

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

Курсовой проект
по дисциплине «Объектно-ориентированное программирование»
«Программирование промышленного манипулятора KUKA»

Пояснительная записка

Выполнил
студент гр.
3331506/10401

(подпись)

Андреев А. И.
Гатауллина Н. Р.
Рыбьяков Д. А.

Работу принял

(подпись)

Ананьевский М.С.

Санкт-Петербург
2024 г.

Оглавление

Оглавление	2
Техническое задание	3
1. Введение	4
2. Ход работы	5
2.1. Знакомство с роботом	5
2.1.1. Общая информация	5
2.1.2. Приведение робота в работоспособное состояние	6
2.2. Управление роботом в ручном режиме	7
2.3 Подключение к роботу	8
2.4. Программирование робота	9
2.4.1. Структура	9
2.4.2. Переменные и типы данных	9
2.4.3. Условные операторы	10
2.4.4. Циклы	10
3. Итоговая реализация	11
Заключение	14
Список литературы	15
Приложение	16

Техническое задание

Изучить теоретические сведения о роботе KUKA, протестировать базовые настройки, произвести юстировку, разработать и написать программу для робота KUKA KR 6 R900 sixx.

1. Введение

Компания KUKA является одним из ведущих производителей промышленных роботов. У них нет жесткого деления на категории, роботы из разных классов могут одинаково хорошо использоваться в решении различных задачи. Они придерживаются следующей философии: робот должен подбираться под задачу, а не наоборот. Их промышленные роботы обычно делятся на 7 категорий по диапазонам рабочих нагрузок:

- Малые роботы (7–14 кг) (компактные и очень быстрые роботы, которые предназначены для работы с миниатюрными деталями в наибольших производственных ячейках 600х600 мм)
- Низкий класс грузоподъемности (5–16 кг) (они оптимизированы для применения в условиях непрерывного производства, подходят для сварки, нанесения клеев и герметиков, имеют даже на большой скорости работы высокую точность позиционирования)
- Средний класс грузоподъемности (30–60 кг) (предназначены для работы роботов в экстремальных условиях литейного производства, также подходят для молярных, сварочных и погрузочно-разгрузочных работ)
- Высокий класс грузоподъемности (80–300 кг) (имеют модульную систему и большое разнообразие модулей, благодаря чему подходят для любой производственной среды)
- Сверхмощные роботы (до 1300 кг) (могут точно и быстро работать с грузами достигающими 1300 кг, могут обрабатывать тяжелые грузы на расстоянии до 6 метров)
- Паллетайзеры (роботы предназначены для укладки грузов на поддоны в точном порядке и с высокой скоростью)
- Коллаборативные роботы (7–14 кг) (эти роботы способны мгновенно реагировать на препятствия и останавливаться, могут безопасно взаимодействовать с человеком, а также имеют функцию обучения с помощью обратной связи)

В рамках выполнения курсового проекта используется робот KUKA KR 6 R900 sixx с системой управления KRC4.

2. Ход работы

2.1. Знакомство с роботом

2.1.1. Общая информация

В комплектацию робота входит:

1. Робот
2. Система управления роботом
3. Переносное программирующее устройство
4. Соединительные кабели

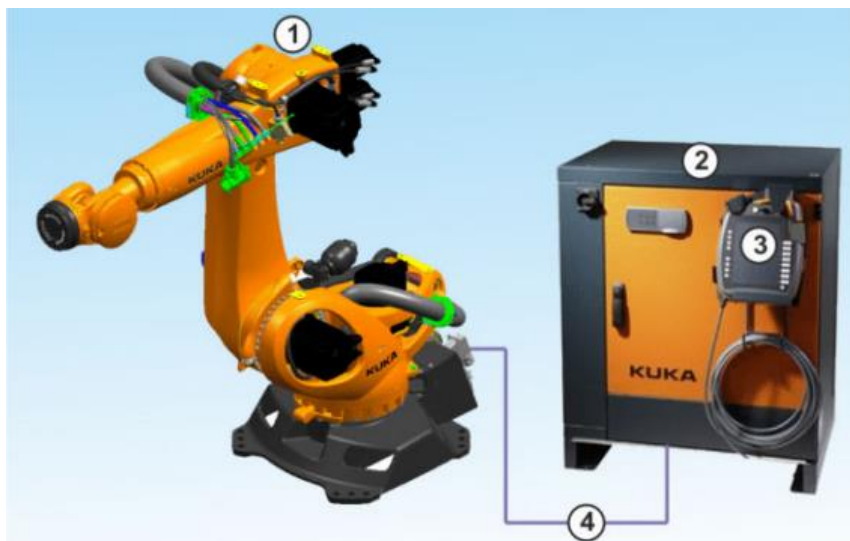


Рисунок 1. Составляющие части робота.

