместно. Рассмотрим пример использования промышленных роботов на конвейере, где из отдельных деталей необходимо собирать крупные партии и передавать на упаковку. В этом случае дельта робот быстро и точно перемещает отдельные детали, артикулированный робот двигается медленнее, но переносит большую массу нескольких деталей сразу на большие расстояния.

### 1.3.3 Параллельные роботы

Параллельный робот состоит из трёх управляемых осей, которые расположены под  $90^\circ$  друг к другу. Вдоль осей происходит параллельное перемещение; такую кинематическую схему очень легко программировать для перемещения грузов в прямоугольной системе координат. Применяется в узкой области задач, где ограничено пространство для размещения робота и требуются только линейные перемещения. Возможно масштабирование системы. Роботы такой кинематики могут быть выполнены как в масштабе стола, например, для автоматизации химических исследований, так и в масштабе склада, например, для решения задачи автоматической сортировки грузов. Пример параллельного робота представлен на рисунке 1.9.

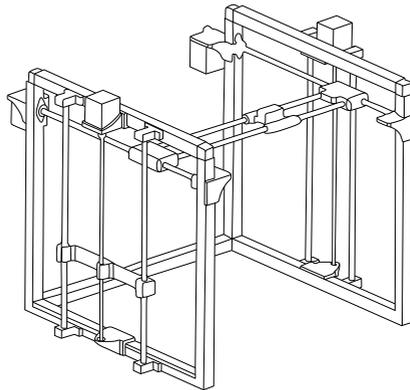


Рис. 1.9. Пример параллельного робота

### 1.3.4 SCARA

SCARA (Selective Compliance Articulated Robot Arm) имеют кинематику, имитирующую движение руки в плоскости. Зачастую содержат две вращательные оси для движения в плоскости, а также еще пару осей для (вращательную и призматическую) для подъема и вращения инструмента. Обладают ограниченной областью работы, но могут двигаться на высокой скорости с высокой точностью. Такая кинематика похожа на артикулированные роботы, но за счёт уменьшенного числа степеней свободы достигаются высокие скорости с высокой точностью в ущерб числу доступных координат для позиционирования. Применяются в «плоских» задачах, например, сборка или перемещение на плоскости с высокой скоростью и точностью. Пример SCARA робота представлен на рисунке 1.10.

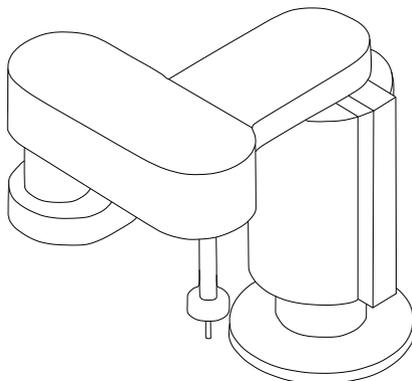


Рис. 1.10. Пример SCARA робота

### 1.3.5 Коллаборативные роботы

Коллаборативные роботы (чаще всего, на основе артикулированных) обладают специальным набором аппаратных и программных средств, позволяющих использовать таких роботов совместно с человеком без необходимости устанавливать специальные защитные устройства, а также упрощающие процесс автоматизации производства. Применяется для облегчения труда рабочих и повышения точности производи-

мых операций в задачах, где требуется гибкость перенастройки или где полная автоматизация процесса приведет к чрезмерному усложнению и удорожанию процесса.

### 1.3.6 Мобильные роботы

Мобильные управляемые платформы и мобильные роботы могут перемещаться в пространстве при необходимости чрезвычайной гибкости производства. Оснащены цифровыми камерами и/или датчиками (оптическими, звуковыми, лазерными) для обеспечения безопасности, картирования местности и навигации. Зачастую требуют специальной подготовки помещений для обеспечения бесперебойной работы. Пример мобильного робота представлен на рисунке 1.11.

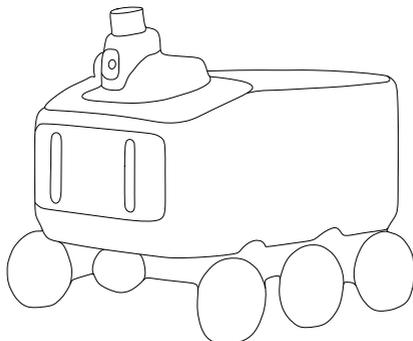


Рис. 1.11. Пример мобильного робота

### 1.3.7 Экзоскелеты

Экзоскелеты чаще всего применяются как закрепляемые на теле человека поддерживающие устройства. Спроектированы для облегчения нагрузок на тела рабочих, позволяют переносить большие массы без вреда для здоровья. Используются на складах для разгрузки и переноса тяжелых разнородных грузов на большие расстояния. Пример экзоскелета представлен на рисунке 1.12.

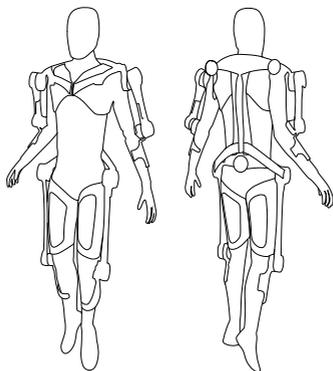


Рис. 1.12. Пример экзоскелета

## 1.4 Захватные устройства

Важным элементом промышленных роботизированных систем являются захватные устройства. Они закрепляются на установочный фланец промышленного робота для носимого инструментария. Захватные устройства используются для захвата и надежного удержания объектов, с которыми робот выполняет технологические операции.

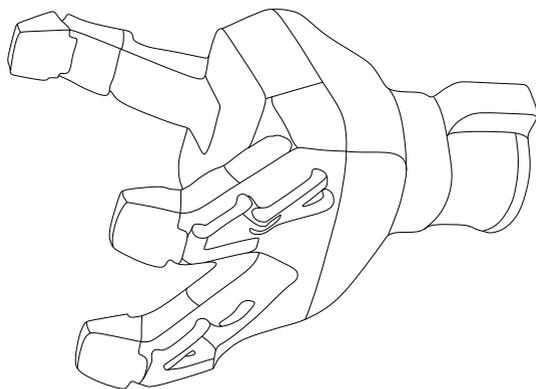


Рис. 1.13. Электромеханическое захватное устройство

По принципу действия могут быть электромеханические захватные устройства; приводятся в действие за счёт управления установленным электроприводом. Могут иметь несколько степеней свободы и возможность точного управления положением пальцев для надежного удержания объектов сложной формы. Пример электромеханического захватного устройства представлен на рисунке 1.13.

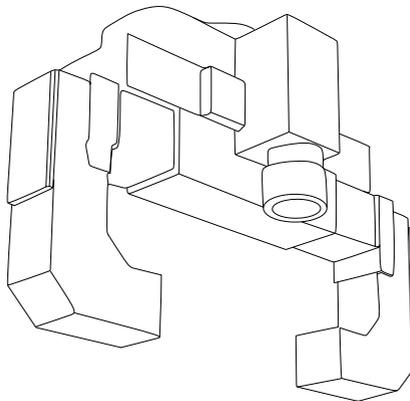


Рис. 1.14. Пневматическое захватное устройство

Пневматические и гидравлические устройства управляются посредством пневмо- или гидросистемы, соответственно. Как правило, имеют два крайних положения (открыто и закрыто). Обладают повышенной скоростью срабатывания и грузоподъемностью, однако требуют установки дополнительного оборудования пневмо- либо гидросистемы. Это оборудование может быть использовано одновременно для нескольких роботов для поддержания давления в общей системе, но каждый робот сможет управлять своим захватным устройством. Пример таких захватных устройств представлен на рисунке 1.14.

Магнитные и вакуумные захватные устройства зачастую применяются для переноса листовых материалов. Если листовая материал имеет большую площадь и подвержен сгибанию, то применяются захватные устройства с несколькими активными элементами для распределения нагрузки по большой площади. Имеют два активных состояния (удерживающее и неудерживающее). Пример вакуумного захватного устройства представлен на рисунке 1.15.

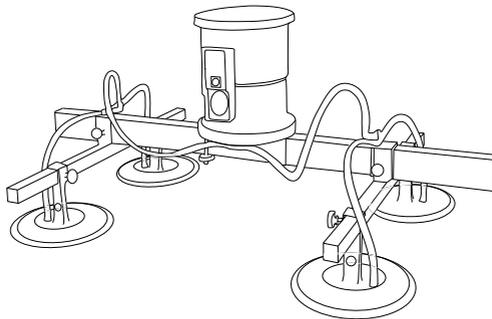


Рис. 1.15. Вакуумное захватное устройство

В особых задачах удержания хрупких объектов без сдавливания применяются эластично-деформируемые захватные устройства. В качестве пальцев в таких устройствах применяются эластичные элементы, которые могут деформироваться при наличии давления пневмосистемы, обеспечивая обхватывающее удержание хрупких объектов. За счёт возможности деформироваться, такие пальцы не раздавливают объект удержания, однако, обладают низкой грузоподъемностью. Пример эластично-деформируемого захватного устройства представлен на рисунке 1.16.

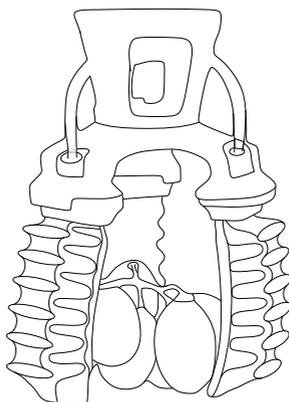


Рис. 1.16. Эластично-деформируемое захватное устройство

Рассмотрим другие характеристики захватных устройств. По типу базирования бывают: центрирующие (удерживаемая деталь при захвате ориентируется соосно оси удержания устройства), базирующие (удерживаемая деталь при захвате ориентируется по плоскости устройства), фиксирующие (удерживаемая деталь при захвате не изменяет ориентации в пространстве). По характеру крепления бывают: несменяемые (жестко фиксируются на установочный фланец робота) и быстросъемные (оснащены механизмом для автоматической или ручной быстрой замены инструмента). По виду управления бывают: неуправляемые (например, постоянные магниты), с бинарным управлением (два возможных состояния), с настраиваемым управлением (возможно точно задавать положение), с адаптивным управлением. Адаптивные захватные устройства принимают форму захватываемой детали для надежного удержания переносимой детали. Адаптивность может достигаться за счёт механики конструкции захватного устройства или программно на основе данных с датчиков и управления с обратной связью.

