Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Удмуртский государственный университет»

Институт гражданской защиты

Кафедра цифровых инженерных технологий

Лабораторная работа по дисциплине

Прикладное программное обеспечение

**Прямая задача кинематики**

Выполнил студент

группы ОМ-20.04.01.04-11

Вологжанин Егор А.

Оглавление

[Цель 3](#_Toc157088429)

[Задачи 3](#_Toc157088430)

[Пакет и urdf-модель 3](#_Toc157088431)

[Создание пакета 3](#_Toc157088432)

[Urdf-модель 4](#_Toc157088433)

[Лидар и rviz 5](#_Toc157088434)

[Подключение лидара 5](#_Toc157088435)

[Настройка rviz 6](#_Toc157088436)

[Результаты выполнения работы 6](#_Toc157088437)

[Gazebo. 6](#_Toc157088438)

[Rviz. 6](#_Toc157088439)

[Код программы 7](#_Toc157088440)

# Цель

Реализовать движение мобильного робота, используя прямую задачу кинематики, в ROS и Gazebo.

# Задачи

* создать пакет с описанием робота, создать urdf-модель робота.
* реализовать движение робота в среде Gazebo, используя линейную и угловую скорость робота.
* добавить к модели лидар, подготовить среду в симуляторе Gazebo с препятствиями, вывести данные с лидара (rviz).

# Пакет и urdf-модель

## Создание пакета

Любой написанный и запущенный код в рамках ROS должен принадлежать конкретному пакету. Для создания пакета переходим в папку ~/catkin\_ws/src. Применяем команду для создания пакета — catkin\_create\_pkg.

«catkin\_create\_pkg Название\_пакета std\_msgs rospy roscpp»

(std\_msgs, rospy, roscpp — зависимости от других пакетов)

## Urdf-модель

В ROS существует собственная спецификация описания кинематики и динамики робота называемый URDF (Unified Robotic Description Format) и реализуется за счёт файлов с расширением «.urdf». Данный файл пишется с использованием формата XML. В данной работе будет использована продвинутая версия URDF спецификации под названием XACRO (XML macro), который обладает следующими основными преимуществами:

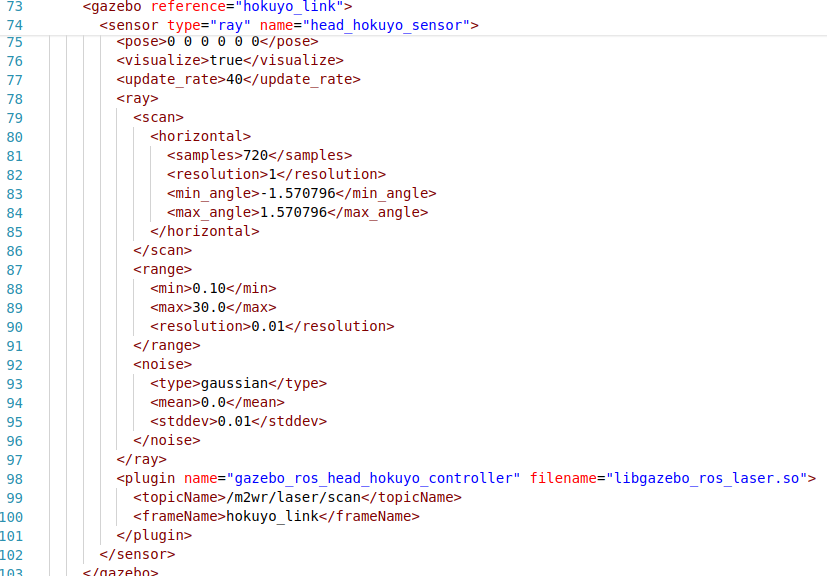
* Встраивание одного файла в другой;
* Инициализация параметров;
* Математический вычисления;
* Описание макросов;

Пример файла urdf.

# Лидар и rviz

## Подключение лидара

В пакете m2wr\_description в файле m2wr.urdf.xacro уже прописаны «линки» и «джойнты» для лидара. Но там есть ссылка на 3Д-модель. Добавим файл модели hokuyo.dae в пакет в папку meshes.

Нужно, чтобы в файле m2wr.gazebo.xacro был прописан код контроллера лидара для gazebo

Код лидара для Gazebo.

## Настройка rviz

Для запуска rviz можно ввести одноимённую команду в консоль — «rviz». При работе с rviz в панели Displays внизу нажимаем кнопку Add, далее выбираем RobotModel. В поле Fixed Frame выбираем base\_link. В Rviz добавляем лидар Add -> LaserScan. Выбираем нужный топик — отображаются точки, там, где препятствия. Препятствия предварительно добавить в gazebo.

# Результаты выполнения работы

## Gazebo.

## Rviz.

# Код программы

#include <ros/ros.h>

#include <geometry\_msgs/Twist.h>

#include <signal.h>

ros::Publisher pub;

geometry\_msgs::Twist vel;

int main(int argc, char \*\*argv)

{

ros::init(argc, argv, "m2wr\_twist", ros::init\_options::NoSigintHandler);

ros::NodeHandle nh;

pub = nh.advertise<geometry\_msgs::Twist>("cmd\_vel", 10);

while (ros::ok())

{

vel.linear.x = 0.3;

vel.angular.z = 0.1;

pub.publish(vel);

ros::spinOnce();

}

return 0;