

基于大语言模型直控无人机仿真实验

1. 实验目的

本实验旨在探索和实现基于大语言模型（LLM）的无人机自然语言控制技术，让使用者可以通过简单的自然语言指令控制无人机的行为。通过本实验，学习者可以掌握以下几个方面的知识和技能：

1. 理解大语言模型在控制系统中的集成和应用方法；
2. 掌握如何将自然语言指令解析为具体的机器控制指令；
3. 学习无人机的基本控制原理以及Offboard模式的应用；
4. 实践PyQt图形界面编程，实现人机交互控制界面；
5. 体验人工智能技术在航空航天领域的实际应用场景。

该实验通过结合深度求索（DeepSeek）大语言模型API，构建一个能够理解和执行自然语言指令的无人机控制系统，使用户可以像与人交流一样通过自然语言与无人机进行交互，实现更加直观便捷的控制方式。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]。
- 硬件要求：笔记本/台式电脑1台^[2]。

3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\e20.LLMUsage\3.LLMDiretCtrlUAV](#)

- [llmCtrluav.py](#)：Python主程序文件，实现大语言模型与无人机控制接口的交互逻辑。
- [SITLRun.bat](#)：批处理文件，用于启动SITL仿真环境。
- [Python38Run.bat](#)：批处理文件，用于启动Python环境。

4. 实验内容或步骤

4.1 步骤1：确认LLM的API Key

本实验中需要将密钥按照格式写入"`\文档\Ogre.env`"，该文件配置和详细说明可见实验[[RflySim安装目录](#)]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\e20.LLMUsage\1.CloudAPIInvocation。打开"`\文档\Ogre.env`"，本例程中所使用的模型为DeepSeek模型如下图，依次输入DeepSeek模型相关的变量。

```
23 def __init__(self):
24     super().__init__()
25     self.llmmsg=[]
26     self.running = True
27     self.input_queue = asyncio.Queue()
28     # 初始化无人机控制实例
29     self.mav = PX4MavCtrl.PX4MavCtrl(1)
30     self.mav.InitMavLoop()
31     time.sleep(0.5)
32     self.message_signal.emit("无人机初始化完成！")
33     # self.message_signal.emit(f"无人机初始位置: {self.mav.uavPosNED}")
34     buf = ctypes.create_unicode_buffer(wintypes.MAX_PATH)
35     ctypes.windll.shell32.SHGetFolderPath(0, 5, 0, 0, buf)
36     documents_dir = Path(buf.value)
37     config_path = documents_dir / 'Ogre' / '.env'
38     if not config_path.exists():
39         raise FileNotFoundError(
40             f"未找到配置文件: {config_path}。请确保文件存在于用户目录下的该文件夹中。"
41         )
42     load_dotenv(config_path)
43     self.client = openai.OpenAI(base_url = os.getenv("DEEPSEEK_ENDPOINT"),
44                                api_key = os.getenv("DEEPSEEK_API_KEY"))
45     self.LLMModel = os.getenv("DEEPSEEK_NAME")
46     # 大模型系统提示词
47     self.Base_Prompts={"role": "system", ...
81     self.Exitlist = ["exit", "quit", "q", "退出", "离开", "退出程序", "关闭", "关闭程序", "结束", "结束程序"]
82     self.Control_Mode_List = ["Forward_Backward_Fly", "Left_Right_Fly", "Up_Down_Fly", "Left_Right_Turn"]
83     # 添加一个标志位, 用于跟踪是否已经执行过指令
84     self.command_executed = False
```

若需要使用其他模型，可按照该文件中的格式来填写，并在[[llmCtrluav.py](#)] (file:///f:/git/6.RflySimExtCtrl/0.ApiExps/e20.LLMUsage/3.LLMDiretCtrlUAV/llmCtrluav.py)文件中进行修改。

4.2 步骤2：初始化仿真环境

双击运行[SITLRun.bat]

