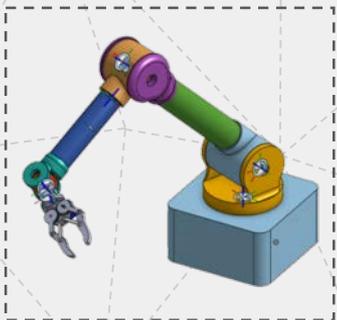


РАЗРАБОТКА И СОЗДАНИЕ ГЕНЕРАТОРА КОМАНД ПОЗИЦИОНИРОВАНИЯ РОБОТА МАНИПУЛЯТОРА

Шидловская Д.В., Довгулевич Д.А.



ЦЕЛЬ РАБОТЫ

Целью работы является разработка программного комплекса на базе мета -операционной системы ROS, позволяющая генерировать программы управления, состоящие из команд точного позиционирования, для робота-манипулятора.

ЗАДАЧИ

1. Анализ программного обеспечения для разработки модели робота манипулятора, его тестирования, запуска на реальном оборудовании;
2. Разработка 3D модели робота манипулятора и его URDF представление;
3. Разработка программно - вычислительного комплекса для управления роботом манипулятором;
4. Разработка серверной части для автоматического запуска генерации траектории и ее выполнения роботом манипулятором;
5. Описание и разработка постпроцессора для взаимодействия с различными видами роботов - манипуляторов;
6. Разработка клиентской части, для управления движениями робота-манипулятора.
7. Разработка единого файла развертывания и сборки проекта.

АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Onshape



docker

Movelt

Разработка 3D модели робота -манипулятора



Создание цифрового робота - манипулятора начинается с разработки 3D модели робота.

- База - основание робота, относительно которого будут располагаться остальные сочленения робота;

- Звенья робота, которые в совокупности образуют гибкую цепь и позволяют дотянуться в любую точку, в пределах досягаемости;

- Конечный эффектор робота - представляет собой инструмент манипуляции, с помощью которого можно выполнять определенные действия, к примеру, перенос предметов из одного места в другое.

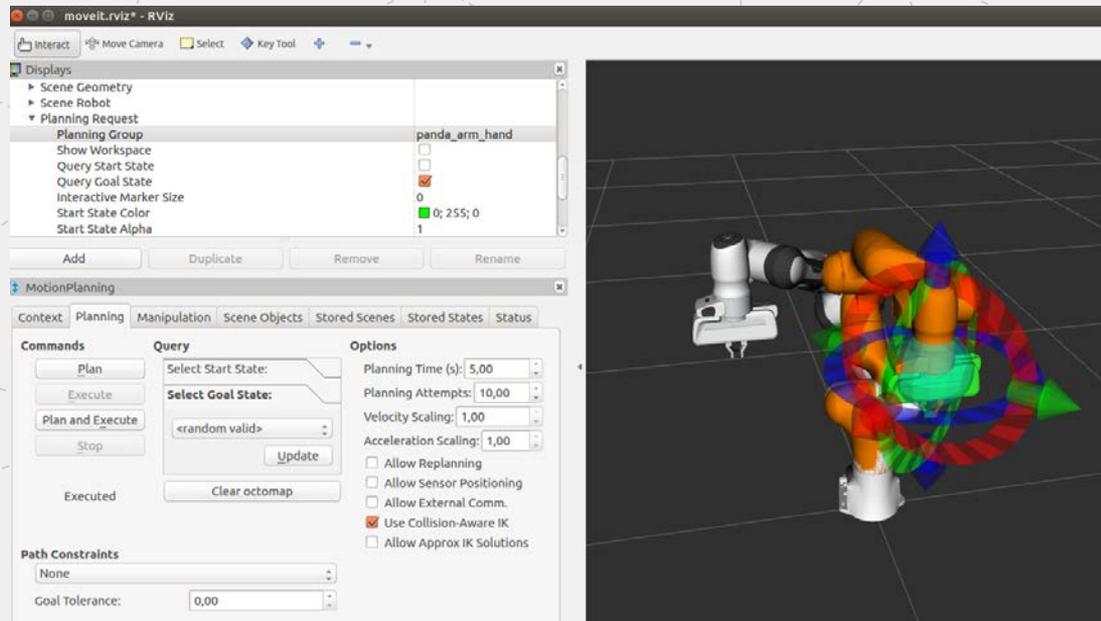
3D МОДЕЛЬ РОБОТА-МАНИПУЛЯТОРА

Onshape

URDF ПРЕДСТАВЛЕНИЕ И ВИЗУАЛИЗАЦИЯ RVIZ

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="./visual/base_link.stl"/>
    </geometry>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="./collision/base_link.stl"/>
    </geometry>
  </collision>
</link>
```

