Multi-car keyboard control

Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be the same. You can check [Must-read before use] to set the IP and ROS_DOMAIN_ID on the board.

1. Program function description

After the program is started, the two cars can be controlled to move synchronously through the keyboard.

2. Basic settings for multi-machine functions

Take two cars as an example. It is recommended to use two computers with matching virtual machines, change the config_robot.py files respectively, set robot.set_ros_namespace() to robot1 and robot2 respectively; set robot.set_udp_config() to the IP addresses of the two virtual machines respectively, and **the ROS_DOMAIN_ID of the two cars and the ROS_DOMAIN_ID of the virtual machine need to be set to the same**. Then open the terminal in the /home/yahboom directory and enter `sudo python3 config_Balance_Car.py` to run this program (you need to change the rest of the programs other than running multiple cars and rerun this program).

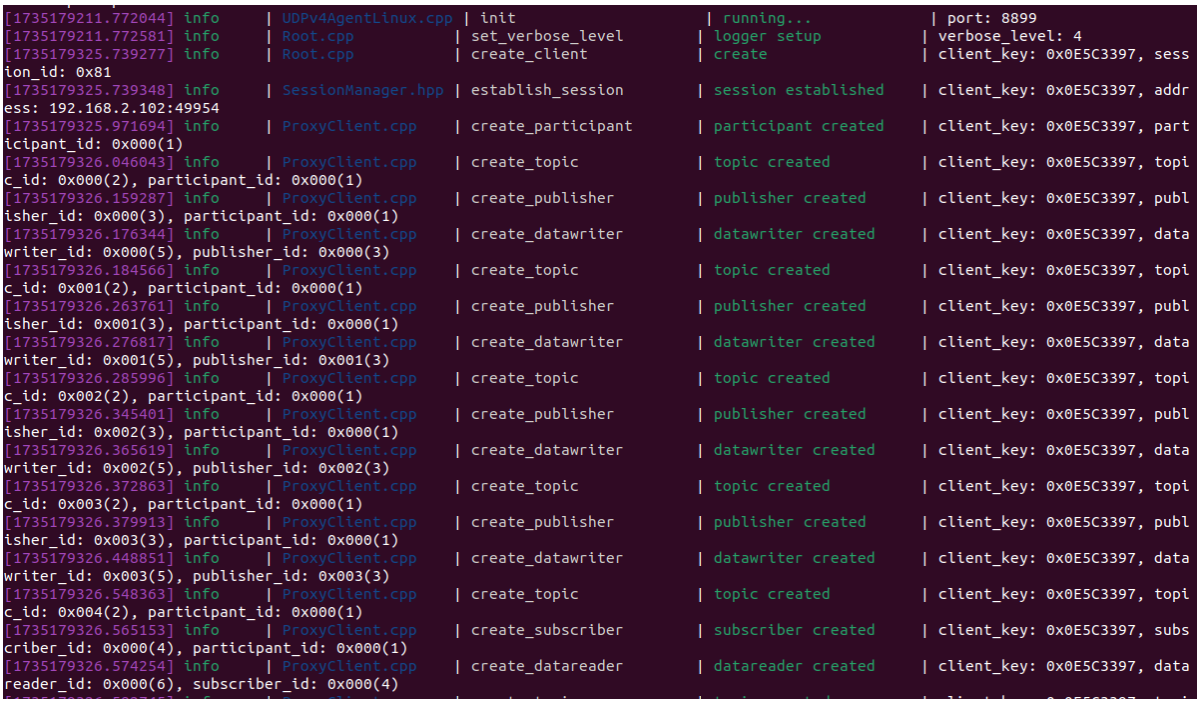
3. Start and connect the agent

Take the matching virtual machine as an example. In the two virtual machines, enter the following command to start the agent of each car.

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --
net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
```

```
yahboom@yahboom-VM:~$ sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm
 --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
[1735179211.772044] info     | UDPv4AgentLinux.cpp | init                     |
running...              | port: 8899
[1735179211.772581] info     | Root.cpp           | set_verbose_level        | l
ogger setup             | verbose_level: 4
```

Then, turn on the switches of the two cars and wait for the two cars to connect to their respective agents. If the connection is successful, the terminal display is as shown in the figure below.



Check the currently started node. In the two virtual machines, randomly select one and open the terminal to enter,

```
ros2 node list
```

```
yahboom@yahboom-VM:~$ ros2 node list
WARNING: Be aware that are nodes in the graph that share an exact name, this can
 have unintended side effects.
/robot2/YB_BalanceCar_Node
/robot2/YB_BalanceCar_Node
```

As shown in the figure above, the nodes of both cars have been started. Query the current topic information, input in the terminal,

```
ros2 topic list
```

```
yahboom@yahboom-VM:~$ ros2 topic list
/parameter_events
/robot1/beep
/robot1/cmd_vel_bl
/robot1/imu
/robot1/mpuimu
/robot1/odom_raw
/robot1/scan
/robot2/beep
/robot2/cmd_vel_bl
/robot2/imu
/robot2/mpuimu
/robot2/odom_raw
/robot2/scan
/rosout
```

4. Start the keyboard control program

Take the matching virtual machine as an example, select one of the two virtual machines at random, open the terminal and input,

