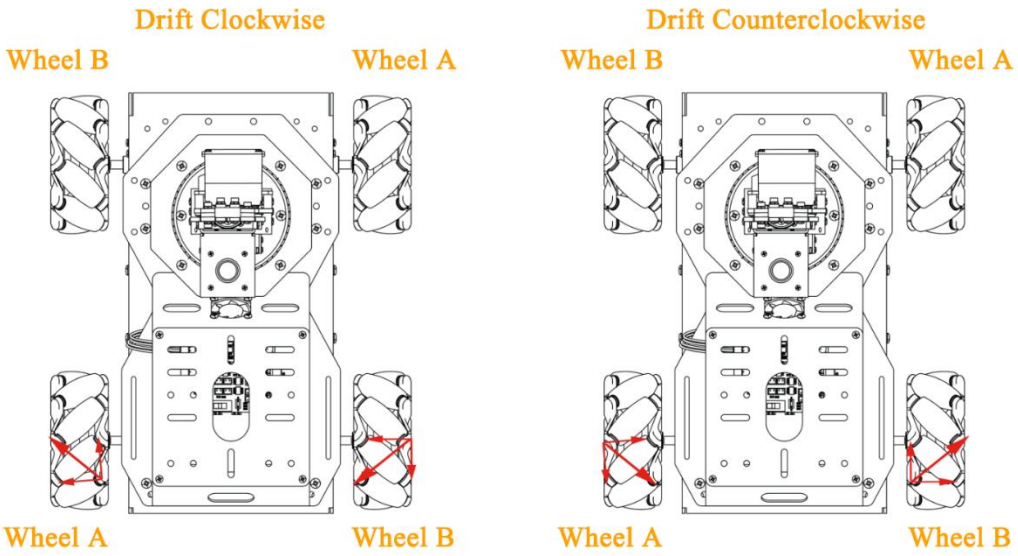
Lesson 7 Drifting Movement

1. Working Principle

According to the characteristic of mecanum wheel characteristics, when the front two wheels do not move, the rear wheel A rotates clockwise and the rear wheel B rotates counterclockwise, the car drift clockwise. The force analysis for car drifting is shown in the following figure:



According to physical kinematics, when forces are equal and opposite to each other, they will counteract each other. Any force can be decomposed into two perpendicular vectors. Take counterclockwise drifting as example. Suppose the speed of wheel A and wheel B rotates at the same speed, a upward force decomposed by wheel A and a downward force decomposed by wheel B will counteract each other, which the direction of resultant velocity is to the right.

Based on Newton’s second law (F=ma), if the direction of acceleration is to the right, so the final resultant force is also to the right. At this time, if the front wheel does not move, the car will drift.

2. Operation Steps

It should be case sensitive when entering command, and the “Tab” key can be used to complete the keywords.

1) Please refer to the content in “7. ArmPi Pro Basic Lesson/1.Mecanum Wheel Chassis Lesson/ Lesson 2 Set Environment Development” to connect system desktop via No Machine.

2) Click 

in the lower left corner and select 

to enter the terminal.

3) Enter command “**cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/**” and press “Enter” to enter the directory of game programmings.

```
ubuntu@ubuntu:~$ cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/
```

4) Enter command “python3 car_drifting_demo.py” and press “Enter” to start game.

```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/chassis_control_demo$
python3 car_drifting_demo.py
```

5) If want to exit the game, press “Ctrl+C” in terminal. If fail to exit, please keep trying until the program is closed.

3. Project Outcome

After starting game, ArmPi Pro will drift counterclockwise first, then clockwise.

4. Function Extension

The default speed of rotation is 0.3. This section will modify the rotation speed of clockwise drifting to 0.5 and the specific operation steps are as follow:

1) Click 

and select 

to enter the terminal.

