Voice Control Car Movement

Experimental Objective

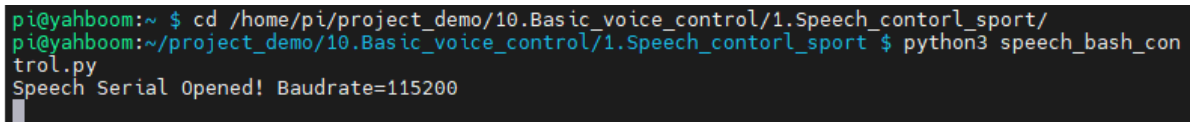
Understand and learn to control car movement using fixed basic semantics.

Experimental Procedures and Results

1. First, enter the terminal:

```
cd /home/pi/project_demo/10.Basic_voice_control/1.Speech_contorl_sport/
python3 speech_bash_control.py
```

2. After entering this interface, wake up the car using the wake-up phrase: **Hello, Xiaoya** in Chinese, or: Hi, Yahboom in English.



3. After successfully waking up, the English version responds: "Hi, I'm here." Then, use fixed statements to control the robot.

Fixed Statement Table

Wake-up Statement (International Users)	Operational Performance	Robot Response (English Version)
Robot Stop	Robot immediately ends its motion and enters the Stop state	OK, I'm stopping.
Go Ahead	Stop after moving forward for x seconds (x: program-defined, default: 5)	OK, let's go.
Back	Stop after moving backward for x seconds (x: program-defined, default: 5)	OK , I'm back.
Turn left	Stop after moving left for x seconds (x: program defined, default is 5)	OK, I'm turning left.
Turn right	Stop after moving right for x seconds (x: program defined, default is 5)	OK, I'm turning right.
Enter A mode	Stop after rotating left for x seconds (x: program defined, default is 5)	OK, I'm working on A mode.
Enter B mode	Stop after rotating right for x seconds (x: program defined, default is 5)	OK, I'm working on B mode.

Main source code analysis

```
speed = 10
stop_mytime = 5

try:
    while 1:
        time.sleep(0.2)
        num = mySpeech.speech_read()
        if num !=999 and num !=0:
            #print(num)
            mySpeech.void_write(num)
            Scommand = num

            if num <= 3 and num > 0:#parking
                stop_robot()
                #print(num)

            else:
                action_thread = threading.Thread(target=start_action)
                action_thread.daemon = True
                action_thread.start()

            num =999
except:
    mySpeech.__del__()
    print("serial close")
```

speed: Movement speed, 0-255, 0: Stop.
stop_mytime: The number of seconds after which the movement should stop (5 seconds in this case).
start_action: Executes an action based on fixed semantic recognition.

stop_robot(): Stops the robot.
speech_read: Reads the recognized message.
void_write: Selects the message sequence to be sent back by the robot's speaker. This can be found in the protocol table.