## 1.5 Области применения

### 1.5.1 Классические области применения

Промышленные робототехнические комплексы зарекомендовали себя для решения таких задач, как перемещение тяжелых грузов и формирование паллет на производствах. Роботы осуществляют данные операции с высокой скоростью работы и высокой точностью позиционирования. Работа с тяжелыми грузами потенциально опасна для здоровья человека, если её выполнять вручную. Ручной труд требует частой смены персонала из-за тяжелой работы. Роботизированная система способна решать эту задачу без остановки.

Другой областью, где часто применяются робототехнические комплексы, является автоматизация сварочных операций. Преимуществом промышленных роботов является возможность позиционирования сварочного инструмента по шести координатам с высокой повторяемостью, что обеспечивает прогнозируемое качество продукции на выходе в течение долгого времени. При ручном выполнении подобных операций предприятие может сталкиваться со снижением качества продукции к концу смены, вызванное усталостью персонала. Роботизация

подобной задачи позволяет снизить количество брака.

Промышленные роботы показали свою эффективность в металлургии в задачах автоматизации работы при высоких температурах и переноса тяжелых заготовок. Данные виды операций могут быть опасны для здоровья людей, что может привести к чрезвычайным ситуациям на производстве и простоям. Роботизация даёт прогнозируемые повторяемость и скорость работы, что в итоге влияет на продуктивность.

Также робототехнические системы применяют для автоматизации операций покраски на предприятиях для получения продукции прогнозируемого качества в прогнозируемые сроки.

### **1.5.2 Новые области применения промышленных роботов**

По мере развития информационных технологий промышленных роботов стали объединять в информационные сети с другим сложным оборудованием на предприятиях для автоматизации операций без участия человека. Так начали применяться практики объединения в единую сеть ЧПУ станков и промышленных манипуляционных роботов. ЧПУ станки обладают более высокой точностью позиционирования режущего инструмента, чем манипуляторы, позволяя изготавливать детали с высокой точностью. Манипуляторы, в свою очередь, выполняют операции по загрузке исходных заготовок в станок и выгрузке готового изделия для передачи на склад в автоматическом режиме. Для функционирования подобной системы разрабатывается специальное программное обеспечение, которое по локальной сети взаимодействует с системами управления станка и манипулятора, взаимодействует с системой склада для формирования плана продукции, осуществляет запуск изготовления деталей и процесса загрузки/выгрузки.

Стали применяться модульные робототехнические ячейки. Такие системы представляют собой мобильную ячейку, содержащую промышленный робот и всю необходимую оснастку. Такая система может быть полезна для малого бизнеса, который не может себе позволить закупку нескольких роботов, и, в тоже время, не обладает объемами производства для полной загрузки робота только на одном участке. Использование модульной ячейки позволяет перемещать один промышленный робот между различными частями производства для автоматизации отдельных участков.

Промышленные роботы дополняются системами технического зрения для автоматизации задач поиска желаемых объектов в рабочей области, обнаружения препятствий при перемещении грузов, контроля качества конечных изделий и отбраковки некачественных единиц.

Активно внедряются коллаборативные роботы на производствах для автоматизации задач с участием человека. Коллаборативные роботы могут применяться в так называемых задачах «ручного обучения», когда сложная траектория обработки детали сначала демонстрируется человеком с применением нужного инструмента, сохраняется в виде алгоритма, а затем повторяется уже роботом. Другим примером использования коллаборативных роботов является задача разгрузки человека при удержании тяжелого инструмента и работе в труднодоступных местах. В труднодоступном месте может быть затруднительно составить алгоритм, который мог бы автоматизировать операцию, либо автоматизация может привести к удорожанию итогового продукта. В этом случае, позиционирование инструмента осуществляется путём наведения в нужное место человеком, но ему не придётся самостоятельно удерживать тяжелый инструмент, так как он закреплен на коллаборативном роботе.

Широкое распространение получают мобильные роботы для автоматизации задач перемещения грузов и манипуляторов на большие расстояния, например, в пределах складов. Можно составить программное обеспечение таким образом, чтобы мобильный робот напрямую взаимодействовал с системой склада и формировал список задач для перемещения. Такие системы зачастую оснащаются системами автономной навигации для перемещения без участия человека с построением необходимых траекторий и обходом препятствий.

Для ряда задач целесообразно объединять несколько мобильных роботов в группы для формирования централизованного управления. Такой подход позволяет одновременно перемещать обширную номенклатуру производимых предприятием деталей. В рамках подобной системы можно предусмотреть возможность быстрой замены неисправного робота другим.

### **1.5.3 В каких процессах используют роботов**

По данным аналитического агентства IFR за 2019 год [6], в мире представлено следующее распределение областей применения промышленных роботов (рисунок 1.17):

- автомобильная промышленность (29%);
- электроника (25%);
- металлургия (11%);
- пластиковая и химическая промышленность (5%);
- пищевая промышленность (3%);
- прочее (27%).

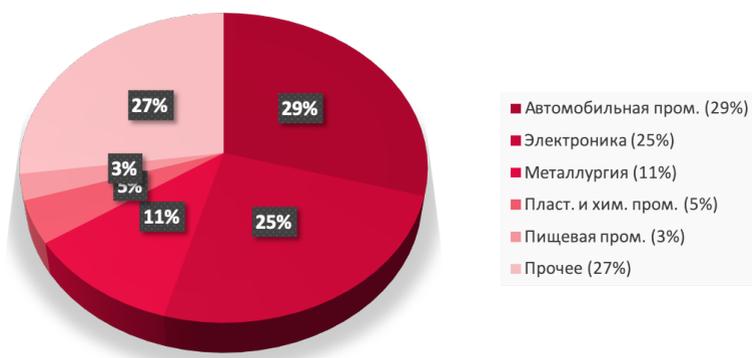


Рис. 1.17. Распределение областей применения промышленных роботов

По данным агентства [6], роботы применяются в процессах, требующих:

- снижения стоимости;
- улучшения качества;
- повышения продуктивности;
- достижения гибкости производства;
- обеспечения безопасности рабочих;
- применения новых технологий;

- применения нового оборудования.

По их же сведениям, типовые операции, требующие роботизации:

- перенос объектов;
- сварка металлов;
- сборка устройств;
- дозирование веществ;
- операции в «чистой комнате»;
- обработка материалов.

#### 1.5.4 Влияние роботизации

Для роботизации производств на сегодняшний день существует ряд барьеров.

В первую очередь, это неявная ценность для бизнеса. Существует сложность расчета экономической выгоды от автоматизации некоторых сложных технологических процессов, что требует четкого понимания стоимости каждого шага производства.

Во-вторых, существуют проблемы на рабочих местах. По мере цифровизации и роботизации производства повышаются требования к сотрудникам (рабочим). Без профессионального развития персонала невозможна дальнейшая автоматизация предприятия.

В-третьих, остро стоит вопрос информационной безопасности. Автоматизация, цифровизация и роботизация предприятия влекут за собой необходимость введения жестких правил поддержания инфраструктуры предприятия и обеспечения информационной безопасности.

Кроме того, для корректного внедрения промышленных роботов на производство необходима точная формализация технологического процесса, которая включает в себя следующие шаги:

1. определение объектов манипулирования;
2. определение подвижных и неподвижных объектов;
3. определение типов операций;

4. определение требуемых инструментов;
5. определение параметров операций;
6. определение предельных значений требований.

Итогом выполнения данных шагов должно стать представление технологического процесса без привязки к существующему оборудованию для возможности пересмотра последовательности операций. Применение промышленных роботов даёт преимущество увеличения числа степеней свободы, применяемых на производстве, что может давать преимущества, например, по скорости изготовления или по сложности производимых деталей.

Внедрение робототехники на производство может также повлиять на производственные процессы.

Во-первых, возникает необходимость формального описания технологического процесса, составления ключевых параметров операций. Определяется экономическая целесообразность роботизации процесса, присваивается приоритет. Характеристики процесса должны быть задокументированы, объединены с описанием других подобных процессов для проведения возможных операций в будущем.

Во-вторых, возникает необходимость создания/модернизации цифровой инфраструктуры предприятия, объединения нескольких технологических операций в единое информационное пространство. Кроме того, на площадках производства устанавливается дополнительное оборудование, необходимое для функционирования робототехнического комплекса.

В-третьих, требуется подготовка помещения для обеспечения безопасности рабочих вблизи работающих промышленных роботов, такие как защитные ограждения, датчики присутствия, кнопки аварийного останова, средства оптического и звукового оповещения.

В-четвёртых, автоматизация, цифровизация и роботизация предприятия влекут за собой необходимость введения жестких правил обеспечения информационной безопасности. При несоблюдении правил информационной безопасности возможно вторжение извне в цифровую сеть предприятия с последующим нанесением урона срокам выполнения процессов, оборудованию и здоровью людей.

В-пятых, по мере цифровизации и роботизации производства повышаются требования к сотрудникам (рабочим). Необходимо приобре-

тение знаний, умений, навыков для обслуживания и взаимодействия с робототехническими комплексами. Без профессионального развития персонала невозможна дальнейшая автоматизация предприятия.

## 1.6 Контрольные вопросы

1. Какие существуют основные характеристики промышленных роботов?
2. Какие существуют виды кинематических пар?
3. Какие шаги формализации технологического процесса необходимо выполнять?
4. Как влияет на производство необходимость описания процессов для роботизации?
5. Как влияет на производство необходимость модернизации инфраструктуры для роботизации?
6. Как влияет на производство необходимость соблюдения мер техники безопасности для роботизации?
7. Как влияет на производство необходимость соблюдения мер цифровой безопасности для роботизации?
8. Как влияет на производство необходимость повышения квалификации сотрудников для роботизации?

