

LIMO 함수 기반 제어

WeGo & LIMO

목 차

1. 함수 기반 제어
2. 사용 방법
3. 코드 설명
4. 예제 설명

01

함수 기반
제어

01 함수 기반 제어

- 함수 기반 제어
 - https://github.com/WeGo-Robotics/limo_examples
 - limo_examples/scripts/function_based_examples 아래에 포함
 - LIMO를 더 쉽게 접근해서 사용해보기 위한 모듈형 코드
 - 파이썬에 대한 기초 지식이 필요
 - 아래의 내용 포함
 - 기본적인 LIMO의 이동에 대한 제어
 - LiDAR 데이터 받기
 - Image 데이터 받기
 - 기타, 센서 데이터 기반의 제어 예제

02

사용 방법

02 사용 방법

- 적용 방법
 - `$ cd agilex_ws/src`
 - `$ git clone https://github.com/WeGo-Robotics/limo_examples.git`
 - `$ cd .. && catkin_make`
 - `$ source devel/setup.bash`
 - `$ rospack profile`

02 사용 방법

- 예제 사용 방법
 - LIMO 내부에서 LIMO, LiDAR, Camera에 대한 Driver 실행 후, 추후 내용 진행
 - `$ roslaunch limo_bringup limo_start.launch`
 - `$ roslaunch astra_camera dabai_u3.launch`
 - Example 코드 실행 (하나씩만 실행, 동시 실행 불가능)
 - `$ rosrun limo_examples move_limo.py`
 - `$ rosrun limo_examples get_image_data.py`
 - `$ rosrun limo_examples get_lidar_data.py`
 - `$ rosrun limo_examples stop_using_lidar.py`
 - `$ rosrun limo_examples patrol_using_lidar.py`

03

코드 설명

03 코드 설명

- 사용하는 함수들이 포함된 utils.py에 대한 상세 문서는 아래 링크를 통해 확인할 수 있습니다.
- https://docs.google.com/presentation/d/10CO-hCtMiG7AYE_dBP_7NSkoYQYM-HxvswAqN1BDSGE/edit?usp=sharing
- 본 문서에서는 해당 모듈을 사용한 코드 작성 방법에 대해 설명합니다.
- 추가로 작성하시는 코드는 필수적으로
~/agilex_ws/src/limo_examples/scripts 폴더 내부에 위치해야 합니다.

03 코드 설명

- 최상단(아래 입력한 내용이 필수적으로 코드 시작 부분에 위치)
 - 추가 모듈
 - ROS 내부 출력 및 ROS 사용을 위한 rospy
 - 함수 기반 제어를 위한 utils
 - 코드 내부 시간 제어를 위한 time
 - 이 외 동작을 위한 추가 모듈 추가가 필요합니다.

```
import rospy
```

```
import utils
```

```
import time
```

03 코드 설명

- 코드 시작 부분
 - `utils.startApplication()` 메서드를 사용해서 초기화를 시켜줘야 정상적으로 이외의 메서드를 사용할 수 있습니다.

```
import rospy
```

```
import utils
```

```
import time
```

```
utils.startApplication()
```

03 코드 설명

- 코드 중간 부분
 - utils 내부의 메서드를 적용하여, 동작하는 알고리즘을 작성합니다.
 - ex) 데이터 수신, 비교 후, 움직이기 등

03 코드 설명

- 코드 마지막 부분 (권장 사항)
 - `rospy.loginfo("limo application end")` 를 입력하여, 코드가 정상적으로 종료되었음을 알리는 문구를 입력
 - 코드가 중간에 멈추었는지, 종료되었는지 확인을 위해 해당 문구를 추가

```
import rospy
import utils
import time
```

```
utils.startApplication()
```

```
## =====
```

```
## 중간 부분 코드를 여기 부분에 작성
```

```
## =====
```

```
rospy.loginfo("limo application end")
```

04

예제 설명

04 예제 설명

- move_limo.py
 - LIMO 움직임을 위한 간단 코드
 1. 전진, 후진 (utils.goStraight)
 2. 좌회전, 우회전 (utils.turn)
 3. 직진 및 좌회전, 직진 및 우회전 (utils.moveCmd)
 4. 후진 및 우회전, 후진 및 좌회전 (utils.moveCmd)
 - utils 내부의 goStraight, turn, moveCmd 세 개의 메서드를 사용하는 방법을 제시하고 있음