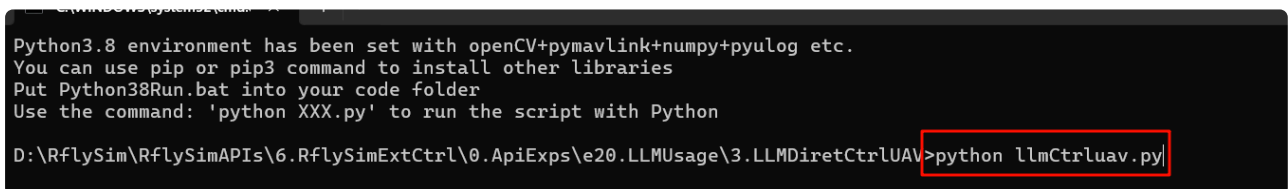
(file:///f:/git/6.RflySimExtCtrl/0.ApiExps/e20.LLMUsage/3.LLMDiretCtrlUAV/SITLRun.bat)文件，等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待CopterSim软件下侧日志栏必须打印出 GPS 3D fixed & EKf initialization finished 字样代表初始化完成。如下图所示：



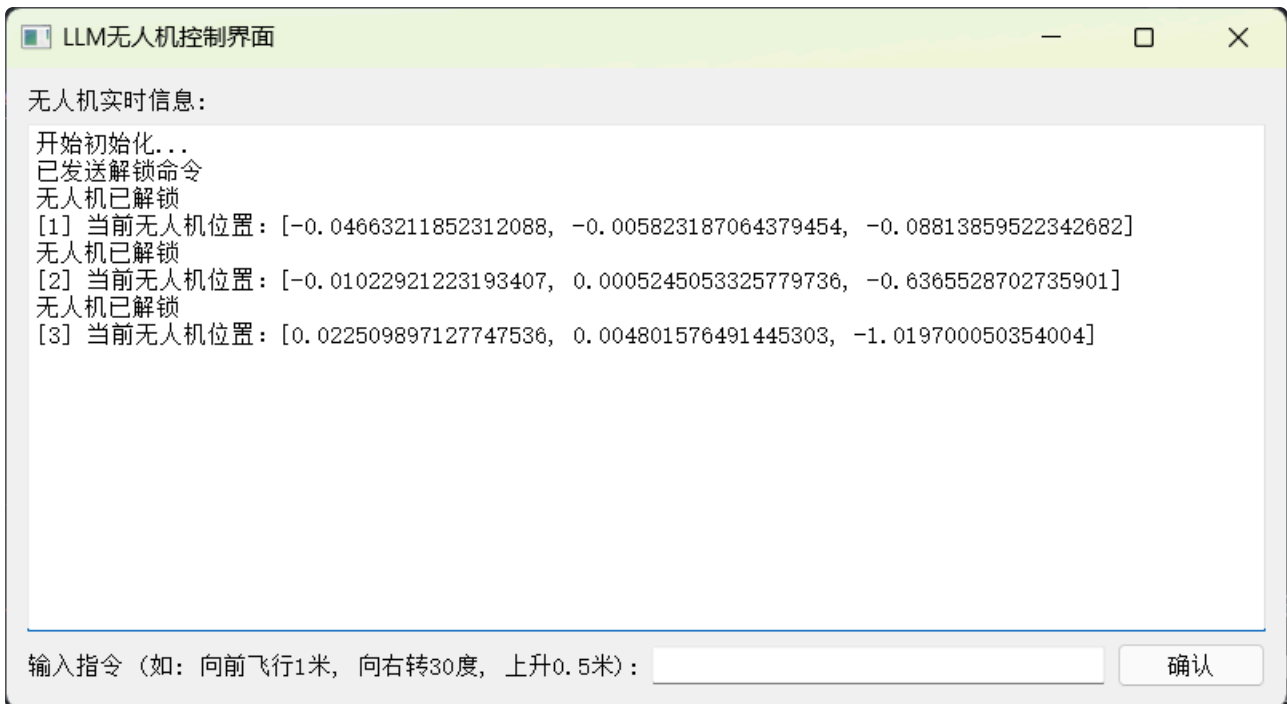
4.3 步骤3：运行仿真

双击运行 [Python38Run.bat]

(file:///f:/git/6.RflySimExtCtrl/0.ApiExps/e20.LLMUsage/3.LLMDirectCtrlUAV/Python38Run.bat) 脚本，在弹出的对话框重视输入：`python llmCtrluav.py`，



将自动弹出如下图所示的对话框，等待初始化完成后，无人机将自动垂直上升到1.7米高度后悬停，等待控制指令，同时将自动开始打印无人机的状态变量...