## Глава 2

### Основы взаимодействия с промышленным роботом

Лабораторные работы в ходе курса «Программное обеспечение промышленных роботов» выполняются на лабораторной установке, включающей промышленного робота компании Mitsubishi серии Melfa, модель RV-3SDB (рисунок 2.1). Робот является неколлаборативным, оснащён защитным ограждением и кнопками аварийной остановки для обеспечения безопасности.



Рис. 2.1. Промышленный робот для лабораторных работ

В данной главе представлено описание органов управления и интерфейсов, необходимых для взаимодействия с робототехнической установкой.

## 2.1 Основные органы управления

### 2.1.1 Кнопки аварийной остановки

При возникновении любой чрезвычайной ситуации в работе робота необходимо прекратить его работу путем нажатия красной аварийной кнопки. Установлено три аварийные кнопки: одна на пульте управления, одна на панели ячейки робота, одна на панели контроллера. Пример кнопки представлен на рисунке 2.2.



Рис. 2.2. Кнопка аварийной остановки на панели ячейки

### 2.1.2 Элементы управления в ручном режиме

Для управления роботом в ручном режиме используется пульт дистанционного управления. Изображение пульта представлено на рисунке 2.3.

Контроллер в режим ручного управления переводится путем поворота ключа на панели контроллера в положение «Manual». Ключ можно



Рис. 2.3. Пульт дистанционного управления

извлекать для предотвращения использования робота на производстве в ручном режиме неквалифицированным персоналом. Только персонал, прошедший соответствующую подготовку должен иметь доступ к ключу переключения режимов.

Пульт управления активируется путем нажатия кнопки «ТВ ENABLE» на торце пульта управления. Кнопка должна загореться. В автоматическом режиме пульт должен быть деактивирован для предотвращения вмешательства в работу алгоритма. В случае активации пульта в автоматическом режиме сработает сигнализация контроллера.

На пульте присутствует кнопка активирования двигателей робота. Для активации необходимо нажать и далее удерживать желтый переключатель на задней стороне пульта в первом положении. Нажать кнопку «Servo» на пульте; раздастся характерный щелчок запуска двигателей. При отпуске или нажатии во второе положение желтого пере-

ключателя на задней стороне двигателя экстренно остановятся.

Для управления положением робота в декартовых координатах необходимо нажать кнопку «Jog», откроется меню с координатами. На экране отобразится кнопка «XYZ» - нажать под экраном физическую кнопку «F1»; убедиться, что сверху экрана появился режим «CURRENT XYZ». Робот приводится в движение кнопками «X, Y, Z, A, B, C». Скорость регулируется кнопками «OVRD».

Для управления захватом необходимо нажать кнопку «Hand». Закрытие и открытие осуществляется кнопками «-C, +C». При возвращении в режим «Jog» запоминается последний режим — кнопка «XYZ» будет недоступна, т.к. режим будет уже активен.

### 2.1.3 Элементы управления в автоматическом режиме

При использовании робота в автоматическом режиме пульт дистанционного управления должен быть деактивирован. Деактивировать пульт управления путем нужно нажатия кнопки «TB ENABLE» на торце пульта управления. Кнопка должна погаснуть.

В автоматическом режиме двери роботехнической ячейки должны быть закрыты, так как они оснащены датчиками открытия системы безопасности. Закрыть двери ячейки робота можно только с помощью ключа.



Рис. 2.4. Панель контроллера робота

Для получения возможности запуска программ необходимо переве-

сти контроллер в режим автоматического управления путем поворота ключа на панели контроллера в положение «Automatic». Фотография панели контроллера робота представлена на рисунке 2.4.

Запуск программ производится из интерфейса программирования на компьютере.

## 2.2 Работа с интерфейсом программирования на компьютере

Рабочее окно студии разработки для промышленного робота серии Melfa Basic представлен на рисунке 2.5. Окно интерфейса разделено на пять областей:

1. симуляция робота;
2. список файлов;
3. код программы;
4. таблица с сохраненными точками;
5. системные сообщения.

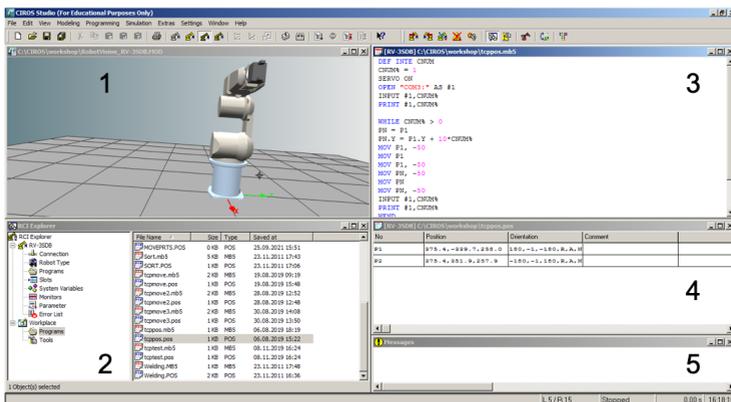


Рис. 2.5. Рабочее окно студии разработки

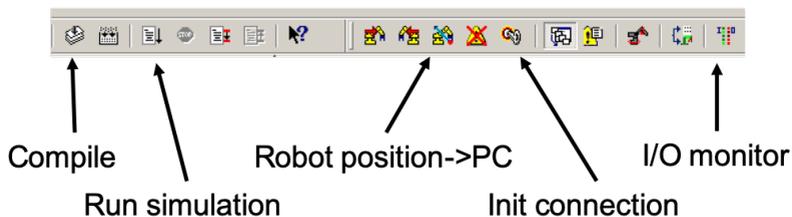


Рис. 2.6. Расположение кнопок на панели инструментов

Порядок написания программы следующий. Устанавливается связь с роботом нажатием кнопки «Init Connection» на панели инструментов. Должно появиться окно с названием робота, соглашаемся нажатием кнопки «ОК». Определяем и сохраняем важные точки в таблице. Для этого в ручном режиме подводим робота к желаемой точке. В интерфейсе на панели инструментов нажимаем кнопку «Robot position -> PC» - симуляция робота должна переместиться в такую же конфигурацию. Расположение необходимых кнопок на панели инструментов показано на рисунке 2.6

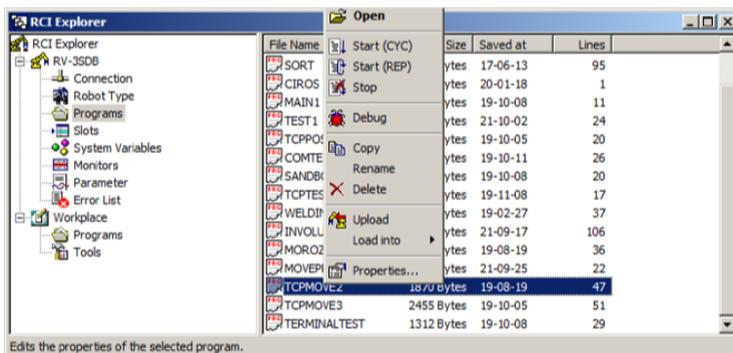


Рис. 2.7. Окно запуска программы на контроллере робота

В таблице точек нажимаем правой кнопкой мыши в пустом месте, выбираем «Insert position». Для сохранения новых значений необходимо выбрать пункт таблицы (он должен подсветиться), нажать по нему

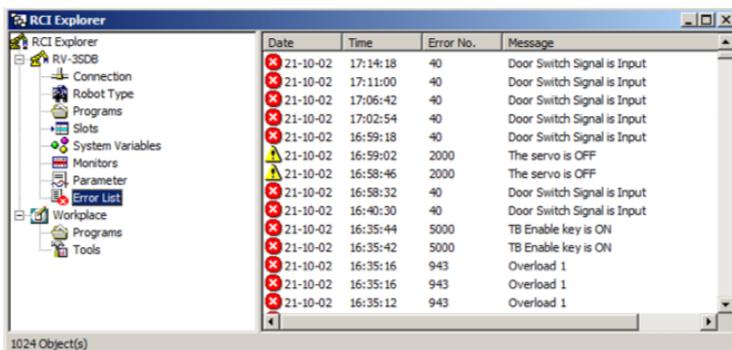


Рис. 2.8. Окно списка ошибок при выполнении программ на контроллере робота

правой кнопкой мыши и выбрать «Асерт position». Так сохраняются все важные для программы точки.

Затем пишется код программы в соответствующем файле. Код проверяется на ошибки нажатием кнопки «Compile» на панели инструментов. В окне системных сообщений проверяется количество ошибок компиляции «error». Запускается симулятор робота нажатием кнопки «Start». Если симуляция прошла успешно, файлы можно загружать на робота.

В окне файлов есть два основных раздела: «RV-3SDB» - файлы робота, «Workplace» - локальные файлы на компьютере. Для загрузки файлов на робота необходимо перейти «Workplace/Programs», выделить сразу два файла программы, например, «MOVEPRTS.MB5, MOVEPRTS.POS», нажать правой кнопкой мыши, выбрать пункт «Download», согласиться с системными сообщениями. Для запуска программы на роботе необходимо перевести робота в автоматический режим, перейти в окне файлов в раздел «RV-3SDB/Programs», выбрать программу «MOVEPRTS», нажать правой кнопкой мыши, выбрать пункт «Start (CYC)». Пример внешнего вида окна при запуске программы показан на рисунке 2.7. Для остановки программы можно нажать правой кнопкой мыши, выбрать пункт «Stop», либо нажать кнопку «STOP» на пульте управления.

В случае возникновения ошибок при выполнении программы на ро-

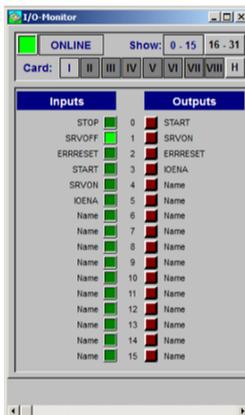


Рис. 2.9. Окно списка сигналов портов ввода-вывода

боте, данные сообщения можно прочитать на главном экране пульта дистанционного управления робота, либо выбрать в студии раздел «Error list». Пример окна со списком ошибок представлен на рисунке 2.8. Список ошибок сопровождается датой и временем появления, описанием, типом сообщения (системная ошибка или предупреждение).