На выходном фланце робота в качестве захвата используется захват SCHUNK MEG 40 EC способный брать небольшие объекты простой формы. Он представлен на рисунке 2.



Рисунок 2. Захват SCHUNK MEG 40 EC.



Рисунок 3. Промышленный манипулятор KUKA.

2.1.2. Приведение робота в работоспособное состояние

Для включения системы управления робота необходимо переключить тумблер, находящийся на задней стороне короба системы управления показанный на рисунке 4.



Рисунок 4. Задняя сторона системы управления

После этого необходимо на пульте управления KUKA smartPAD выбрать один из трех режимов работы робота.

Режим T1 – ручной снижение скорости. Используется для тестовой эксплуатации, программирования и обучения. Имеет пониженные скорости.

Режим T2 – ручной высокая скорость. Используется для тестовой эксплуатации. Скорость соответствует запрограммированной.

Режим AUT – автоматический режим. Используется для нормальной эксплуатации.

2.2. Управление роботом в ручном режиме

Робот имеет возможность управляться в режиме реального времени с помощью KUKA smartPAD представленного на рисунках 5 и 6.

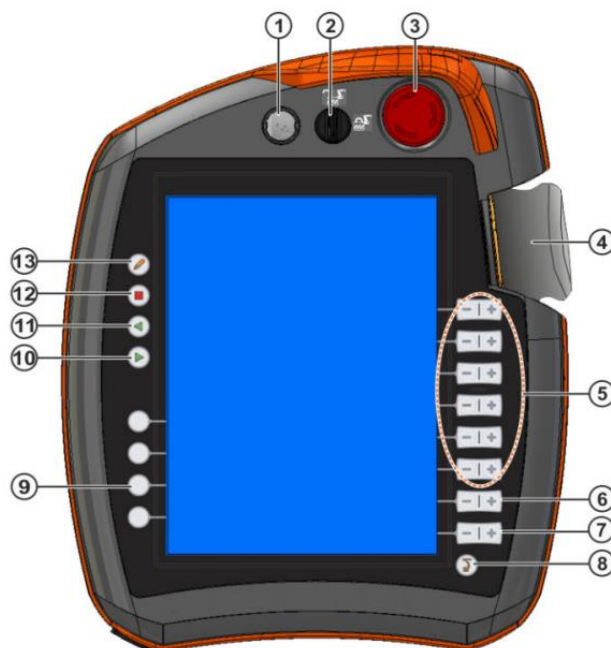


Рисунок 5. Передняя сторона пульта управления KUKA smartPAD.



Рисунок 6. Задняя сторона пульта управления KUKA smartPAD.

После включения системы управления необходимо отжать кнопку 3 с передней панели пульта и выбрать режим T1. Для управления роботом необходимо удерживать в среднем положении кнопки 1, 3 или 4 с задней панели пульта и с помощью блока кнопок 5 с передней панели пульта регулировать положение робота. Также робот может управляться с помощью пространственной 6D-мыши под номером 4 на передней панели пульта.

В ручном режиме легче всего управлять каждой из его степеней свободы. Всего степеней подвижности 6 (без учета степеней подвижности устанавливаемого захвата), все кинематические пары в манипуляторе являются вращательными. Также роботом можно управлять в различных системах координат, описываемых положением относительно осей X, Y, Z и углом поворота относительно этих осей A, B, C. Системы координат могут быть связаны с базой робота, с инструментом робота, а также они могут быть настроены так как удобно пользователю.

Также в ручном режиме, как и в программном можно менять системы координат, в которых производится управление. Основных системы координат 3: WORLD, BASE, TOOL. BASE это система координат связанная с местом крепления робота. WORLD переназначаемая которую можно поставить в любое место для упрощения написания программы. TOOL связана с инструментом. Каждая из этих систем координат может быть изменена в программе и в дальнейшем использована.

2.3 Подключение к роботу

В первую очередь необходимо подключить компьютер к KLI порту контроллера KUKA с помощью кабеля Ethernet. Далее необходимо установить IP-адрес компьютера в регион IP-адреса робота. Чтобы узнать IP-адрес робота необходимо нажать на значок робота на SmartPad, перейти в раздел Start-up, затем в Network Configuration. IP - адрес робота – 172.31.1.147, поэтому мы установили адрес 172.31.1.146, чтобы он находился в том же регионе, но не совпадал с адресом робота.