04 예제 설명

- move_limo.py

```
#!/usr/bin/env python
import rospy
import utils
import time
utils.startApplication()

utils.goStraight(_linear_speed=0.2) # 0.2 m/s for linear_speed (move forward)
time.sleep(2) # wait until the move end
utils.goStraight(_linear_speed=-0.2) # -0.2 m/s for linear_speed (move backward)
time.sleep(2) # wait until the move end
utils.turn(_angular_speed=0.5, _unit="rad") # 0.5 rad/s for angular_speed (turn left)
time.sleep(2) # wait until the move end
utils.turn(_angular_speed=-0.5, _unit="rad") # -0.5 rad/s for angular_speed (turn right)
time.sleep(2)
utils.moveCmd(_linear_speed=0.2, _angular_speed=0.5, _unit="rad")
# 0.2 m/s for linear, 0.5 rad/s for angular, forward left
time.sleep(2)
utils.moveCmd(_linear_speed=0.2, _angular_speed=-0.5, _unit="rad")
# 0.2 m/s for linear, -0.5 rad/s for angular, forward right
time.sleep(2)
utils.moveCmd(_linear_speed=-0.2, _angular_speed=0.5, _unit="rad")
# -0.2 m/s for linear, 0.5 rad/s for angular, backward right
time.sleep(2)
utils.moveCmd(_linear_speed=-0.2, _angular_speed=-0.5, _unit="rad")
# 0.2 m/s for linear, -0.5 rad/s for angular, backward left
time.sleep(2)

rospy.loginfo("limo application end")
```


04 예제 설명

- `get_lidar_data.py`
 - LiDAR 센서 데이터를 받기 위한 간단 코드
 1. 전체 라이다 데이터를 받기 (`utils.getLidar`)
 2. 특정 각도의 거리 값을 받기 (`utils.getLidarRangeUsingAngle`)
 3. 해당 각도와 거리를 출력 (`print`)
 4. 1 ~ 3의 내용을 반복
 - `utils` 내부의 `getLidar`, `getLidarRangeUsingAngle` 메서드의 사용 방법을 제시하고 있음
 - 라이다의 특정 각도의 값을 지속적으로 받을 수 있도록 동작
 - 코드 실행 종료를 위해서는 실행한 터미널에서 `Ctrl + C` 입력

04 예제 설명

- get_lidar_data.py

```
#!/usr/bin/env python
```

```
import rospy
```

```
import utils
```

```
import time
```

```
utils.startApplication()
```

```
while not rospy.is_shutdown():
```

```
    lidar_data = utils.getLidar()
```

```
    angle, center_range = utils.getLidarRangeUsingAngle(_lidar=lidar_data, _angle=0.0, _type="deg")
```

```
    print("{:.4f} degree, {:.4f} meter".format(angle, center_range))
```

```
rospy.loginfo("limo application end")
```

04 예제 설명

- `get_image_data.py`
 - Camera 센서 데이터를 받기 위한 간단 코드
 1. Image 데이터를 받기 (`utils.getImage`)
 2. Image를 화면에 출력 (`utils.showOpenCvImage`)
 3. 1 ~ 2의 내용을 반복
 - `utils` 내부의 `getImage`, `showOpenCvImage` 메서드의 사용 방법을 제시하고 있음
 - 지속적으로 카메라의 이미지를 받을 수 있도록 동작
 - 코드 실행 종료를 위해서는 실행한 터미널에서 `Ctrl + C` 입력

04 예제 설명

- get_image_data.py

```
#!/usr/bin/env python

import rospy

import utils

import time

utils.startApplication()

while not rospy.is_shutdown():
    image_data = utils.getImage(_type="opencv")
    utils.showOpenCvImage(image_data)

rospy.loginfo("limo application end")
```

04 예제 설명

- stop_using_lidar.py
 - LiDAR 데이터를 받아서, 충돌 전에 멈추기 위한 코드
 1. 전체 라이다 데이터를 받기 (utils.getLidar)
 2. 정면의 거리 값을 받기 (utils.getLidarRangeUsingAngle)
 3. 거리 값이 0.5 (meter) 아래로 내려가면 멈추기, 아닐 경우 직전
 4. 1 ~ 3의 내용을 반복
 - utils 내부의 getLidar, getLidarRangeUsingAngle, goStraight 메서드의 사용 방법을 제시하고 있음
 - 지속적으로 정면의 거리 데이터를 받고, 가까워지면 차량이 멈추게 함
 - 코드 실행 종료를 위해서는 실행한 터미널에서 Ctrl + C 입력