Отслеживание состояний сигналов портов ввода-вывода осуществляется в окне «I/O Monitor» (рисунок 2.9). В строке «Card:» выбирается номер карты расширения, установленной в контроллер. Тёмно-серыми кнопками указываются свободные слоты. Светло-серыми кнопками показаны карты, доступные для выбора. Ниже, в центре окна указываются номера портов. В левой колонке «Inputs» показывается логическое значение на порте ввода, темные значения означают логический 0, светлые означают логический 1. В колонке «Outputs» нажатием на кнопки можно задавать выходные значения на соответствующих портах вывода.

Для использования сетевых возможностей в алгоритме необходимо указать список интерфейсов в окне настройки системного параметра COMDEV. Доступные интерфейсы определяются установленными картами расширения. В свойствах системного параметра указываются через запятую все интерфейсы, которые в алгоритме будут привязаны к системным переменным COM1-8. На рисунке 2.10 показан пример настройки, при котором переменной COM1 соответствует интерфейс

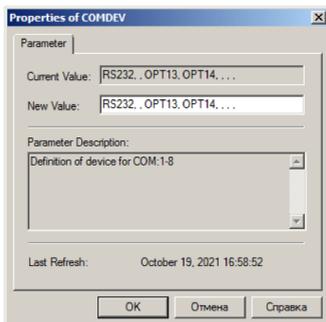


Рис. 2.10. Окно настройки системного параметра COMDEV

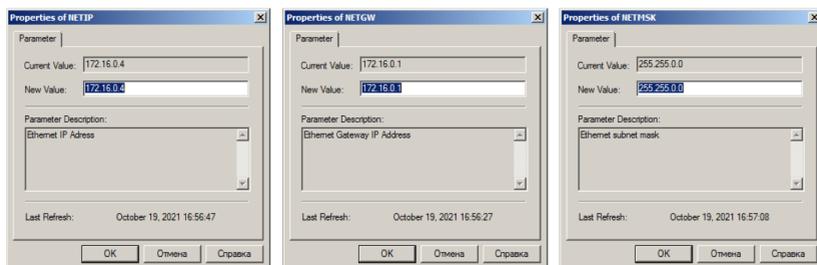


Рис. 2.11. Окна настройки сетевых параметров контроллера.

RS232, COM3 соответствует OPT13, COM4 — OPT14. Остальные переменные COM не привязаны ни к одному интерфейсу, им соответствуют пустые запятые в свойствах настройки.

Интерфейс TCP/IP настраивается в окнах системных параметров NETIP, NETGW и NETMSK для задания, соответственно, IP адреса контроллера, адрес маршрутизации, маска подсети (рисунок 2.11). Данные значения также должны быть идентично внесены в настройки студии разработки, окно Communication Port (рисунок 2.12).

Для взаимодействия по TCP/IP необходимо также настроить входящие сетевые параметры (рисунок 2.13). В окне NETHSTIP указываются IP адреса компьютеров в локальной сети, от которых будут разрешены входящие сообщения. В окне NETPORT указываются номера портов, которые будут открываться для подключений. В окне NETMODE ука-

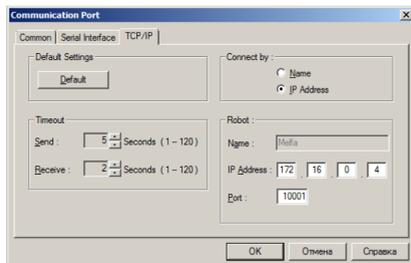


Рис. 2.12. Окно настройки параметров подключения студии разработки

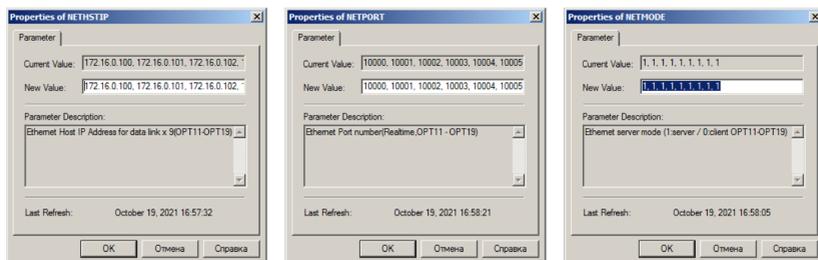


Рис. 2.13. Окна настройки входящих сетевых параметров

зывается роль контроллера для каждого подключения, возможны варианты «сервер» (значение по умолчанию) или «клиент». Для каждого из указанных трех параметров указывают параметры через запятую, они будут соответственно применены для виртуальных интерфейсов OPT11–OPT19.

## 2.3 Контрольные вопросы

1. Для чего предназначен пульт настройки и управления промышленного робота?
2. Для чего предназначена аварийная кнопка?
3. Для чего предназначен переключатель безопасности на пульте?

4. Какое назначение переключателя режимов работы на промышленном контроллере?
5. Как производится настройка сетевых параметров контроллера?
6. Как осуществляется проверка портов ввода-вывода промышленного контроллера?

# Глава 3

## Синтаксис языка программирования Melfa Basic

Лабораторные работы в ходе курса «Программное обеспечение промышленных роботов» выполняются на лабораторной установке, включающей промышленного робота компании Mitsubishi серии Melfa, с применением языка программирования Melfa Basic. В данной главе представлен синтаксис языка, необходимый для выполнения лабораторных работ и составления программ автоматизации различных технологических процессов.

### 3.1 Типы переменных

Переменная с указанием типа создаётся с помощью команды DEF, далее через пробел указывается обязательный параметр с типом создаваемой переменной и параметр имени.

Основными типами переменных являются:

DEF INTE WORK — целочисленные переменные (+/-32767);

DEF FLOAT SMALLSIM — переменные типа с плавающей точкой (+/-1.7E+38);

DEF DOUBLE BIGSIM — переменные типа с плавающей точкой (+/-1.7E+308);

DEF POS PHELPH — переменные положения в прямоугольной системе координат.

Присваивания значения переменной осуществляется с использованием знака равенства:

```
WORK = 100
```

Для задания значения положения в прямоугольной системе координат необходимо указать в качестве параметра матрицу с шестью элементами, равными, соответственно, значениям координат X, Y, Z, A, B, C, даже если они нулевые:

```
PHLPH = (+75.00,+0.00,+0.00,+0.00,+0.00,+0.00)
```

В синтаксисе языка Melfa Basic предусмотрен краткий формат создания переменных без использования команды DEF. Если первая буква имени переменной содержит символ I, то будет создана переменная целочисленного типа; если P, то будет создана переменная положения в прямоугольной системе координат:

```
IX = 2
```

```
PY = (0,-100,0,0,0,0)
```

## 3.2 Алгоритмические конструкции

### 3.2.1 Команды начала и завершения работы

Перед началом движения робота необходимо выполнить команду включения электроприводов, часто указывается в начале программы:

```
SERVO ON
```

Регулирование значения максимальной угловой скорости, развиваемой роботом, осуществляется в процентах от максимальной, может быть указано в начале программы и в середине кода по необходимости регулирования скорости:

```
JOVRD 100
```

Регулирование значения максимальной линейной скорости, развиваемой роботом, осуществляется в процентах от максимальной:

```
SPD 100
```

В конце программы часто указывается команда отключения приводов:

```
SERVO OFF
```

В процессе выполнения алгоритмов также можно использовать команды выключения и включения двигателей, например, в случаях долгих простоев робота (например, в периоды долгого ожидания команды от других устройств).

Выполнение алгоритма завершается соответствующей командой, контроллер робота переходит в режим ожидания:

```
END
```

### 3.2.2 Условия

Условие создаётся с помощью операторных скобок `IF ... THEN`, между которыми указывается условие истинного значения. После команды `THEN` перечисляются команды, которые будут выполняться в случае истинности условия. Далее следует команда `ELSE`, после которой перечисляются команды для использования в случае невыполнения условия. Завершается конструкция условия командой `ENDIF`. Возможно создание неполного условия с перечислением команд только истинного условия, тогда команда `ELSE` не указывается, конструкция закрывается командой `ENDIF`. Пример использования конструкции полного условия:

```
IF M1=1 THEN
    MOV P1
ELSE
    MOV P2
ENDIF
```

Пример использования конструкции неполного условия (команда `MOV P2` будет выполнена в любом случае):

```
IF M1=1 THEN
    MOV P1
ENDIF

MOV P2
```

### 3.2.3 Циклы

Цикл с использованием итеративной переменной создаётся с помощью команды FOR, далее указывается имя переменной и границы итераций (включая оба значения границ). После указываются команды, которые будут выполняться в цикле. Переменную итерации можно использовать для проведения вычислений внутри цикла. Завершается конструкция цикла командой изменения значения переменной NEXT, после которой пишется имя переменной итерации. Пример создания цикла, который будет повторять движения три раза (итерации 0, 1 и 2):

```
FOR I1=0 TO 2
    MOV P1
    MOV P2
NEXT I1
```

Цикл с предусловием создаётся с помощью команды WHILE, далее указывается условие выхода из цикла. На последующих строках перечисляются команды для повторения в цикле. Завершение цикла осуществляется командой WEND, в которой происходит повторная проверка условия выхода из цикла. Для корректной работы цикла с предусловием необходимо внутри реализовать изменение значений, используемых в проверке условия. Пример использования цикла с предусловием:

```
WHILE M1>0
    . . .
WEND
```

Также возможна реализация бесконечного цикла. Прерывание такого цикла возможно внутренним условием с командами завершения работы или выхода из цикла. Возможно построение алгоритма с выходом из бесконечного цикла по внешнему событию или прерыванию. Пример использования бесконечного цикла:

```
WHILE 1
    . . .
WEND
```

Принудительный выход из цикла осуществляется командой BREAK. Часто используется вместе с условиями проверки переменных. Пример использования команды выхода из цикла:

```
WHILE M1 <> 0
  IF M2 = 0 THEN BREAK
WEND
```