Помимо этого, для корректной работы с контроллером KUKA, необходимо установить язык системы, такой же как язык системы SmartPad, в нашем случае английский. WorkVisual позволяет скачивать существующие в работе проекты редактировать, удалять и создавать новые и затем загружать их обратно в память робота. Для того, чтобы можно было принять проекты выгружаемые из WorkVisual необходимо установить на SmartPad уровень доступа Expert. Переходим в раздел Users, устанавливаем режим Expert, используя стандартный пароль KUKA.

2.4. Программирование робота

2.4.1. Структура

Программирование всех роботов KUKA осуществляется с помощью языка программирования высокого уровня KRL (KUKA Robot Language) разработанным компанией KUKA.

Любая программа на языке KRL состоит из двух текстовых файлов с расширениями *.src и *.dat, где * - одно и то же имя. В файле *.src (файл кода) находится код программы, а в файле *.dat (файл данных) расположены описания переменных, точек, массивов и т. д. для сокращения размера файла *.src.

Файл *.src состоит из «главной» и «дополнительных» функций. «Главная» функция должна называться так же, как и сам (*.src) файл, в котором она находится.

Функции в KRL состоят из трех основных разделов которые строго должны идти друг за другом.

Раздел объявления служит для объявления переменных, массивов, точек, структур и т. д. Раздел инициализации нужен для инициализации переменных, массивов и т. д., объявленных в разделе объявления. Раздел инструкций является основным разделом, в котором находится основной код программы. Объявление функции начинается с ключевого слова «DEF» и заканчивается словом «END».

2.4.2. Переменные и типы данных

Для объявления переменной необходимо сначала прописать DECL (выделяет место в памяти под переменную), затем тип данных, а потом имя переменной.

```
DECL INT VAR_1
```

На рисунке 7 представлены простые типы данных языка KRL.

Типы данных	Целое	С точкой	Логический	Символьный
Ключевое слово	INT	REAL	BOOL	CHAR
Диапазон значений	$-2^{31} \dots 2^{31}-1$	$\pm 1.1\text{E}-38 \dots \pm 3.4\text{E}+38$	TRUE, FALSE	ASCII character

Рисунок 7 – простые типы данных языка KRL

Также в KRL существуют структурные типы данных, в которых очень удобно обозначать позицию робота. На рисунке представлены основные структурные типы заранее определенные в KRL.

STRUC AXIS	REAL	A1,A2,A3,A4,A5,A6
STRUC E6AXIS	REAL	A1,A2,A3,A4,A5,A6,E1,E2,E3,E4,E5,E6
STRUC FRAME	REAL	X,Y,Z,A,B,C
STRUC POS	REAL	X,Y,Z,A,B,C, INT S,T
STRUC E6POS	REAL	X,Y,Z,A,B,C,E1,E2,E3,E4,E5,E6, INT S,T

Рисунок 8 – основные структурные типы

2.4.3. Условные операторы

Условный оператор IF

```
IF условие THEN
    Инструкции, если условие равно TRUE
ELSE
    Инструкции, если условие равно FALSE
ENDIF
```

Условный оператор SWITCH

```
SWITCH имя переменной
    CASE значение переменной
        Инструкции, если переменная приняла значение выше
    CASE
        Инструкции, если переменная приняла значение выше
    DEFAULT
        Инструкции в ином случае
ENDSWITCH
```

2.4.4. Циклы

Цикл WHILE

```
WHILE условие
    инструкции
ENDWHILE
```

Цикл FOR

```
FOR счетчик = начало TO конец STEP шаг
    инструкции
ENDFOR
```

3. Итоговая реализация

В ходе изучения программирования роботов KUKA была реализована программа на языке KRL для робота KUKA KR 6 R900 sixx.

В первой использовалась среда программирования WorkVisual разработанная специально для работы с роботами KUKA. В ходе выполнения программы робот построит и разберет трехэтажную плоскую пирамиду. Полный код программы представлен в приложении, а здесь основные моменты.