2) Enter command "**cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/**" and press "Enter" to come to the directory of game programmings.

```
ubuntu@ubuntu:~$ cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/
```

3) Enter "**vim car_drifting_demo.py**" command and press "Enter" to open the program file

```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/chassis_control_demo$
vim car_drifting_demo.py
```

4) Find the code to be modified and press "i". When the prompt "INSERT" appears in the lower left corner, it means the terminal has been switched to the editing mode.

```
43          set_velocity.publish(60,180,-0.5) # The linear velocity is 60;
    The directional angel is 180; The yaw rate is -0.3 (When the value is
    negative, it will rotate clockwise. )
:wq
```

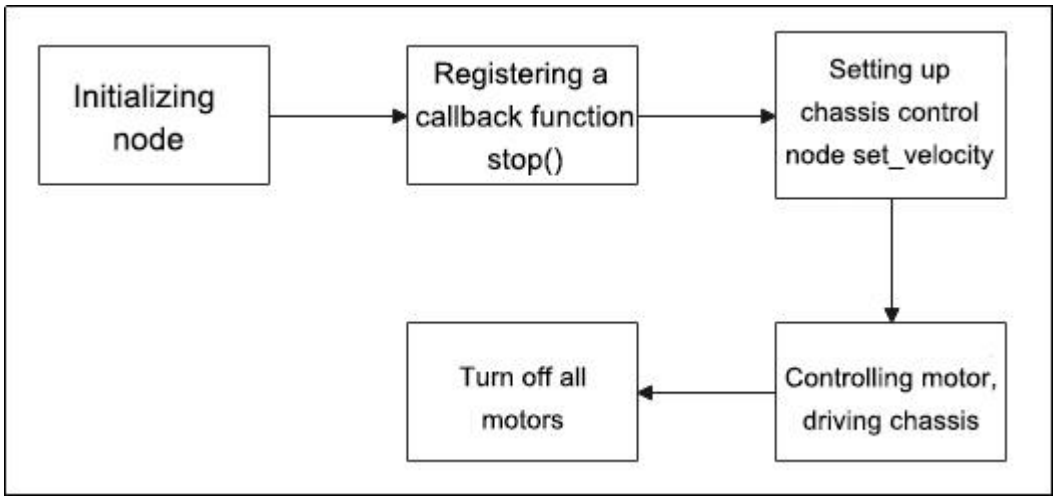
5) In "**set_velocity.publish()**" function, the third parameter represents the rotation speed of the car and we change it to -0.5. After modifying, press "Esc" and enter ":wq", and then press "Enter" to save and exit.

```
43          set_velocity.publish(60,180,-0.5) # The linear velocity is 60;
    The directional angel is 180; The yaw rate is -0.3 (When the value is
    negative, it will rotate clockwise. )
:wq
```

Note: The adjustable range of speed is from -2 to 2. When the value is negative, the car will rotate clockwise. When the value is positive, the car will rotate counterclockwise.

6) After modifying, you can refer to the content of "2.Operation Steps" to check the effect.

5. Program Analysis



The source code of program is located in: **home/ubuntu/armpi_pro/src/armpi_pro_demo/chassis_control_demo/car_drifting_demo.py**

```
start = True
#Process before closing
def stop():
    global start

    start = False
    print('closing...')
    set_velocity.publish(0,0,0)  # close all motors


if __name__ == '__main__':
    # Initialize node
    rospy.init_node('car_drifting_demo', log_level=rospy.DEBUG)
    rospy.on_shutdown(stop)
    # Mecanum chassis control
    set_velocity = rospy.Publisher('/chassis_control/set_velocity', SetVelocity, queue_size=1)

    while start:
        set_velocity.publish(60,180,-0.3) # The linear velocity is 60; The directional angel is 180;
        The yaw rate is -0.3 (When the value is negative, it will rotate clockwise. )
        rospy.sleep(3)
        set_velocity.publish(60,0,0.3)
        rospy.sleep(3)
    set_velocity.publish(0,0,0)  # Close all motors
    print('Closed')
```

Control motor through set_velocity function. There are three parameters in function. Take the code "**chassis.set_velocity(50,180,0.3)**" as an example:

1) The first parameter "50" represents the motor speed, its unit is mm/s and it ranges from -100 to 100. When the value is negative, the motor rotates counterclockwise.

2) The second parameter “180” represents the movement direction of car, its unit is degree and it ranges from 0 to 360. The value of 90° refer to move forward. 270° refers to move backward. 0° refers to move to the right. 180° refers to move the left. Other movement directions are obtained according to the same reference method.

3) The third parameter “0.3” represents the rotation speed of the car, its unit is 5°/s and it ranges from -2 to 2. When the parameter value is positive, the car will rotate clockwise. When the parameter value is negative, the car will rotate counterclockwise.