### 3.2.4 Процедуры

Простейшим способом перехода к определенной последовательности действий является использование команды GOTO, после которой через пробел указывается стока кода, которую необходимо выполнить. Требуемую строку необходимо сначала пометить соответствующим номером. Рассмотрим пример использования, в данном случае команда MOV P1 не будет выполнена, а MOV P2 будет:

```
70 GOTO 90
80 MOV P1
90 MOV P2
```

При написании кода удобно не нумеровать каждую строку, а использовать текстовые метки. Для указания метки строка должна начинаться с символа «\*» и содержать текстовое название. В данном случае, после команды GOTO указывается название текстовой метки. Рассмотрим пример, аналогичный приведенному выше:

```
GOTO *L1
MOV P1
*L1
MOV P2
```

При описании в алгоритме последовательности действий, которые должны исполняться в разные моменты времени, применение команды GOTO может быть затруднительно, так как надо возвращаться в разные участки кода. Для описания структур процедур следует применять команду GOSUB для перехода к нужной последовательности действий с созданием точки вызова и возможностью к ней вернуться. После команды указывается через пробел название текстовой метки процедуры. Возврат к точке вызова осуществляется командой RETURN. Пример программы, в которой будут последовательно выполнены точки P1, P11, P12, P2:

```
SERVO ON
```

```
MOV P1
GOSUB *L1
MOV P2
SERVO OFF
END

*L1
    MOV P11
    MOV P12
RETURN
```

Для команд GOTO и GOSUB возможно создание конструкций по условию с выбором из нескольких вариантов с использованием команды ON. Рассмотрим пример использования команды:

```
ON M1 GOSUB *L1, *L6
```

В данном случае M1 является переменной, которая будет проверяться на условия. Если M1=1, то произойдёт переход к процедуре \*L1. Если M1=2, то произойдёт переход к процедуре \*L6. Если M1=0, то произойдёт переход к следующей строке. При вызове команды GOSUB создаётся точка вызова для команды RETURN. Вместо команды GOSUB можно также использовать в данной конструкции команду GOTO без создания точки вызова.

### 3.2.5 Функции

Для сокращения описания часто используемых математических операций можно создавать функции и обращаться к ним по имени. Для создания функции используется команда DEF с указанием имени функции, параметрами функции в скобках и выражением для расчёта значения после знака равно. Рассмотрим пример создания математической функции расчёта среднего значения двух чисел:

```
DEF FNMAVE (MA, MB) = (MA+MB) / 2
```

где FNMAVE — имя функции, (MA, MB) — параметры функции, (MA+MB) / 2 — операция функции.

Рассмотрим пример, в котором создаётся функция, присваиваются значения двум переменным и вычисляется их среднее значение:

```
DEF FNMAVE (MA, MB) = (MA+MB) / 2
MDATA1=20
MDATA2=30
MAVE=FNMAVE (MDATA1, MDATA2)
```

### 3.2.6 Прерывания и таймеры

Прерывание — это события, которые прекращают выполнение основного кода алгоритма программы и переходят к выполнению специального кода, описанного для данного события. Могут вызываться по условию, описанному для вызова прерывания. Условие может быть описано для переменной алгоритма, портов ввода-вывода, таймеров. Прерывания используют для контроля данных от внешних устройств, например, кнопок или датчиков, контроля состояния системы, обработки событий по времени.

Объявляется прерывание с помощью команды DEF АСТ. В качестве параметров через запятую указываются порядковый номер прерывания, условие выполнения, затем, через пробел, команда, которую необходимо выполнить в случае истинности условия. В качестве команды выполнения зачастую указывают команду перехода к процедуре обработки прерывания GOSUB. Рассмотрим пример объявления прерывания:

```
DEF АСТ 1, M_IN(10)=1 GOSUB *SUB1
```

Здесь DEF АСТ 1 — объявление прерывания с порядковым номером 1, M\_IN(10)=1 — условие срабатывания прерывания, GOSUB \*SUB1 — команда перехода к процедуре \*SUB1 в случае срабатывания условия.

Прерывания в языке MELFA BASIC создаются неактивированными. Это означает, что в памяти контроллера создаётся условие вызова прерывания, однако, оно не будет фоном проверяться при выполнении основного кода программы. Для проверки условия прерывание необходимо активировать специальной командой в необходимом месте исходного кода:

```
АСТ 1 = 1
```

Созданное прерывание можно по необходимости деактивировать, если требуется роботом выполнить последовательность действий, которая не может быть остановлена по соображениям безопасности. Например,

если роботом выполняется розлив жидких материалов в формы, выполнение прерывания может повлечь за собой корректное выполнение действий и нанесение ущерба предприятию. Деактивация прерывания осуществляется следующей командой:

```
ACT 1 = 0
```

Вызов прерываний часто совмещают с использованием таймеров для обработки событий системы по времени. Таймеры запускаются и проверяются через системные переменные M\_TIMER, в скобках указывается номер таймера. Рассмотрим пример вычисления времени, необходимого для выполнения двух команд:

```
M_TIMER(1)=0  
MOV P1  
MOV P2  
M1=M_TIMER(1)
```

В данном примере создаётся системный таймер номер 1 начиная с нуля секунд, выполняются две команды, затем в переменную M1 записывается значение времени таймера, затраченное на выполнение двух команд.

В следующем примере рассмотрим случай, когда необходимо вызывать прерывание каждые 10 секунд при выполнении программы. Для этого в коде описывается прерывание, создается таймер и активируется прерывание; при срабатывании прерывания таймер обнуляется:

```
DEF ACT 1, M_TIMER(1)>10 GOSUB *SUB1  
M_TIMER(1) = 0  
ACT 1 = 1  
...  
*SUB1  
    MOV P_RES  
    M_TIMER(1) = 0  
RETURN 0
```

### 3.2.7 Паузы

Приостановка выполнения программы на заданное количество времени осуществляется с помощью команды DLY, далее через пробел в

качестве параметра указывается время приостановки в секундах. Паузы полезно использовать для ожидания завершения действий внешних устройств, не оборудованных сигналами обратной связи. Пример паузы в 0.1 секунды:

```
DLY 0.1
```

В случае наличия внешних сигналов, означающих наступление события, возможно использовать команду WAIT. Данная команда приостанавливает выполнение программы, пока условие, описанное в качестве параметра команды, не станет истинным. При использовании данной команды необходимо организовать внешние события, которые приведут к срабатываю условия, в противном случае, выполнение программы будет приостановлено до отключения контроллера. Данный вид ожидания используют, например, для обработки кнопок начала действия или для ожидания сигнала о завершении выполнения внешнего устройства. Пример использования команды ожидания:

```
WAIT M_IN(1) = 1
```

## 3.3 Команды управления движением

### 3.3.1 Приближённое движение в точку

Перемещение робота в точку из текущего положения с приближением к прямой линии осуществляется командой:

```
MOV P1
```

где P1 — переменная, содержащая координаты прямоугольного пространства в формате (X, Y, Z, A, B, C), привязанные к основанию робота. Перемещение будет осуществляться с приближением к прямой линии, однако, в случае наличия ограничений на кинематику при перемещении, движение может быть представлено в виде дуги для исключения недопустимых положений. Пример движения показан на рисунке 3.1, здесь P\_CURR — текущее положение робота.

Так называемые команды подхода к детали и отхода от детали являются часто используемыми в промышленной автоматизации, поэтому в команде MOV предусмотрен специальный параметр:



Рис. 3.1. Пример использования команды MOV

```
MOV P1, -50
```

где -50 — смещение от исходной точки P1 в системе координат инструмента по оси Z на -50 мм. Направление смещения определяется ориентацией инструмента робота в точке P1. Например, если инструмент сориентирован вниз, то значение -50 мм по оси Z инструмент будет означать смещение положения вверх на 50 мм.

Рассмотрим пример, когда нам в начале программы необходимо из начального положения перейти в точку на 100 мм выше детали (инструмент сориентирован вертикально и смотрит вниз), а затем опустить к положению самой детали в точке P1. Программа будет иметь следующий вид:

```
MOV P1, -100
MOV P1
```

Для описания смещения в прямоугольной системе координат, привязанной к роботу, можно в коде программы создавать вспомогательные переменные координат:

```
RHELPH=(75,0,0,0,0,0)
```

где в скобках записываются декартовы координаты смещения (X, Y, Z, A, B, C) соответственно.

В команде MOV кроме явного указания точек (например, P1) можно проводить математические операции над точками, например сумма переменных для смещения:

```
MOV P1+RHELPH, -50
```

или умножение на переменные для выполнения операций в цикле:

```
MOV P1+RHELPH*I1, -50
```

Логика выполнения подобных операций описывается матричными преобразованиями: при складывании переменных положений складываются соответствующие элементы матрицы координат; при умножении на скаляр, каждый элемент матрицы умножается на него.

Так как исполнение команды движения в точку занимает значительное время, особенно на низких скоростях движения робота, в синтаксисе присутствует дополнительный параметр `WTHIF` команды `MOV`, позволяющий проверять дополнительное условие при выполнении движения. Такая конструкция полезна в случае необходимости прервать движение в точку, например, при поступлении внешнего сигнала, как в примере:

```
MOV P1 WTHIF M_IN(11) = 0, SKIP
```

где `M_IN(11) = 0` — проверяемое условие, `SKIP` — команда перехода к следующей строке в случае истинности выражения.

### 3.3.2 Точное движение в точку

Перемещение робота в точку из текущего положения точно по прямой линии осуществляется командой:

```
MVS P1
```

Данная команда применяется для точного позиционирования инструмента и движения по прямой. Однако, рекомендуется не использовать данную команду в начале программы, так как, если движение из текущего положения в желаемое по прямой линии пересекает области ограничений кинематики, то команда не выполнится и робот остановится с ошибкой. Пример использования команды `MVS` представлен на рисунке 3.2.



Рис. 3.2. Пример использования команды `MVS`

Совместно с командой `MVS` можно использовать вспомогательный параметр и математические операции, по аналогии с командой `MOV`:

```
MVS P1+PHLFX, -50  
MVS P1+PHLFX*11, -50
```

### 3.3.3 Движение по дуге

В языке программирования предусмотрено MELFA BASIC предусмотрено несколько команд движения по дуге. Первая команда MVR содержит три параметра P1, P2, P3.

```
MVR P1, P2, P3
```

При выполнении данной команды робот движется приблизительно по прямой из текущего положения в точку P1, затем движется по дуге, проходящей через все три точки P1, P2, P3. Пример использования команды MVR представлен на рисунке 3.3.