Сначала идет раздел объявления, в котором были объявлены все переменные, которые используются далее.

```
DECL INT COUNTER
DECL INT CUBE
DECL INT CUBE_DISTANCE
DECL FRAME CUBE_POS
DECL FRAME VERT_MOVE
DECL FRAME START_PYRAMID
DECL FRAME NEW_HOME
```

Далее идет раздел инициализации, в котором были присвоены некоторые из переменных. Мы привели робота в рабочее положение (NEW_HOME) из которого будем начинать и установили эту точку как систему координат BASE. Также задали точки, из которых будем начинать строить пирамиду и брать кубики.

```
CUBE = 30
CUBE_DISTANCE = 10
CUBE_POS = {X 0, Y 0, Z 0, A 0, B 0, C 0}
START_PYRAMID = {X 0, Y 300, Z 0, A 0, B 0, C 0}
NEW_HOME = {X 94.9, Y 862.6, Z 43.9, A 146, B 0, C 0}
```

Потом проводятся некоторые настройки робота чтобы дальнейшая программа выполнялась корректно.

```
GRAB_OPEN()
$BASE = $WORLD
$BASE = NEW_HOME
```

После этого идут два цикла которые собирают и разбирают пирамиду, после них основная функция заканчивается.

```
FOR COUNTER = 1 TO 6 STEP 1

    CUBE_POS.X = CUBE_POS.X - CUBE - CUBE_DISTANCE

    VERT_MOVE = CUBE_POS
    VERT_MOVE.Z = VERT_MOVE.Z + 60
    PTP VERT_MOVE
    PTP CUBE_POS
    GRAB_CLOSE()
```

```

PTP VERT_MOVE
VERT_MOVE = START_PYRAMID
VERT_MOVE.Z = VERT_MOVE.Z + 60

PTP VERT_MOVE
PTP START_PYRAMID
GRAB_OPEN()
PTP VERT_MOVE

SWITCH COUNTER
CASE 1,2
    START_PYRAMID.X = START_PYRAMID.X + 35
CASE 3
    START_PYRAMID.Z = START_PYRAMID.Z + 30
    START_PYRAMID.X = START_PYRAMID.X - 17.5
CASE 4
    START_PYRAMID.X = START_PYRAMID.X - 35
CASE 5
    START_PYRAMID.Z = START_PYRAMID.Z + 30
    START_PYRAMID.X = START_PYRAMID.X + 17.5
DEFAULT
    GRAB_OPEN()
ENDSWITCH
ENDFOR

FOR COUNTER = 1 TO 6 STEP 1

    VERT_MOVE = START_PYRAMID
    VERT_MOVE.Z = VERT_MOVE.Z + 60
    PTP VERT_MOVE
    PTP START_PYRAMID
    GRAB_CLOSE()
    PTP VERT_MOVE
    VERT_MOVE = CUBE_POS
    VERT_MOVE.Z = VERT_MOVE.Z + 60
    PTP VERT_MOVE
    PTP CUBE_POS
    GRAB_OPEN()
    PTP VERT_MOVE

    CUBE_POS.X = CUBE_POS.X + CUBE + CUBE_DISTANCE

    SWITCH COUNTER
    CASE 1
        START_PYRAMID.Z = START_PYRAMID.Z - 30
        START_PYRAMID.X = START_PYRAMID.X - 17.5
    CASE 2
        START_PYRAMID.X = START_PYRAMID.X + 35
    CASE 3
        START_PYRAMID.Z = START_PYRAMID.Z - 30
        START_PYRAMID.X = START_PYRAMID.X + 17.5
    CASE 4,5
        START_PYRAMID.X = START_PYRAMID.X - 35
    DEFAULT
        GRAB_OPEN()
        PTP NEW_HOME
    ENDSWITCH
ENDFOR

```

Далее две функции которые отвечаю за открывание и закрывание захвата.

```
DEF GRAB_CLOSE ()  
    $OUT[3]=TRUE  
    WAIT SEC 1  
    $OUT[3]=FALSE  
END
```

```
DEF GRAB_OPEN ()  
    $OUT[4]=TRUE  
    WAIT SEC 1  
    $OUT[4]=FALSE  
END
```

Заключение

В ходе выполнения курсовой работы была изучена среда программирования промышленных манипуляторов KUKA WorkVisual, а также язык программирования KRL. Была написана программа сбора и разбора плоской трехэтажной пирамиды.

