|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ СПЕЦИАЛЬНОЕ МАШИНОСТРОЕНИЕ

КАФЕДРА СМ-7 РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ И МЕХАТРОНИКА

**Домашняя работа №2**

по курсу

**“Управление роботами”**

Студент К.С. Балкунов

Группа СМ7-74Б

*2024 г.*

***Задача***

Создать и отладить программы управления роботом манипуляторомна языке *Python* в среде *CoppeliaSim.*

Описание работы:

* Изучить базовые функции и элементы моделирования и управления роботом манипуляторомс оснасткой в среде *CoppeliaSim* в соответствии с методическими указаниями.
* Изучить возможности программирования робота на языке *Python*.
* Организовать управление роботом манипулятором для выполнения задачи рисования окружности по трем заданным точкам.

Условия задания:

**Входные данные:** Пользователь задает три точки – координаты x, y, z относительно базовой системы координат

**Реализовать:**

* Проверку принадлежности точек окружности рабочей области манипулятора;
* Определение последовательности положений манипулятора для рисования окружности.

1. **Настройка CoppeliaSim.**

Открываем файлы программы CoppeliaSim с манипуляционным роботом *CRP-RA-27-80,* файл *“scene”:*

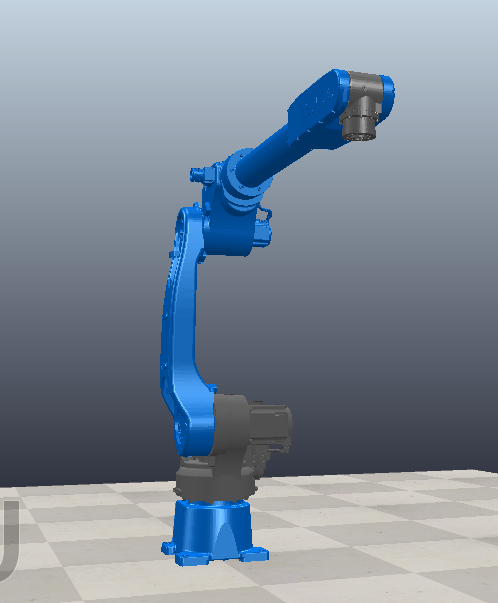
**

Рисунок 1. Модель манипулятора в CoppeliaSim

1. **Организация движения робота**

В программе **run.py** пользователь вводит три точки в трехмерном пространстве.

После этого проводиться проверка на принадлежность точек окружности рабочему пространству манипулятора. Модуль радиус-вектор любой точки окружности должен быть больше наименьшего радиуса достижимости и меньше наибольшего радиуса достижимости манипулятора. Проверка проводиться в функции **is\_point\_reachable().** Если условия выполняются, то функция возвращает **True**, иначе – программа останавливается.

Метод класса Manipulator **solve\_ik()** решает обратную задачу по положению. Внутри метода используется минимизация Бройдена-Флетчера-Гольдфарба-Шанно из пакета scipy.

После выполнения минимизации ошибки получаем следующие значения обобщенных координат, к которым осуществляем движение.

***Вывод***

Результатом лабораторной работы является:

* Создание и отладка программы управления манипуляционным роботомна языке P*ython* в среде *CoppeliaSlim.*
* Базовая проверка принадлежности точек окружности рабочей области манипулятора;
* Реализация рисования окружности по трем точкам.

***Приложение***

Репозиторий проекта можно получить по ссылке: [coppelia-robot-ctrl/hm\_2 at main · bobrotopor/coppelia-robot-ctrl (github.com)](https://github.com/bobrotopor/coppelia-robot-ctrl/tree/main/hm_2)