然后就可以开始在界面的下方输入自然语言指令，

如： 、 、 等等来控制无人机飞行。



5. 关键知识点

关键知识点1

本实验的核心是将自然语言指令转化为标准化控制命令的技术流程。整个系统主要由三个模块组成：

1. 图形用户界面模块：基于PyQt5开发，提供了用户与系统交互的可视化界面，包括无人机状态信息显示区域和自然语言指令输入区域。
2. 大语言模型解析模块：使用深度求索（DeepSeek）API作为核心处理单元，通过精心设计的系统提示词（System Prompt）引导模型将用户的自然语言指令解析为标准化的控制命令。
3. 无人机控制模块：基于PX4MavCtrl库，负责与无人机仿真器通信，将标准化控制命令转化为具体的飞控指令。

程序工作流程如下：

1. 用户在图形界面中输入自然语言指令，例如"向前飞行10米"、"向右转动30度"等；
2. 系统将用户输入发送给大语言模型，通过特定的提示词引导模型解析指令；
3. 大语言模型分析用户意图，生成标准化的控制命令，格式为[控制模式, 参数值, 无人机编号]；
4. 程序解析模型返回的结果，将其转换为具体的飞控指令发送给无人机；
5. 无人机根据接收到的指令执行相应的飞行动作。

■ 关键知识点2

大语言模型提示词工程是本实验的关键技术点之一。为了确保模型能准确地将自然语言指令转化为标准化控制命令，我们设计了一套详细的系统提示词，其中包含了严格的规则：

1. 控制模式定义：系统支持四种基本控制模式：
 - Forward_Backward_Fly（前后飞行）
 - Left_Right_Fly（左右飞行）
 - Up_Down_Fly（升降飞行）
 - Left_Right_Turn（左右转向）
2. 参数解析规则：
 - 对于直线飞行指令，如果用户没有明确指定距离，默认飞行距离为0.5米
 - 对于旋转指令，如果用户没有明确指定角度，默认旋转角度为10度
 - 设定合理的参数范围，防止无人机执行危险动作
3. 输出格式规范：
 - 模型必须严格按照["模式名称", 数值参数, 无人机编号]的格式输出
 - 如果用户输入无法识别或不符合规范，必须返回["输入语义错误，请重新输入", 0, 0]

这套提示词工程确保了系统的稳定性和准确性，即使面对表达方式多样的自然语言输入，也能正确解析用户的真实意图。

6. 实验效果

通过本实验，用户可以在图形界面上看到无人机的实时状态信息，包括当前位置坐标等，并能够通过自然语言指令控制无人机执行各种动作。实验效果包括：

1. 成功建立与无人机仿真器的连接并进入Offboard控制模式；
2. 实现无人机自动起飞至预设高度（1.7米）并保持悬停；
3. 用户可通过自然语言指令控制无人机前后、左右、上下移动；
4. 用户可通过自然语言指令控制无人机左右转向；
5. 系统能够在图形界面中实时显示无人机的位置信息和执行状态；
6. 支持连续输入指令，实现复杂的组合动作；
7. 对于不合规的输入指令，系统能够正确识别并给出反馈而不执行错误动作。

实验验证了大语言模型在无人机控制领域应用的可行性，为人机交互控制提供了新的思路和技术方案。

7. 参考资料

1. [RflySim官方文档](#)
2.
3.

8. 常见问题

Q1：详细描述问题出现的现象，必要时可以配图说明。

A1：详细描述问题出现的原因和解决方案。

Q2：详细描述问题出现的现象，必要时可以配图说明。

A2：详细描述问题出现原因和解决方案。

I Q3:

A3:

1. <https://rflysim.com/> ↩
2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