МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Дисциплина: «**Программирование робототехнических комплексов**».

**Лабораторная работа №1**

**«Простые алгоритмы движения робота»**

Выполнил:

студент 3 курса, гр. ИВТВМбд-31

Захарычев Никита

Проверил:

Преподаватель кафедры ВT

Святов Кирилл Валерьевич.

г. Ульяновск, 2017

**Задание**

В среде ROS (Robotic Operating System) необходимо разработать программу для робота и проверить ее работоспособность во встроенном эмуляторе Gazebo на примере робота TurtleBot, обеспечивающую передвижение робота по фиксированной траектории. Траектория определяется вариантом (который выдается преподавателем). Выбор среды эмуляции (манифест) и вид робота остается на усмотрение студента.

При запуске программы должна появляться среда эмуляции с роботом, для которого должны быть предусмотрены средства манипуляции (движение вперед, назад, влево, вправо, разгон и торможение, запуск движения по траектории). При запуске команды «Запуск движения по траектории» робот начинает движение.

Отчет по выполненной работе должен содержать:

1. Титульный лист.
2. Описание задания по варианту.
3. Описания запускаемых команд
4. Скриншот программы Gazebo, RViz и описание программы и всех функциональных блоков.
5. Листинг программы с комментариями и описание работы программы.
6. Выводы по лабораторной работе.

**Язык –** Python

**5. Вариант -** Движение по n-угольнику (n – количество углов фигуры). Выбран шестиугольник (n =6)

**Описание запускаемых команд**

1. **roslaunch turtlebot\_gazebo turtlebot\_world.launch**

При помощи ROS запускается окружающая среда Gazebo с Turtlebot и другими окружающими его объектами.

1. **roslaunch turtlebot\_rviz\_launchers view\_robot.launch**

Rviz это 3D визуализация среды для ROS. Rviz позволяет видеть, что робот видит, думает и делает. Визуальное оформление и протоколирование информации датчика является важной частью в разработке и отладке.

1. **roslaunch turtlebot\_teleop keyboard\_teleop.launch**

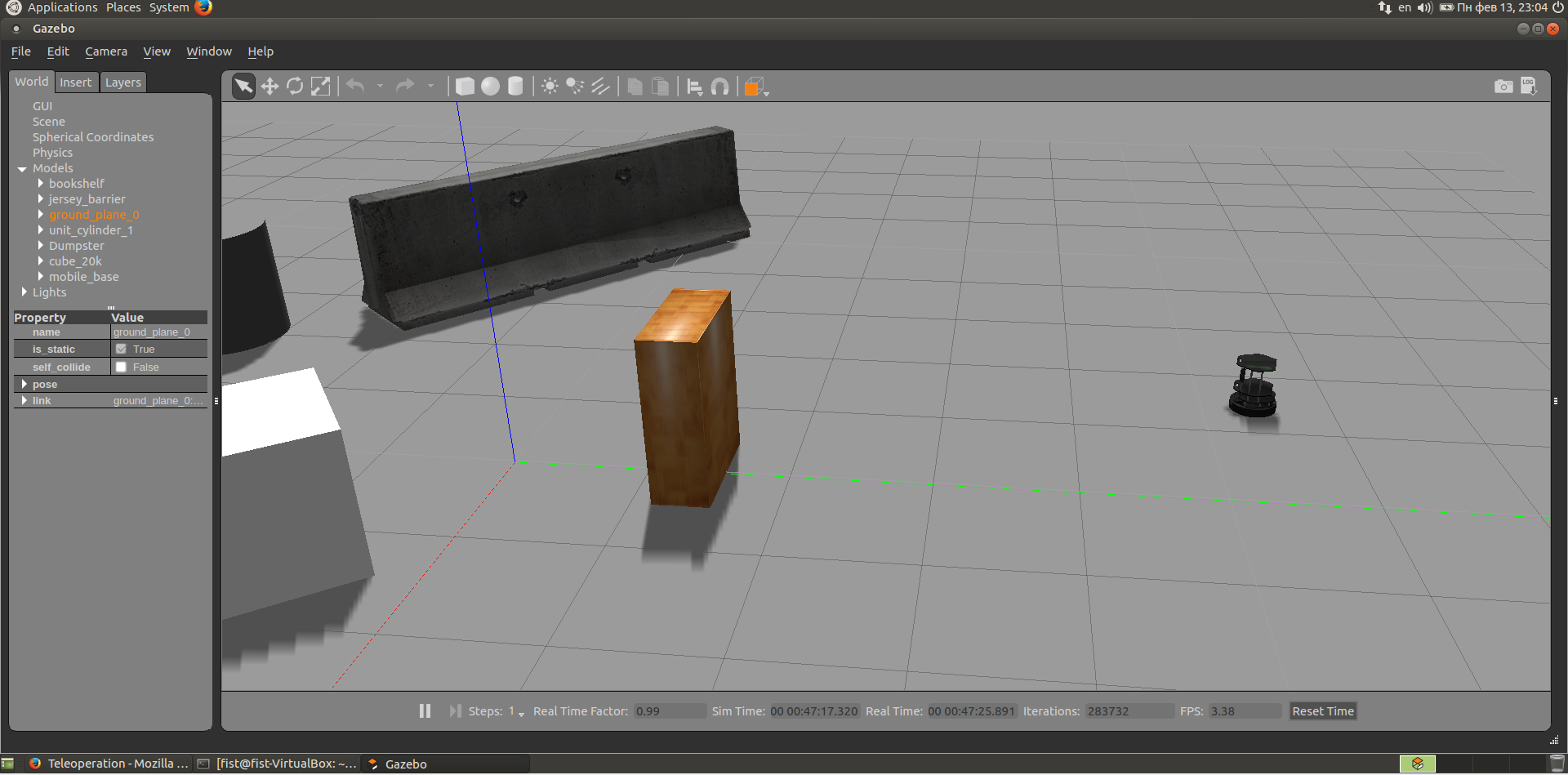
Когда Gazebo и Rviz запущены, можно подключить клавиатуру. После выполнения команды TurtleBot может перемещаться вперед и назад, поворачивать. Также можно увеличить / уменьшить скорость.

