

## 3、Keyboard control

### 3、Keyboard control

3.1、teleop\_twist\_keyboard.py

3.2、transbot\_keyboard.py

Function package path: ~/transbot\_ws/src/transbot\_ctrl

Input following command to start up:

```
roslaunch transbot_bringup bringup.launch
```

### 3.1、teleop\_twist\_keyboard.py

Wiki: [http://wiki.ros.org/teleop\\_twist\\_keyboard](http://wiki.ros.org/teleop_twist_keyboard)

Source code: [https://github.com/ros-teleop/teleop\\_twist\\_keyboard](https://github.com/ros-teleop/teleop_twist_keyboard)

This feature pack can be installed directly into the system.

- Install

```
sudo apt-get install ros-melodic-teleop-twist-keyboard
```

- Run

```
roslaunch teleop_twist_keyboard teleop_twist_keyboard.py
```

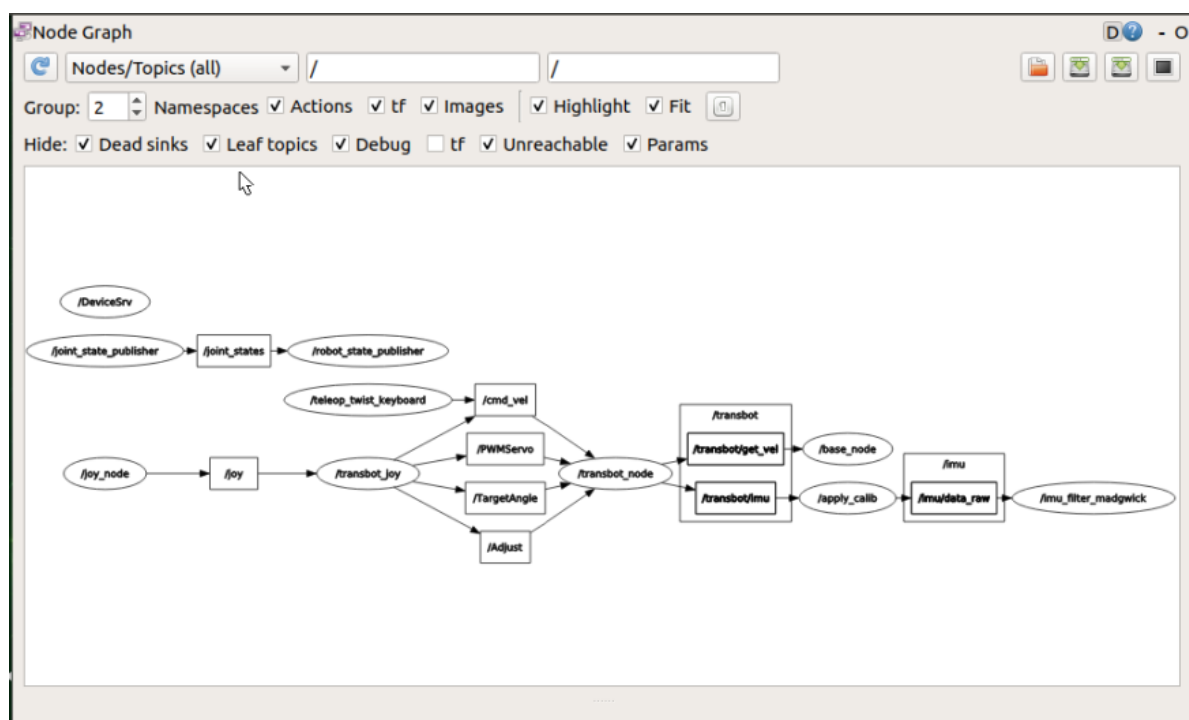
- Control

Key	Car[linear, angular]	Key	Car[linear, angular]
<b>【i】</b> or <b>【I】</b>	<b>【 linear, 0】</b>	<b>【u】</b> or <b>【U】</b>	<b>【linear, angular】</b>
<b>【,】</b>	<b>【-linear, 0】</b>	<b>【o】</b> or <b>【O】</b>	<b>【linear, - angular】</b>
<b>【j】</b> or <b>【J】</b>	<b>【0, angular】</b>	<b>【m】</b> or <b>【M】</b>	<b>【- linear, - angular】</b>
<b>【l】</b> or <b>【L】</b>	<b>【0, - angular】</b>	<b>【.】</b>	<b>【 - linear, angular】</b>
<b>Key</b>	<b>Speed change</b>	<b>Key</b>	<b>Speed change</b>

Key	Car[linear, angular]	Key	Car[linear, angular]
【q】	Linear velocity and angular velocity are both increased by 10%	【z】	linear velocity and angular velocity are both reduced by 10%
【w】	Only the linear velocity increased by 10%	【x】	Only the linear velocity reduced by 10%
【e】	Only the angular velocity increased by 10%	【c】	Only the angular velocity reduced by 10%

Except for the above keys, any key stops the movement. 【Ctrl】 + 【c】 Exit.

rqt\_graph



The node [teleop\_twist\_keyboard] publishes a message to the topic [/cmd\_vel] and is subscribed by the node [/transbot\_node].

## 3.2、transbot\_keyboard.py

**Note:** The key control method is the same as above

Input following command to start up

```
roslaunch transbot_ctrl transbot_keyboard.launch
```

Code analysis

Mainly use select module, termios module and tty module

```
import sys, select, termios, tty
```

- The select module is mainly used for socket communication.
- The termios module provides an IO-controlled POSIX call interface for tty
- The tty module is mainly used to change the mode of the file descriptor fd

Get current key information

```
def getKey():
    # tty.setraw():Change the file descriptor fd mode to raw; fileno(): returns
    # an integer file descriptor (fd)
    tty.setraw(sys.stdin.fileno())
    # select():Directly call the IO interface of the operating system; monitor
    # all file handles with fileno() method
    rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
    # Read a byte of input stream
    if rlist: key = sys.stdin.read(1)
    else: key = ''
    # tcsetattr sets the tty attribute of the file descriptor fd from the
    # attribute
    termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
    return key
```

Get speed limit

```
linear_limit = rospy.get_param('~linear_limit', 0.45)
angular_limit = rospy.get_param('~angular_limit', 2.0)
```

Control flow

```
# Get current key information
key = getKey()
# Key string to determine whether it is in the dictionary
if key in moveBindings.keys():
    x = moveBindings[key][0]
    th = moveBindings[key][1]
    count = 0
# Key string to determine whether it is in the dictionary
elif key in speedBindings.keys():
    speed = speed * speedBindings[key][0]
    turn = turn * speedBindings[key][1]
    count = 0
# speed limit
if speed > linear_limit: speed = linear_limit
if turn > angular_limit: turn = angular_limit
print(vels(speed, turn))
# Print msg information once accumulated a certain number of
times

if (status == 14): print(msg)
status = (status + 1) % 15
# If the button is ' ' or 'k', then stop the movement
elif key == ' ': (x, th) = (0, 0)
else:
    # If it is not a long press, stop this function
    count = count + 1
    if count > 4: (x, th) = (0, 0)
```

```
        if (key == '\x03'): break
    # Publish
    twist = Twist()
    twist.linear.x = speed * x
    twist.angular.z = turn * th
    pub.publish(twist)
```