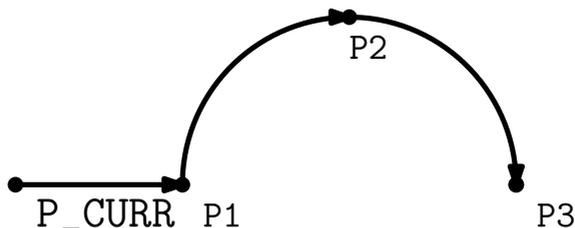


Рис. 3.3. Пример использования команды MVR

Вторая команда MVR2 содержит также три параметра P1, P2, P3.

```
MVR2 P1, P2, P3
```

При выполнении данной команды робот движется приблизительно по прямой из текущего положения в точку P1, затем движется по дуге, проходящей через три точки P1, P2, P3, однако движется от точки P1 до P2, до P3 не доходит. Пример использования команды MVR2 представлен на рисунке 3.4.

Третья команда MVR3 содержит также три параметра P1, P2, P3.

```
MVR3 P1, P2, P3
```

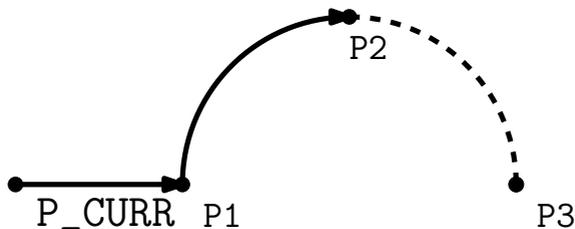


Рис. 3.4. Пример использования команды MVR2

При выполнении данной команды робот движется приблизительно по прямой из текущего положения в точку P1, затем движется по дуге, проходящей через две точки P1, P2, точка P3 является центром радиуса описываемой дуги. Пример использования команды MVR3 представлен на рисунке 3.5.

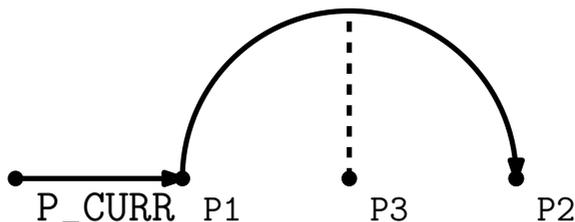


Рис. 3.5. Пример использования команды MVR3

Последняя команда MVC описывает движение по полной окружности и содержит три параметра P1, P2, P3.

MVC P1, P2, P3

При выполнении данной команды робот движется приблизительно по прямой из текущего положения в точку P1, затем движется по полной окружности, проходящей через все три точки P1, P2, P3. Пример использования команды MVC представлен на рисунке 3.6.

В зависимости от доступных для измерений координат движения робота, удобно применять одну из команд MVR, MVR2, MVR3, MVC для решения конкретных задач. Необходимые координаты положений

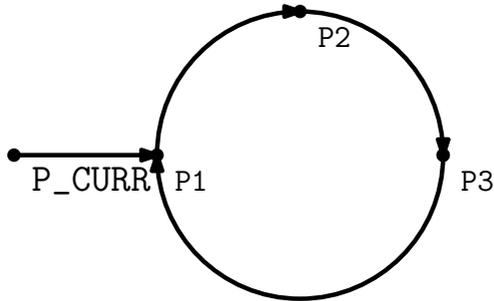


Рис. 3.6. Пример использования команды MVC

могут быть измерены в САПР при наличии модели рабочей области, либо путем подведения инструмента робота к нужным точкам в ручном режиме.

### 3.3.4 Формирование комплексных траекторий

В случае необходимости формирования комплексной траектории следует использовать упомянутые выше команды, стыкуя разные участки между собой. В этом случае, начало следующего участка должно являться окончанием предыдущего. При этом, могут использоваться одинаковые переменные положения для обозначения границ траекторий.

Рассмотрим пример траектории, содержащей последовательно дугу, прямую, дугу, прямую, дугу, прямую линию (рисунок 3.7). В данном случае траектория начинается с дуги (MVR P2, P3, P4), затем переходит в прямую линию (MVS P5), затем снова в дугу (MVR P5, P6, P7), в данной случае точка P5 совпадает для обеих команд MVS и MOV. Код полученной программы будет выглядеть следующим образом:

```
MVR P2, P3, P4
MVS P5
MVR P5, P6, P7
MVS P8
MVR P8, P9, P10
MVS P11
```

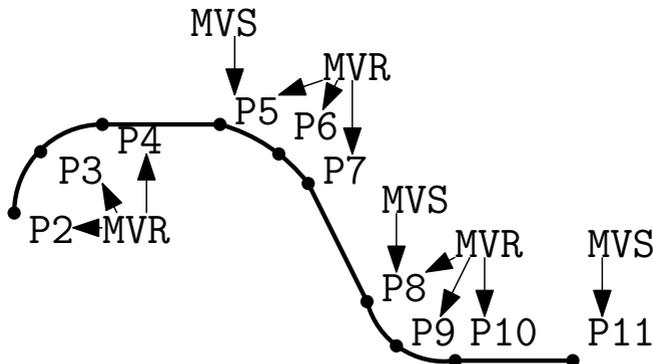


Рис. 3.7. Пример комплексной траектории

## 3.4 Работа с внешними устройствами

Взаимодействие с внешними устройствами осуществляется посредством подключения к платам расширения. Возможна установка плат расширения управления захватными устройствами, управления портами ввода-вывода, сетевых плат.

### 3.4.1 Управление захватными устройствами

Плата управления захватным устройством предусматривает дискретный контроль положения. К одной плате расширения предусмотрена возможность подключения нескольких устройств сразу. В базовой конфигурации настроено задание двух положений: открытого и закрытого. Открытие захватного устройства осуществляется следующей командой:

```
HOPEX 1
```

где через пробел указывается обязательный параметр с номером захватного устройства, даже если на работе в данный момент времени установлено только одно. Пример варианта установки нескольких захватных устройств на одного робота представлен на рисунке 3.8.

Закрытие захватного устройства осуществляется следующей командой:

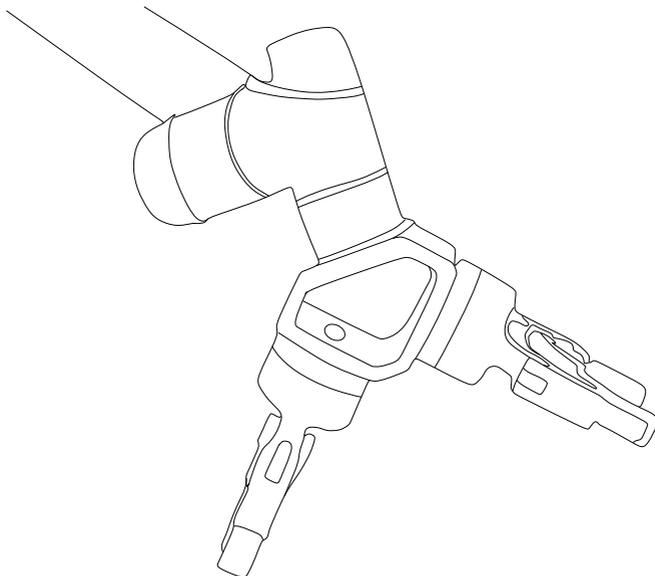


Рис. 3.8. Вариант установки двух захватных устройств

```
HCLOSE 1
```

В силу возможности роботом развивать высокую скорость перемещения, команды взаимодействия с захватным устройством следует совмещать с командами паузы:

```
MOV P1  
DLY 0.1  
HCLOSE 1  
DLY 0.1  
MOV P2
```

где DLY является командой паузы в секундах. Пауза до команды закрытия необходима для того, чтобы робот успел остановиться в нужной точке после выполнения предыдущей команды движения. Пауза после закрытия необходима для того, чтобы захватное устройство успело выполнить необходимые механические движения до начала выполнения команды движения в новую точку. Зачастую 0.1 секунды достаточно

для выполнения данных операций; эти значения пауз могут быть настроены экспериментально.

### 3.4.2 Порты ввода-вывода

Установленные платы расширения позволяют управлять дискретным состоянием портов ввода и вывода. Логическая единица на дискретном порте ввода или вывода соответствует уровню напряжения примерно в 24 вольта, логический ноль соответствует примерно 0 вольт. Управляя состоянием портов можно отправлять дискретные команды на внешние устройства, либо принимать сигналы от внешних устройств.

Считывание значений состояний дискретных портов ввода осуществляется через обращение к системной переменной `M_IN`, далее в скобках указывается номер порта. Значение системной переменной можно использовать в логических конструкциях, например, условиях, циклах с предусловием, задержках и так далее. Перед командой считывания значения состояния порта может потребоваться пауза для остановки движения робота и корректной обработки сигнала, например:

```
DLY 0.1
IF M_IN(2) = 0 THEN
    ...
```

Для отправки дискретного сигнала на порт вывода создаётся переменная, содержащая название порта, системный номер:

```
DEF IO MAGN = Bit, 8
DLY 0.1
MAGN = 1
DLY 0.1
```

где `MAGN` — название переменной, `8` — номер порта. Задание значения на порте вывода осуществляется через присваивание переменной логического значения `MAGN = 1`. Как показано в примере выше, иногда может быть полезно сопровождать паузами команду задания сигнала на дискретном порте вывода, по аналогии с работой с захватными устройствами.