1. **Создание скрипта для выполнения задания**

* Создать новую директорию mkdir ~/helloworld
* Зайти в директорию ~/helloworld/
* Скачать по команде git clone https://github.com/markwsilliman/turtlebot/
* Зайти в скачанную папку cd turtlebot
* Запустить скрипт python goforward.py

**Скриншоты программы**

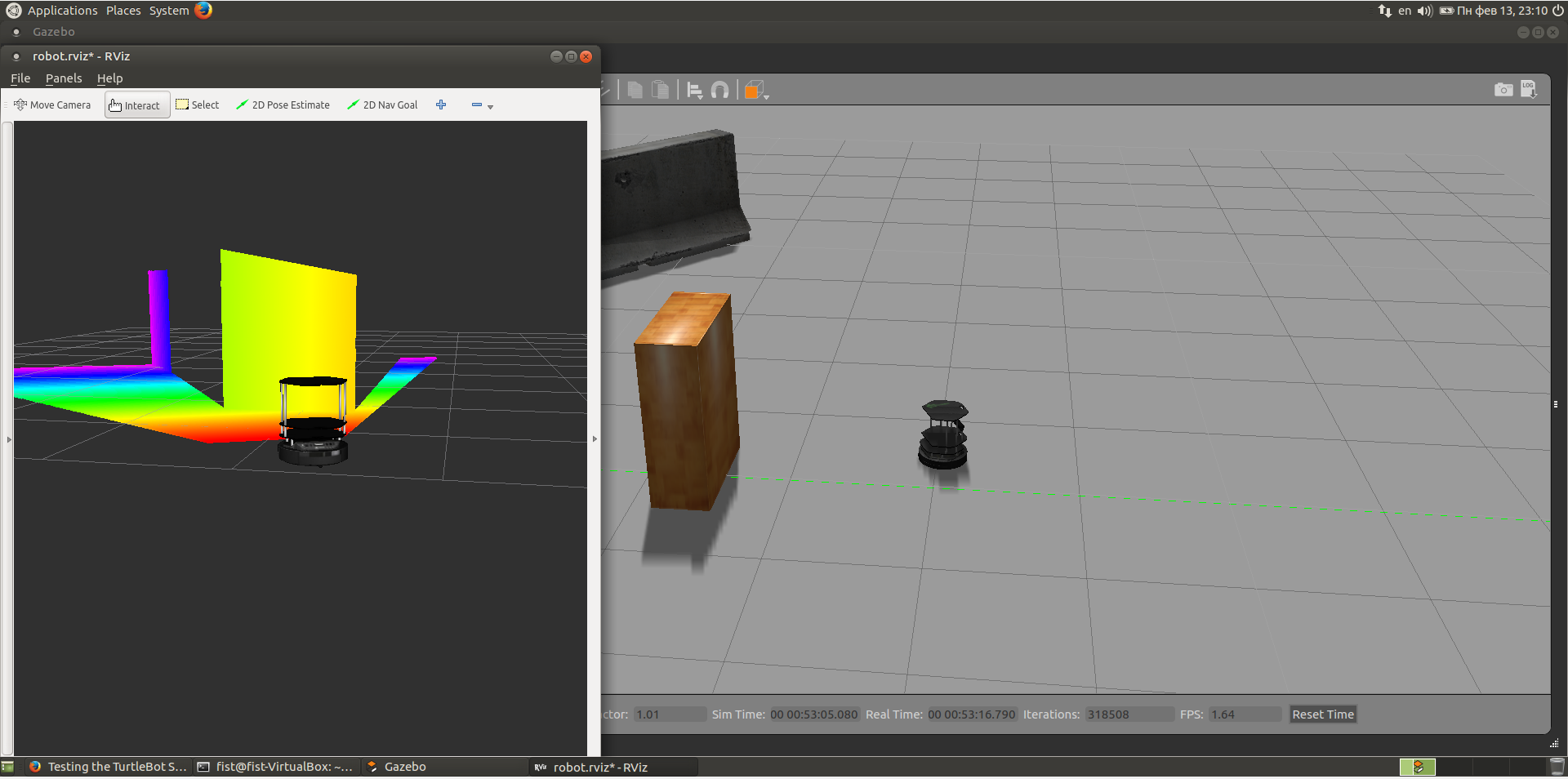
1. Gazebo



Программный комплекс Gazebo представляет собой среду для симулирования работы виртуальных роботов с различными сенсорами в окружении всевозможных объектов. Приложение состоит из графической части и части по имитированию взаимодействия твердых объектов, позволяя моделировать динамику и кинематику механизмов роботов (включая моменты взаимодействия с телами внешней среды) и формировать физически правдоподобные показания виртуальных датчиков. Программа Gazebo имеет гибкий дизайн и удобный интерфейс, поддерживающий одновременную работу с несколькими устройствами. Для работы в данной среде необходимо хорошее знание C++ и UNIX-систем.

Симулятор Gazebo имеет свой собственный редактор, позволяющий без программирования создавать трехмерные сцены и включающий огромную библиотеку моделей. Программа также предоставляет следующие возможности:

1. Использование популярных общеизвестных моделей роботов, таких как: iRobot Create, PR2, TurtleBot, Pioneer 2 DX, Segway RMP, Pioneer 2 AT. Помимо заранее созданных разработчиками моделей есть возможность самостоятельного проектирования необходимых устройств (сенсоров и роботов), загрузки их в мир и дальнейшей симуляции. Однако устройства, модели которых уже есть в программе, эмулируются с гораздо более высокой точностью.
2. Поддержка и симулирование работы множества различных сенсоров, в том числе сонара, лазерного дальномера, датчиков семейства IMU, моно- и стереокамер, кинект-сенсоров, прибора для чтения RFID-меток и других.
3. Gazebo + Rviz



Rviz является мощным инструментом визуализации робота. Это обеспечивает удобный графический интерфейс для визуализации данных датчиков, модели роботов, карты окружающей среды, которые полезны для разработки и отладки контроллеров робота. На картинке мы видим, что датчики помогают видеть препятствия и расстояния до них.

**Листинг программы**

#!/usr/bin/env python