04 예제 설명

- stop_using_lidar.py

```
#!/usr/bin/env python
```

```
•
```

```
import rospy
```

```
import utils
```

```
import time
```

```
utils.startApplication()
```

```
while not rospy.is_shutdown():
```

```
    lidar_data = utils.getLidar()
```

```
    angle, center_range = utils.getLidarRangeUsingAngle(_lidar=lidar_data, _angle=0.0, _type="deg")
```

```
    if center_range >= 0.5 or center_range == 0.0:
```

```
        utils.goStraight(_linear_speed=0.2)
```

```
        rospy.loginfo("Center is clear, GO")
```

```
    else:
```

```
        utils.goStraight(_linear_speed=0.0) # means STOP
```

```
        rospy.loginfo("Blocked by Something, STOP")
```

```
rospy.loginfo("move_limo application end")
```

04 예제 설명

- patrol_using_lidar.py
 - LiDAR 데이터를 받아서, 충돌 전에 다른 방향으로 회전해서 이동하는 코드
 1. 전체 라이다 데이터를 받기 (utils.getLidar)
 2. -60, 0, 60도의 거리 값을 받기 (utils.getLidarRangeUsingAngle)
 3. 세 개의 거리 값 중, 최소가 되는 값과 각도를 확인
 4. 최소 거리가 0.5 (meter) 이하가 될 경우, 정지
 - a. 왼쪽 각도가 최소 거리인 경우, 오른쪽으로 회전
 - b. 오른쪽 각도가 최소 거리인 경우, 왼쪽으로 회전
 - c. 정면의 각도가 최소 거리인 경우, 랜덤하게 회전
 5. 1 ~ 4의 과정을 반복
 - utils 내부의 getLidar, getLidarRangeUsingAngle, goStraight, moveCmd 메서드의 사용 방법을 제시하고 있음
 - 코드 실행 종료를 위해서는 실행한 터미널에서 Ctrl + C 입력

04 예제 설명

- patrol_using_lidar.py

```
#!/usr/bin/env python

import rospy

import utils

import time
import random

utils.startApplication()

angles = [-60.0, 0.0, 60.0]

while not rospy.is_shutdown():
    center_ranges = []
    lidar_data = utils.getLidar()
    for angle in angles:
        angle, center_range = utils.getLidarRangeUsingAngle(_lidar=lidar_data, _angle=angle, _type="deg")
        center_ranges.append(center_range)

    min_idx = 0
    min_val = 100.0
    for i, range in enumerate(center_ranges):
        if range < min_val:
            min_idx = i
            min_val = range
```


04 예제 설명

- patrol_using_lidar.py

```
if min_val == 100.0:
    utils.goStraight(_linear_speed=0.2)
    rospy.loginfo("Clear, GO")
elif min_val < 0.5 and not min_val == 0.0:
    utils.goStraight(_linear_speed=0.0) # means STOP
    rospy.loginfo("Blocked by Something, STOP")
    time.sleep(3)
if min_idx == 0:
    # utils.moveCmd(_linear_speed=0.1, _angular_speed=1.0)
    move_command = "LEFT"
    rospy.loginfo("Right is Blocked, TURN Left")
elif min_idx == 1:
    # utils.moveCmd(_linear_speed=0.1, _angular_speed=2.0)
    move_command = "TURN"
    rospy.loginfo("Center is Blocked, TURN")
elif min_idx == 2:
    # utils.moveCmd(_linear_speed=0.1, _angular_speed=-1.0)
    move_command = "RIGHT"
    rospy.loginfo("Left is Blocked, TURN Right")
start_time = rospy.Time.now().to_sec()
if random.choice([True, False]):
    tmp_speed = 0.7 + random.random() * 0.5
else:
    tmp_speed = -0.7 - random.random() * 0.5

while rospy.Time.now().to_sec() - start_time <= 3.0:
    if move_command == "LEFT":
        utils.moveCmd(_linear_speed=0.0, _angular_speed=.7)
    elif move_command == "RIGHT":
        utils.moveCmd(_linear_speed=0.0, _angular_speed=-.7)
    elif move_command == "TURN":
        utils.moveCmd(_linear_speed=0.0, _angular_speed=tmp_speed)
```



WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554



제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com