### 3.4.3 Сетевые интерфейсы

Для взаимодействия с сетевыми интерфейсами необходимо в первую очередь открыть их с помощью команды:

```
OPEN "COM1:" AS #1
```

где COM1 — номер интерфейса в конфигурации контроллера робота, #1 — локальный краткий номер интерфейса внутри алгоритма. Тип открываемого интерфейса зависит от системных настроек в памяти контроллера. Дальнейшее обращение к интерфейсу в алгоритме будет осуществляться по краткому номеру. Закрывать интерфейсы можно по номеру:

```
CLOSE #1
```

либо закрыть сразу все интерфейсы:

```
CLOSE
```

Отправка сообщения в открытый интерфейс осуществляется командой PRINT, далее через пробел указываются краткий номер интерфейса, отправляемые значения в пакете. Указывать для отправки можно как имена переменных, так и прямые значения. Пример использования команды отправки сообщения:

```
PRINT #1, "OUTPUT TEST", M1, P1
```

где #1 — краткий номер интерфейса, "OUTPUT TEST" — непосредственно отправляемые значения, M1, P1 — отправляемые переменные.

Входящие сообщения от интерфейса считываются командой INPUT, далее через пробел указываются краткий номер интерфейса и переменные, в которые сохраняются считанные значения:

```
INPUT #1, IX, IY
```

При использовании команды PRINT выполнение программы приостанавливается, пока во входящем буфере не появится сообщение. При этом, если входящее сообщение придёт раньше использования команды PRINT, оно останется в буфере, пока выполнение алгоритма не дойдёт до команд обработки. В языке программирования Melfa Basic предусмотрено считывание входящих сообщений сразу по прерыванию. Создаётся прерывание по входящему сообщению следующей командой:

```
ON COM(1) GOSUB *RECV
```

где COM(1) — номер подключения для прерывания, GOSUB \*RECV — переход к процедуре обработки прерывания \*RECV по входящему событию. Как и для общих прерываний в контроллере, рассмотренных ранее, есть возможность активации и деактивации прерываний по сообщениям. COM(1) ON активирует прерывание по сообщению, COM(1) OFF — деактивация прерывания, COM(1) STOP — приостановка прерывания до следующей команды.

Пример использования команд обработки сообщений сетевых интерфейсов:

```
OPEN "COM1:" AS #1
ON COM(1) GOSUB *RECV
COM(1) ON
MOV P1
COM(1) STOP
MOV P2
...
COM(1) OFF
CLOSE #1
END

*RECV
    INPUT #1, M0001
    OUTPUT #1, P0001
RETURN 1
```

### 3.5 Контрольные вопросы

1. Какие существуют команды движения для языка Melfa Basic V?
2. Какие существуют виды циклов для языка Melfa Basic V?
3. Как осуществляется инициализация интерфейсов подключения для языка Melfa Basic V?
4. Как создаются процедуры для языка Melfa Basic V?
5. Как создаются прерывания для языка Melfa Basic V?

# Глава 4

## Лабораторные работы по курсу «Программирование промышленных роботов»

### 4.1 Лабораторная работа 1 «Преобразования координат и циклы»

#### 4.1.1 Задание на лабораторную работу

Переставить кластер деталей (9 штук) с одного стола (рисунок 4.1) на другой (рисунок 4.2) в цикле, используя только 2 точки в таблице сохраненных точек: одна точка — захват первой детали на первом столе, вторая точка — отпускание детали на втором столе. В коде программы можно использовать циклы, преобразования координат и вспомогательные переменные со смещением по осям  $X$  и  $Y$  75 мм.

#### 4.1.2 Основные команды программирования

При выполнении программы использовать следующие команды программирования языка MELFA BASIC:

END — завершение программы, обязательно размещается в конце файла.

SERVO ON — включение двигателей, размещается в начале программы.

JOVRD 100 — скорость движения в процентах от максимальной, обычно размещается в начале программы.

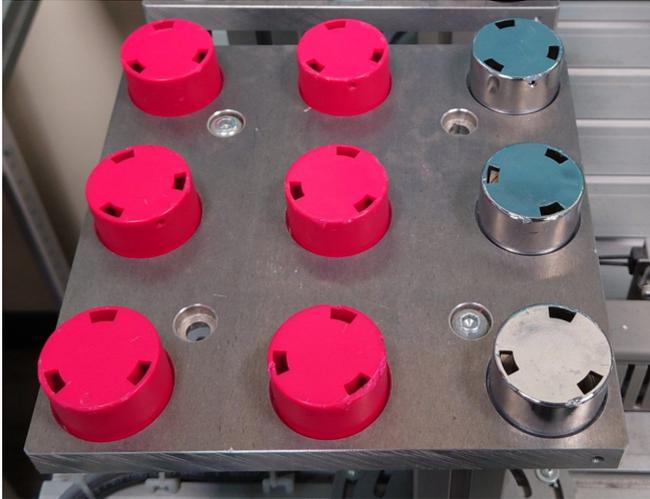


Рис. 4.1. Исходный стол с деталями для лабораторной работы

SERVO OFF — выключение двигателей, размещается в конце программы.

DLY 0.1 — пауза выполнения программы в секундах.

HOPEH 1 — открытие захватного устройства.

HCLOSE 1 — закрытие захватного устройства.

MOV P1 — движение в точку P1 из таблицы сохраненных точек.

MOV P1, -50 — движение в точку P1 со смещением 50 мм вверх по оси Z.

FOR I1= 0 TO 2 — начало выполнения цикла, I1 — переменная итерации цикла.

NEXT I1 — окончание цикла.

Для создания настраиваемого смещения координат можно в коде программы создавать вспомогательные переменные координат:

```
RHELPH=(+75.0,+0.0,+0.0,+0.0, +0.0, +0.0)
```

где в скобках записываются декартовы координаты смещения X, Y, Z, A, B, C соответственно.

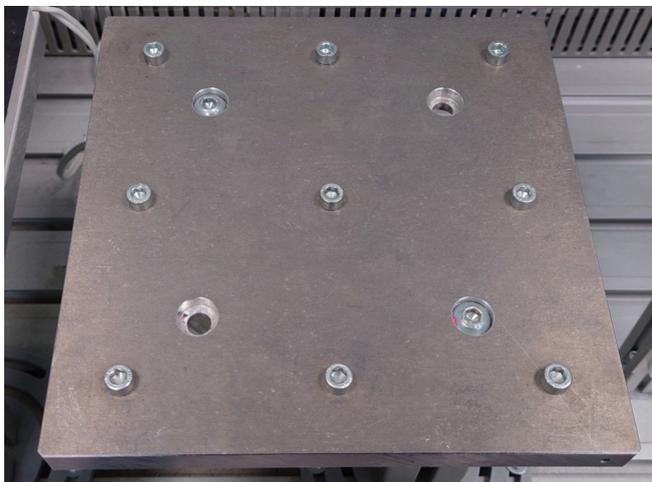


Рис. 4.2. Второй стол для лабораторной работы

В команде MOV кроме явного указания точек (например, P1) можно проводить математические операции над точками, например сумма переменных для смещения:

```
P1+PHELPRX, -50
```

или умножение на переменные для выполнения операций в цикле:

```
MOV P1+PHELPRX*I1, -50
```

### 4.1.3 Порядок выполнения

1. В ручном режиме управления робот определить две точки, сохранить в таблицу. Написать программу, которая открывает захват, зависает над первой точкой, опускается к первой точке, закрывает захват, поднимает деталь, зависает над второй точкой, опускает деталь, открывает захват, поднимает захват над деталью.
2. Дополнить программу, чтобы робот перенес вторую деталь, используя смещение.

3. Дополнить программу, чтобы робот в цикле мог переносить пару деталей со смещением по одной координате.
4. Дополнить программу, чтобы робот в цикле мог переносить кластер деталей со смещением по двум координатам.

#### **4.1.4 Содержание отчета**

1. Титульный лист
2. Цели выполнения работы
3. Код конечной программы
4. Описание команд
5. Таблица сохраненных точек
6. Четыре фотографии, иллюстрирующие этапы выполнения программы
7. Выводы по работе

## **4.2 Лабораторная работа 2 «Порты ввода-вывода и условия»**

### **4.2.1 Задание на лабораторную работу**

Написать программу с бесконечным циклом проверки наличия деталей в ряду из трёх единиц. Манипулятор должен подходить к месту правее предполагаемого расположения детали, по датчику на захватном устройстве (рисунок 4.3) определять наличие детали, в случае наличия детали на месте — подняться на 100 мм и опуститься в исходное положение, переместиться в следующее положение. При запуске программы робот должен стоять неподвижно; бесконечный цикл запускается путем нажатия кнопки “Reset” на пользовательской панели под дверью ячейки. Выход из цикла осуществляется путем нажатия кнопки “Stop” на пользовательской панели. В программе использовать только одну исходную точку в таблице сохраненных точек. Остальные положения определять смещениями.

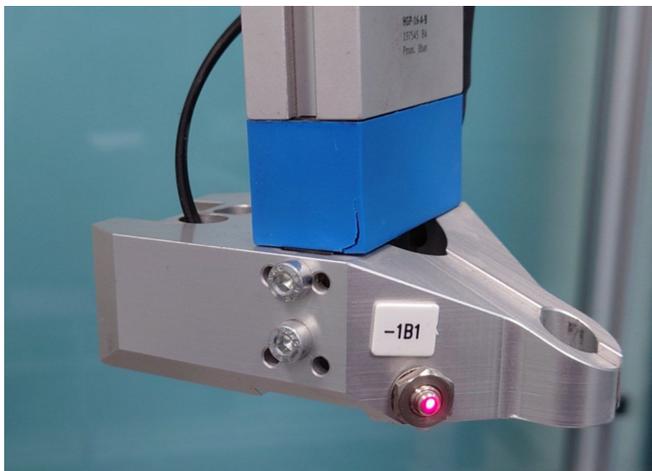


Рис. 4.3. Захватное устройство с датчиком наличия деталей

## 4.2.2 Основные команды программирования

При выполнении программы использовать следующие команды программирования языка MELFA BASIC:

`WAIT M_IN(2)=1` – приостановка выполнения программы и ожидание сигнала «1» на дискретном порте «2».

`DEF INTE M1` – создание вспомогательной переменной M1.

`M1=1` – присваивание значения переменной M1.

`WHILE M1>0` – начало цикла с условием.

`WHILE 1` – начало бесконечного цикла.

`WEND` – завершение цикла с условием.

`IF M_IN(2)=0 THEN` – начало условия при наличии «0» на дискретном порте «2».