import rospy

from geometry\_msgs.msg import Twist

from math import radians

class DrawFigure():

def \_\_init\_\_(self):

**# Инициализация**

rospy.init\_node('drawasquare', anonymous=False)

**# Если нажимаем ctrl + c – работа завершается**

rospy.on\_shutdown(self.shutdown)

self.cmd\_vel = rospy.Publisher('cmd\_vel\_mux/input/navi', Twist, queue\_size=10)

**# 5 HZ**

r = rospy.Rate(5);

**# Создаем две различных Twist() переменных. Одна для движения вперед. Другая для поворота на 20 градусов.**

**# Движемся со скоростью 0.2 m/s**

move\_cmd = Twist()

move\_cmd.linear.x = 0.2

**# Поворачиваем на 20 градусов deg/s**

turn\_cmd = Twist()

turn\_cmd.linear.x = 0

turn\_cmd.angular.z = radians(20); **#20 deg/s in radians/s**

**#Начинаем реализацию движения**

count = 0

while not rospy.is\_shutdown():

**# Движемся вперед 0.4 m (2 seconds \* 0.2 m / seconds)**

rospy.loginfo("Go!")

for x in range(0,10):

self.cmd\_vel.publish(move\_cmd)

r.sleep()

**# Поворачиваем на 20 градусов**

rospy.loginfo("Turning")

for x in range(0,10):

self.cmd\_vel.publish(turn\_cmd)

r.sleep()

**# Если выполнили 6 поворот с движениями то останавливаемся**

count = count + 1

if(count == 6):

count = 0

if(count == 0):

rospy.loginfo("TurtleBot should be close to the original starting position (but it's probably way off)")

**#Функция обработки нажатия Ctrl+C для остановки программы**

def shutdown(self):

# stop turtlebot

rospy.loginfo("Stop Drawing Squares")

self.cmd\_vel.publish(Twist())

rospy.sleep(1)

**#Если все в порядке выпоняем программу, иначе выдаем ошибку**

if \_\_name\_\_ == '\_\_main\_\_':

try:

DrawFigure()

except:

rospy.loginfo("node terminated.")

**Вывод**

 В данной лабораторной работе было изучен процесс установки и компиляции ROS Keneetic из исходного кода. Сделали мы это с образовательной целью, чтобы немного разобраться с устройством системы ROS и ее команд, а также для того, чтобы подключить последнюю версию симулятора Gazebo 3D. Далее мы загрузили модель робота TurtleBot и настроили простую систему управления. Освоили запуск программ и подприложений, а также средства манипуляции (движение вперед, назад, влево, вправо, разгон и торможение, запуск движения по траектории).