Список литературы

1. <https://wikis.utexas.edu/display/SOAdigitech/KUKA+Programming+KRL+Examples>
2. <https://drstienecker.com/tech-332/11-the-kuka-robot-programminglanguage/>
3. https://swsu.ru/sveden/files/PROGRAMMIROVANIE_PROMYSHLENNOGO_ROBOTA_KUKA_LAB.pdf
4. https://www.youtube.com/watch?v=GtxShP_Wtec&t=171s&ab_channel=FutureRobotics

Приложение

```
DEF Andreev ( )

;FOLD DECL
DECL INT COUNTER
DECL INT CUBE
DECL INT CUBE_DISTANCE

DECL FRAME CUBE_POS
DECL FRAME VERT_MOVE

DECL FRAME START_PYRAMID

DECL FRAME NEW_HOME
;ENDFOLD

;FOLD INI
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
;FOLD USER INI
CUBE = 30
CUBE_DISTANCE = 10

CUBE_POS = {X 0, Y 0, Z 0, A 0, B 0, C 0}

START_PYRAMID = {X 0, Y 300, Z 0, A 0, B 0, C 0}

NEW_HOME = {X 94.9, Y 862.6, Z 43.9, A 146, B 0, C 0}
;ENDFOLD (USER INI)
;ENDFOLD (INI)

;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS,%CMOVE,%VPTP,%P
1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
PDAT_ACT=PDEFAULT
FDAT_ACT=FHOME
BAS (#PTP_PARAMS,100 )
$H_POS=XHOME
PTP XHOME
;ENDFOLD

;FOLD MAIN

GRAB_OPEN()
$BASE = $WORLD
$BASE = NEW_HOME

FOR COUNTER = 1 TO 6 STEP 1

    CUBE_POS.X = CUBE_POS.X - CUBE - CUBE_DISTANCE

    VERT_MOVE = CUBE_POS
    VERT_MOVE.Z = VERT_MOVE.Z + 60
```



```

PTP VERT_MOVE
PTP CUBE_POS
GRAB_CLOSE()
PTP VERT_MOVE

VERT_MOVE = START_PYRAMID
VERT_MOVE.Z = VERT_MOVE.Z + 60

PTP VERT_MOVE
PTP START_PYRAMID
GRAB_OPEN()
PTP VERT_MOVE

SWITCH COUNTER
CASE 1,2
    START_PYRAMID.X = START_PYRAMID.X + 35
CASE 3
    START_PYRAMID.Z = START_PYRAMID.Z + 30
    START_PYRAMID.X = START_PYRAMID.X - 17.5
CASE 4
    START_PYRAMID.X = START_PYRAMID.X - 35
CASE 5
    START_PYRAMID.Z = START_PYRAMID.Z + 30
    START_PYRAMID.X = START_PYRAMID.X + 17.5
DEFAULT
    GRAB_OPEN()
ENDSWITCH
ENDFOR

FOR COUNTER = 1 TO 6 STEP 1

    VERT_MOVE = START_PYRAMID
    VERT_MOVE.Z = VERT_MOVE.Z + 60

    PTP VERT_MOVE
    PTP START_PYRAMID
    GRAB_CLOSE()
    PTP VERT_MOVE

    VERT_MOVE = CUBE_POS
    VERT_MOVE.Z = VERT_MOVE.Z + 60

    PTP VERT_MOVE
    PTP CUBE_POS
    GRAB_OPEN()
    PTP VERT_MOVE

    CUBE_POS.X = CUBE_POS.X + CUBE + CUBE_DISTANCE

    SWITCH COUNTER
    CASE 1
        START_PYRAMID.Z = START_PYRAMID.Z - 30
        START_PYRAMID.X = START_PYRAMID.X - 17.5
    CASE 2
        START_PYRAMID.X = START_PYRAMID.X + 35
    CASE 3
        START_PYRAMID.Z = START_PYRAMID.Z - 30
        START_PYRAMID.X = START_PYRAMID.X + 17.5

```

```

        CASE 4,5
            START_PYRAMID.X = START_PYRAMID.X - 35
        DEFAULT
            GRAB_OPEN()
            PTP NEW_HOME
    ENDSWITCH
ENDFOR

;ENDFOLD

;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS,%CMOVE,%VPTP,%P
1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
PDAT_ACT=PDEFAULT
FDAT_ACT=FHOME
BAS (#PTP_PARAMS,100 )
$H_POS=XHOME
PTP XHOME
;ENDFOLD

END

DEF GRAB_CLOSE ()
    $OUT[3]=TRUE
    WAIT SEC 1
    $OUT[3]=FALSE
END

DEF GRAB_OPEN ()
    $OUT[4]=TRUE
    WAIT SEC 1
    $OUT[4]=FALSE
END

```