`ENDIF` – завершение условия.

`DEF ACT 1, M_IN(3)=0 GOSUB *SUBSTOP` – объявление прерывания №1 при условии наличия «0» на дискретном порте «3» с вызовом процедуры \*SUBSTOP. Объявляется в начале программы.

`ACT 1 = 1` – активация прерывания №1.

\*SUBSTOP – объявление процедуры \*SUBSTOP. Пишется после основного кода программы.

RETURN — завершение процедуры.

### 4.2.3 Порядок выполнения

1. Зайти в студии разработки в раздел «RV-3SDB Monitors I/O Monitor». В окне монитора определить номера дискретных портов ввода для кнопок «Reset», «Stop» пользовательской панели, а также датчика на захватном устройстве. Номера и начальные значения кнопок определяются путем нажатия, номер датчика и дальность действия определяются путем поднесения предмета к датчику.
2. Написать программу, которая включает двигатель, ждёт нажатия кнопки «Reset» на пользовательской панели, по нажатию кнопки отключает двигатели и завершает программу.
3. Дополнить программу, чтобы робот без бесконечного цикла мог перемещаться между тремя деталями, определял их наличие, в случае наличия детали — в данном положении поднять захватное устройство на 100 мм и опустить в исходное положение.
4. Дополнить программу бесконечным циклом, чтобы робот постоянно перемещался между деталями и проверял их наличие. Написать обработчик прерывания по нажатию «Stop» на пользовательской панели, который отключает двигатели и корректно завершает выполнение программы.

### 4.2.4 Содержание отчета

1. Титульный лист
2. Цели выполнения работы
3. Код конечной программы
4. Описание команд
5. Таблица сохраненных точек

6. Фотографии или рисунки расположения необходимых элементов робототехнической ячейки во время работы
7. Четыре фотографии, иллюстрирующие этапы выполнения программы
8. Выводы по работе

### **4.3 Лабораторная работа 3 «Формирование комплексных траекторий»**

#### **4.3.1 Задание на лабораторную работу**

Захватить инструмент захватным устройством. Касаясь щупом инструмента, обойти весь контур изогнутой листовой детали (рисунок 4.4). При касании щупа детали загорятся светодиоды на инструменте, имитируя процесс сварки. По окончании обхода контура детали вернуть инструмент в исходное положение. При составлении программы использовать команды `MOV`, `MVS`, `MVR`, смещения по оси *Z* инструмента.



Рис. 4.4. Листовая деталь для лабораторной работы

### 4.3.2 Основные команды программирования

При выполнении программы использовать следующие команды программирования языка MELFA BASIC:

END — завершение программы, обязательно размещается в конце файла.

SERVO ON — включение двигателей, размещается в начале программы.

JOVRD 100 — скорость движения в процентах от максимальной, обычно размещается в начале программы.

SERVO OFF — выключение двигателей, размещается в конце программы.

DLY 0.1 — пауза выполнения программы в секундах.

HOPEN 1 — открытие захватного устройства.

HCLOSE 1 — закрытие захватного устройства.

MOV P1 — движение в точку P1 из таблицы сохраненных точек.

MOV P1, -50 — движение в точку P1 со смещением 50 мм вверх по оси Z.

MVS P1 — точное движение в точку P1 по прямой линии.

MVR P1, P2, P3 — движение по дуге через точки P1, P2, P3.

### 4.3.3 Порядок выполнения

1. Составить программу, в которой робот будет зависать на 250 мм выше инструмента, опускаться к инструменту, захватывать инструмент, поднимать на 250 мм, ждать 1 секунду, возвращать инструмент на место, снова зависать на 250 мм выше инструмента.
2. Дополнить программу движением подхода к первой точке касания, комплексной траекторией движения по детали, состоящей из дуг и прямых линий, движением отхода от последней точки траектории.

### 4.3.4 Содержание отчета

1. Титульный лист
2. Цели выполнения работы

3. Код конечной программы
4. Описание команд
5. Таблица сохраненных точек
6. Четыре фотографии, иллюстрирующие этапы выполнения программы
7. Выводы по работе

## 4.4 Лабораторная работа 4 «Сетевые сообщения»

### 4.4.1 Задание на лабораторную работу

Написать программу, которая будет считывать из TCP/IP порта 10003 номера деталей по осям X и Y (от 0 до 2 по каждой из осей), затем переносить указанную деталь с первого стола с деталями на второй пустой стол. На втором столе детали следует выкладывать последовательно линиями. Указанные операции должны производиться 9 раз (по общему числу деталей).

### 4.4.2 Основные команды программирования

При выполнении программы использовать следующие команды программирования языка MELFA BASIC:

DEF INTE IX, IY — создание вспомогательных переменных IX, IY.

FOR I1= 0 TO 2 — начало выполнения цикла, I1 — переменная итерации цикла.

NEXT I1 — окончание цикла.

WHILE M1>0 — начало цикла с условием.

WEND — завершение цикла с условием.

IF M\_IN(2)=0 THEN — начало условия при наличии «0» на дискретном порте «2».

ENDIF — завершение условия.

OPEN "COM3:" AS #1 — открытия TCP/IP порта 10003 для подключения интерфейса #1.

INPUT #1, IX, IY — считывание содержимого буфера интерфейса #1 в переменные IX и IY.

PRINT #1, IX, IY — вывод содержимого переменных IX и IY в интерфейс #1.

Для создания настраиваемого смещения координат можно в коде программы создавать вспомогательные переменные координат:

```
RHELPH=(75,0,0,0,0,0)
```

где в скобках записываются декартовы координаты смещения X, Y, Z, A, B, C соответственно.

В команде MOV кроме явного указания точек (например, P1) можно проводить математические операции над точками, например сумма переменных для смещения:

```
MOV P1+RHELPH, -50
```

или умножение на переменные для выполнения операций в цикле:

```
MOV P1+RHELPH*I1, -50
```

### 4.4.3 Необходимые команды программирования на Python

Отправка пакета и чтение ответного сообщения:

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("172.16.0.4", 10003))
print("Введите положение детали (X, Y)")
p_out = input()
s.send(p_out.encode())
p_in = s.recv(1024)
print("Ответ:")
print(p_in.decode())
s.close()
```

Цикл:

```
for i in range(4):  
    ...
```

Цикл с предусловием:

```
while i <= 10:  
    ...
```

#### 4.4.4 Порядок выполнения

1. В ручном режиме управления робот определить две точки, сохранить в таблицу. Написать программу, которая открывает захват, зависает над первой точкой, опускается к первой точке, закрывает захват, поднимает деталь, зависает над второй точкой, опускает деталь, открывает захват, поднимает робота над деталью.
2. Дополнить программу, чтобы робот в цикле мог переносить ряд деталей со смещением по одной координате.
3. Дополнить программу, чтобы робот в цикле мог переносить кластер деталей со смещением по двум координатам.
4. Дополнить программу, чтобы робот в цикле считывал координаты новой детали из TCP/IP пакета и переносил её на второй стол.
5. Составить программу для компьютера на Python, которая будет из консоли 9 раз считывать необходимые координаты новой детали и отправлять их на контроллер робота.

#### 4.4.5 Содержание отчета

1. Титульный лист
2. Цели выполнения работы
3. Код конечной программы
4. Описание команд
5. Таблица сохраненных точек

6. Четыре фотографии, иллюстрирующие этапы выполнения программы.
7. Выводы по работе

# Заключение

В учебном пособии рассматриваются основные органы управления, алгоритмические конструкции, команды управления движением, принципы работы с внешними устройствами промышленных роботов.

Первая глава посвящена общему пониманию промышленных роботов. Описываются типовые составные части промышленных роботов, критичные характеристики, отличия от непромышленной робототехники, требования к подготовке помещений. Раскрывается классификация промышленных роботов на основе базовых характеристик и требований, которые к ним выдвигаются. Представлены основные типы промышленных роботов с описанием конструкций, преимуществ и недостатков, основных областей применения. Внимание уделено также описанию захватных устройств, применимых для промышленной робототехники. Приведены классические и новые области применения промышленных роботов; описаны процессы, в которых промышленные роботы показывают свою эффективность.

Во второй главе описаны основы взаимодействия с промышленным роботом серии Mitsubishi Melfa. Описаны основные органы управления, необходимые для управления в ручном дистанционном и автоматическом режимах работы. Представлено описание интерфейса программирования на компьютере, последовательности действий, необходимых для подготовки и тестирования алгоритмов.

Третья глава содержит синтаксис языка программирования промышленных роботов серии Mitsubishi Melfa. Представлены базовые типы переменных и алгоритмические конструкции для построения основных программ. Приведены различные команды движения промышленного робота с учетом его конструкции, представлены примеры их использования. Описаны принципы работы с внешними устройствами

различных типов.

В четвертой главе приведены описания лабораторных работ по курсу «Программирование промышленных роботов» по направлению подготовки «Робототехника» Университета ИТМО. Содержание лабораторных работ основано на материалах данного учебного пособия. В главе представлены задания на лабораторные работы, основные необходимые команды программирования промышленного робота, порядок выполнения, содержание отчета.

# Литература

- [1] *Spong M. W., Hutchinson S., Vidyasagar M.*, Robot modeling and control. John Wiley & Sons, 2020.
- [2] *Борисов О. И., Громов В. С., Пыркин А. А.*, Методы управления робототехническими приложениями, СПб.: Университет ИТМО, 2016.
- [3] *Siciliano B., Khatib O.*, Springer handbook of robotics. Springer Science & Business Media, 2008.
- [4] *Фу К., Гонсалес Р., Лу К.*, Робототехника: Пер. с англ. М.: Мир, 1989.
- [5] *Артоболевский И. И.*, Теория машин и механизмов. М. Наука 1988.
- [6] *Bill M., Muller C., Kraus W.*, Executive Summary World Robotics 2019 Industrial Robots, International Federation of Robotics, 2019.

Громов Владислав Сергеевич  
Борисов Олег Игоревич  
Синетова Мадина Мансуровна

**Программирование промышленных роботов серии  
Mitsubishi Melfa**

**Учебное пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49