

开始语音交互

本 demo 基于科大讯飞的离线命令词识别，离线语音合成功能和 VAD 的结合来实现的语音控制。

离线命令词负责检测麦克风输入数据

离线语音合成负责合成反馈输出数据

VAD 负责获取麦克风输入数据,并保存

实现流程

实时判断周围的能量，根据能量的门限判断是否识别为语音，如果识别为语音，将其保存下来然后通过离线关键词检测，判断是否有唤醒关键词，有唤醒关键词再反馈给唤醒状态机，否则保持休眠状态。当唤醒状态机处于唤醒状态时。这时候下达的指令如果是有效的指令，则机器人执行相应的指令同时唤醒状态机返回休眠模式。

使用方式

1. 前往[科大讯飞官网](#)注册帐号，并下载离线语音合成和离线命令词识别 SDK，这里可以将两个一同打包下载。将下载的压缩包解压，将解压的文件夹里的 bin/msc/res/tts/common.jet 和 bin/msc/res/asr/common.jet 和我们提供的 demo 中的 cfg 文件中同名文件(这里的 common.jet 有两个，一个是离线语音合成的，一个是离线命令词识别的,要一一对应)并修改 appid 登录参数。因为每一个用户的 SDK 都绑定相应的 appid，如果不匹配的话,会有相应的错误信息反馈。

3. 编译代码

```
catkin_make
```

4. 运行。

```
roslaunch handsfree_voice offline_interactive_ros.launch //语音交互节点
roslaunch handsfree_voice vad_record.py //VAD语音活动侦测节点
roslaunch handsfree_hw handsfree_hw.launch //驱动控制节点
roslaunch handsfree_voice voice_cmd_vel.py //语音控制驱动节点
roslaunch handsfree_voice set_goal.py //语音导航节点
roslaunch handsfree_voice multi_point_patrol.py //语音巡逻节点
```

这里我们可以使用仿真环境先看一下效果，这里我们也提供了相应的仿真环境。

```
roslaunch handsfree_stage demo_handsfree_room_stage.launch
roslaunch handsfree_stage demo_handsfree_xuda_stage.launch
```

这两个都是仿真环境，在空间不够空旷的情况下，执行其中任意一个就能进行仿真测试。可以通过仿真环境，用我们的语音交互的功能去看执行效果，进行驱动功能的调试。

下面是一个科大讯飞的错误码信息查询，如果在调用科大讯飞的时候报错了相关的错误码，可以通过这个链接查询。

[科大讯飞错误码信息查询](#)