

Day4_hw2

20 기 인턴 우정윤

목차

1. Dualshock 컨트롤러 제어 패키지 제작
2. Joysim 패키지 구현
3. 실행 결과
4. 참고문헌

1. Dualshock 컨트롤러 제어 패키지 제작

우선 조이스틱 제어를 위한 패키지를 설치해 준다.

```
sudo apt install joystick jstest-gtk
```

다음의 패키지를 설치하면 ros2 에서 조이스틱 드라이버가 설치되어 기본적인 동작을 구현할 수 있다. Dualshock 패키지는 Joy_node 가 퍼블리시하는 메시지를 입력받아 표준 속도 명령 메시지인 twist 메시지로 변환하여 cmd_vel 토픽에 퍼블리시 하는 역할을 하게 된다.

따라서 동작 흐름은 다음과 같다.

- 1) 조이스틱 입력
- 2) Joy_node 가 입력 받은 값을 퍼블리쉬
- 3) Ds4 노드가 joy_node 의 값을 서브스크라이브 후 cmd_vel 퍼블리쉬

```

void joy_callback(const sensor_msgs::msg::Joy::SharedPtr msg)
{
    auto twist = geometry_msgs::msg::Twist();

    twist.linear.x = max_lin_ * msg->axes[1];
    twist.angular.z = max_ang_ * msg->axes[0];

    pub_cmd_->publish(twist);

    RCLCPP_INFO(this->get_logger(), "linear.x=%.2f, angular.z=%.2f",
                twist.linear.x, twist.angular.z);
}

```

그리고 이를 변환하는 가장 핵심적인 함수는 다음과 같다.

- 1) twist 메시지를 생성하고
- 2) 조이스틱 입력을 속도로 매핑하고
- 3) 이때 왼쪽 스틱의 y 축이 이동속도
- 4) 왼쪽 스틱의 x 축이 회전속도
- 5) 그리고 이를 퍼블리쉬하고
- 6) 변환된 속도를 디버깅을 위해 로그로 출력

하도록 하였다.

2. Joysim 패키지 구현

조이스틱의 값을 subscribe 하여 각속도와 선속도로 ds4 패키지가 변환하였다면 그 값을 불러와서 cmd_vel 의 값을 받아 가상 좌표계를 적분을 통해 추정하고 이를 메시지와 좌표계로 퍼블리쉬 해야 한다.

따라서 동작 흐름은 다음과 같다.

- 1) Cmd_vel 값 수신
- 2) 좌표계로 변환
- 3) Odom 에 송신

```
void cmd_callback(const geometry_msgs::msg::Twist::SharedPtr msg)
{
    rclcpp::Time current_time = this->now();
    double dt = (current_time - last_time_).seconds();
    last_time_ = current_time;

    double vx = msg->linear.x;
    double vth = msg->angular.z;

    x_ += vx * cos(theta_) * dt;
    y_ += vx * sin(theta_) * dt;
    theta_ += vth * dt;

    tf2::Quaternion q;
    q.setRPY(0, 0, theta_);

    geometry_msgs::msg::TransformStamped t;
    t.header.stamp = current_time;
    t.header.frame_id = "map";
    t.child_frame_id = "odom";
    t.transform.translation.x = x_;
    t.transform.translation.y = y_;
    t.transform.translation.z = 0.0;
    t.transform.rotation.x = q.x();
    t.transform.rotation.y = q.y();
    t.transform.rotation.z = q.z();
    t.transform.rotation.w = q.w();

    tf_broadcaster_->sendTransform(t);

    RCLCPP_INFO(this->get_logger(), "Odom updated: x=%.2f, y=%.2f, theta=%.2f",
                x_, y_, theta_);
}
```

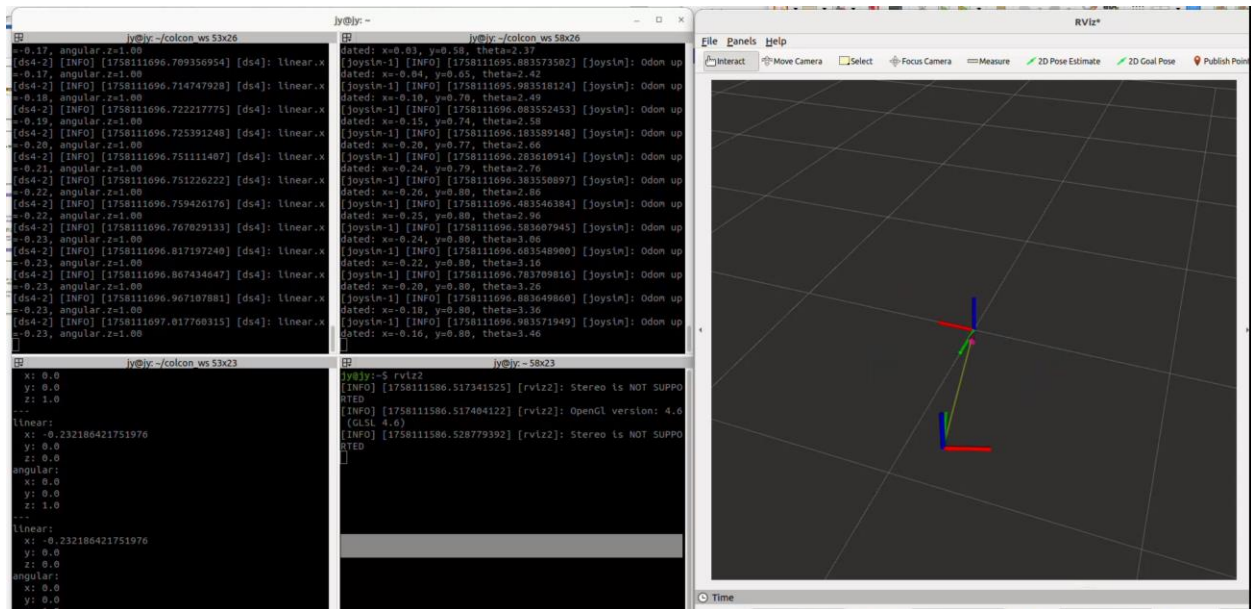
Joysim 패키지에서 가장 중요한 함수 역시 콜백 함수이다. 콜백 함수는 다음과 같이 구성되는데

- 1) Cmd_vel 메시지가 들어올때마다 호출되고
- 2) Linear.x, angular.z 값을 읽는다.

- 3) 좌표를 적분하고 세타 값으로 변환한다.
- 4) Transforstamped 메시지를 생성한다.
- 5) A 좌표계에서 b 좌표계 까지의 위치와 회전을 담은 메시지 형식이고
- 6) 기준 평면에서 이동한 좌표계가 어디에 있는지 나타낸다.

3. 패키지 동시 실행 및 결과 확인

각각의 패키지를 빌드하고 노드를 실행하여 주면 다음과 같이 rviz2 에서 실행하였을때에 정상적으로 시뮬레이션 되는 것을 볼 수 있다.



4. 참고 문헌

1) https://docs.ros.org/en/humble/p/geometry_msgs/

2) https://docs.ros.org/en/humble/p/nav_msgs/

3) <https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Cpp.html>

4) <https://docs.ros.org/en/foxy/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Cpp.html>

5) <https://index.ros.org/p/joy/>