```
<joint name="joint_1" type="revolute">
  <origin xyz="0 0 0.450" rpy="0 0 0"/>
  <parent link="base_link"/>
  <child link="link_1"/>
  <axis xyz="0 0 1"/>
  <limit lower="-3.1415" upper="3.1415"/>
</joint>
```



URDF МОДЕЛЬ РОБОТА-МАНИПУЛЯТОРА



УДАЛЕННЫЙ ЗАПУСК ГЕНЕРАЦИИ ПРОГРАММ

```
syntax = "proto3";
import "google/protobuf/any.proto";
package robot6r_message;

// primitive for coords save
message Point {
    double x = 1; ///< x coord
    double y = 2; ///< y coord
    double z = 3; ///< z coord

    double w = 4; ///< x coord
    double p = 5; ///< y coord
    double r = 6; ///< z coord

    enum Grip {
        GRIP = 0;
        UNGRIP = 1;
    }
    optional Grip grip = 7 [default = UNGRIP];
}

message Trajectory {
    repeated Point point = 1; ///< point array
}

message Operation {
    repeated Trajectory polygon = 1; ///< operation trajectory
    string operation_type = 4; ///< operation type
    string meta_data = 5; ///< meta info
}
```

Используя фреймворк для удаленного вызова процедур **gRPC** возможно однозначно определить обмен сообщениями между клиентом и сервисом посредством использования языка определения интерфейса.

Для gRPC в основном языке определения интерфейса является буфер протокола **Protobuf**.

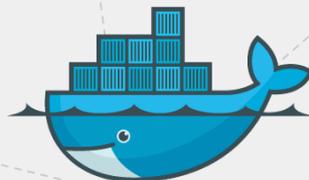
Он позволяет хранить свои данные и функциональные контракты в виде прото-файла. Поскольку это форма контракта, и клиент, и сервер должны иметь один и тот же прото-файл. Файл proto действует как посреднический контракт для клиента для вызова любых доступных функций с сервера.

PROTOBUF ФАЙЛ ОПИСАНИЕ ТРАЕКТОРИИ



СИСТЕМА ГЕНЕРАЦИИ ПРОГРАММ И УПРАВЛЕНИЯ

Структура проекта на базе мета - операционной системы ROS является многокомпонентной, имеющая различные встроенные и внешние зависимости.



docker

СБОРКА И РАЗВЕРТЫВАНИЕ ПРОЕКТА

Перенос такого проекта на другую машину затруднителен, так как придется устанавливать разного рода библиотеки, которую в свою очередь могут конфликтовать с уже установленными приложениями.

Поэтому для установки проекта на другую машину лучше воспользоваться технологией контейнеризации.

РЕАЛЬНЫЕ РОБОТЫ

Различные виды роботов от разных производителей



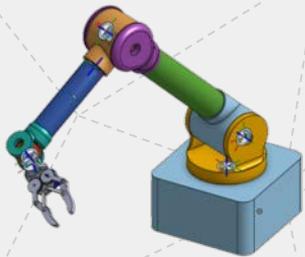
ЗНАЧИМОСТЬ РАБОТЫ

Робот-манипулятор может быть полезен в большом количестве отраслей, поэтому знание и управление таким устройством позволяет автоматизировать рутинные процессы, как на производстве, так и в жизни.

3D модель робота позволяет обучиться принципам автоматизации, не имея реального оборудования, проектировать и моделировать реальные процессы. Процесс управления и программирования такого робота включает знания из различных областей математики, позволяет использовать современные технологии и подходы в современном программировании.

Мета-операционная система ROS позволяет создать собственную инфраструктуру роботу или уже управлять готовыми решениями, за счет разработки собственных сервисов.

Данная разработка может быть использована для обучения студентов или школьников при разработке и автоматизации роботизированных решений. А так же для применения управления промышленными роботами - манипуляторами, где требуется генерация программ на «лету».



СПАСИБО ЗА ВНИМАНИЕ

Шидловская Д.В., Довгулевич Д.А.