```
ros2 run yahboomcar_multi multi_keyboard_ctrl
```

```
yahboom@yahboom-VM:~$ ros2 run yahboomcar_multi multi_keyboard_ctrl

Control Your SLAM-Bot!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
t/T : x and y speed switch
s/S : stop keyboard control
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 20.0      turn 300.0
```

The keyboard key description is as follows,

Direction control:

Key	Linear angular velocity	Direction	Key	Linear angular velocity	Direction
【i】or 【I】	【linear，0】	Go forward	【u】or 【U】	【linear，angular】	Turn left

Key	Linear angular velocity	Direction	Key	Linear angular velocity	Direction
【,】	【-linear，0】	Go backward	【o】or 【O】	【linear，-angular】	Turn right
【j】or 【J】	【0，angular】	Rotate left	【m】or 【M】	【-linear，-angular】	Reverse left
【l】or 【L】	【0，-angular】	Rotate right	【.】	【-linear，angular】	Reverse right

Speed control:

Button	Speed change	Button	Speed change
【q】	Both linear and angular velocities increase by 10%	【z】	Both linear and angular velocities decrease by 10%
【w】	Only linear velocity increases by 10%	【x】	Only linear velocity decreases by 10%
【e】	Only angular velocity increases by 10%	【c】	Only angular velocity decreases by 10%
【t】	Linear velocity X-axis/Y-axis direction switching	【s】	Stop keyboard control

Note: Since the car is a two-wheel drive structure with ordinary tires and cannot move sideways, the 【t】 key is meaningless. Before using keyboard control each time, you need to click the terminal to start the program, otherwise the key event cannot be detected.

5. Code analysis

Source code reference path (taking the matching virtual machine as an example):

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/yahboomcar_multi
```

multi_yahboom_keyboard.py,

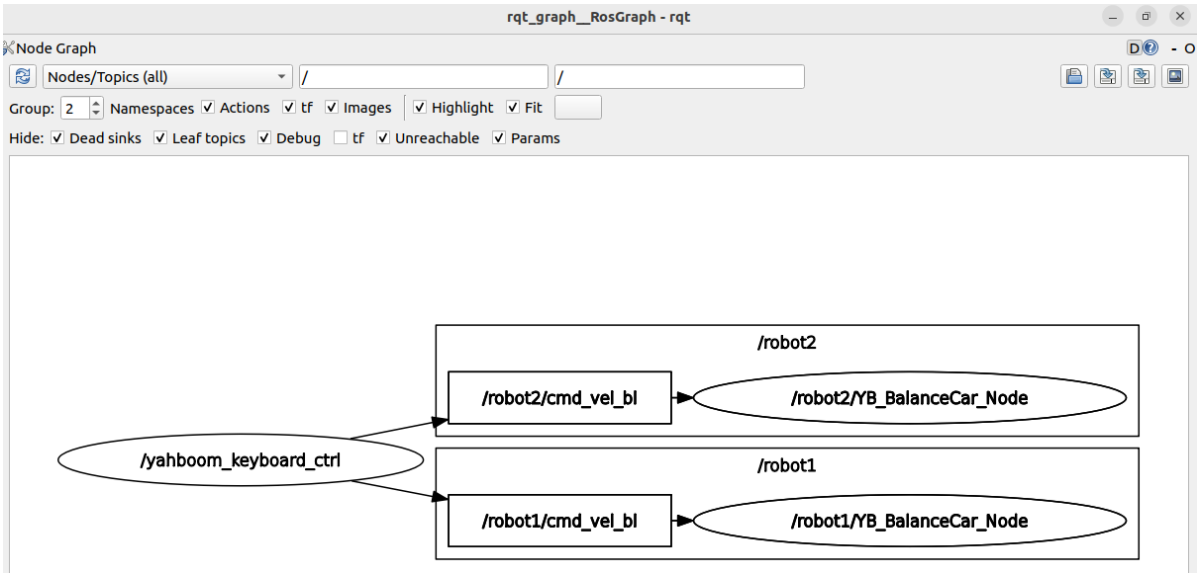
```
#Create two speed publishers
self.pub_robot1 = self.create_publisher(Twist,'/robot1/cmd_vel_bl',1000)
self.pub_robot2 = self.create_publisher(Twist,'/robot2/cmd_vel_bl',1000)
#Get key events
def getKey(self):
    tty.setraw(sys.stdin.fileno())
    rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
    if rlist: key = sys.stdin.read(1)
else: key = ''
termios.tcsetattr(sys.stdin, termios.TCSADRAIN, self.settings)
return key
#Analyze key events
```

```
while (1):
    key = yahboom_keyboard.getKey()
    if key=="t" or key == "T": xspeed_switch = not xspeed_switch
    elif key == "s" or key == "S":
    print ("stop keyboard control: {}".format(not stop))
    stop = not stop
    if key in moveBindings.keys():
        x = moveBindings[key][0]
        th = moveBindings[key][1]
        count = 0
    elif key in speedBindings.keys():
        speed = speed * speedBindings[key][0]
        turn = turn * speedBindings[key][1]
        count = 0
        if speed > yahboom_keyboard.linenar_speed_limit:
            speed = yahboom_keyboard.linenar_speed_limit
            print("Linear speed limit reached!")
            if turn > yahboom_keyboard.angular_speed_limit:
                turn = yahboom_keyboard.angular_speed_limit
                print("Angular speed limit reached!")
                print(yahboom_keyboard.vels(speed, turn))
                if (status == 14): print(msg)
                status = (status + 1) % 15
            elif key == ' ': (x, th) = (0, 0)
        else:
            count = count + 1
            if count > 4: (x, th) = (0, 0)
            if (key == '\x03'): break
#Publish the car speed
yahboom_keyboard.pub_robot1.publish(robot1_twist)
yahboom_keyboard.pub_robot2.publish(robot2_twist)
```

6. View the node communication graph

Select any of the two virtual machines, open the terminal and enter,

```
ros2 run rqt_graph rqt_graph
```



If it is not displayed at first, select [Nodes/Topics(all)] and click the refresh button in the upper left corner.