

## inSIL28d外部数据输入接口实验

1. 文件目录
  2. 总体说明
    - 2.1. 大场景仿真综合模型的需求
    - 2.2. 接口数据解析
  3. 关键功能的实现
    - 3.1. Python发送
      - 3.1.1. 发送接口函数sendSILIntDouble
      - 3.1.2. 接口使用示例解析
      - 3.1.3. Simulink监听UDP30101端口
    - 3.2. Python发送
      - 3.2.1. 发送接口函数sendInDoubCtrls
      - 3.2.2. 接口使用示例解析
      - 3.2.3. Simulink监听UDP30101端口
  4. 相关文献
- 附加资源

## 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\0.ApiExps\11.inSILAPI\3.inSIL28d](#)

文件夹/文件名称	说明
<a href="#">1.sendSILIntDouble\Readme.pdf</a> <a href="#">2.sendInDoubCtrls\Readme.pdf</a>	Python调用inSIL28d接口实验步骤

# 总体说明

## 大场景仿真综合模型的需求

在原有动力学模型(被控对象)的基础上实现控制器从而实现更明显的控制闭环。

## 接口数据解析

结构体定义如下：

```
struct DllInSIL28d{  
  
int checksum;//校验码1234567897  
  
int CopterID; // 飞机的ID  
  
double inSIL28d[28];//28维的double型输入  
  
};
```

DllSimCtrlAPI中的sendSILIntDouble()函数和sendInDoubCtrls()函数将输入数据按照结构体定义格式打包并以UDP的方式通过30100++2系列端口发出；CopterSim运行中会循环监听该UDP端口号，监听到后将收到的数据转发给DLL模型的inSIL28d和inDoubCtrls接口。

## 关键功能的实现

### sendSILIntDouble发送

#### 发送接口函数sendSILIntDouble

- 详细解析见[4]中的python控制接口 DllSimCtrlAPI

要注意的是，sendSILIntDouble()函数分了inSILInts和inSILDoubs传入数据，inSILInts为8维数据。

```
def sendSILIntDouble(self,inSILInts=[0]*8,inSILDoubs=[0]*20,copterID=-1):  
  
checksum=1234567897
```

```
ID=copterID

if copterID<=0:

ID=self.CopterID

PortNum = 30100+(ID-1)*2

inSILInts=self.fillList(inSILInts,8)

inSILDoubs=self.fillList(inSILDoubs,20)

buf = struct.pack("2i28d",checksum,ID,*inSILInts,*inSILDoubs)

self.udp_socket.sendto(buf, (self.ip, PortNum))
```

## 接口使用示例解析

1.[sendSILIntDouble\inSIL28dTest.py](#) 关键代码如下：

### 导入必需的库和仿真API

```
import numpy as np

import time

import DllSimCtrlAPI
```

numpy：用于创建和操作数值数组，常用于科学计算。

time：用于处理时间相关的任务。

DllSimCtrlAPI：导入DllSimCtrlAPI模块，包含用于传输外部数据给dll模型的多个函数。

### 初始化DLL控制接口

```
dll = DllSimCtrlAPI.DllSimCtrlAPI()
```

创建 DllSimCtrlAPI 类的一个对象，命名为 dll。

### 执行发送到dll的任务

```
dll.sendSILIntDouble([1,2,3],np.array([4,5,6,7]))
```

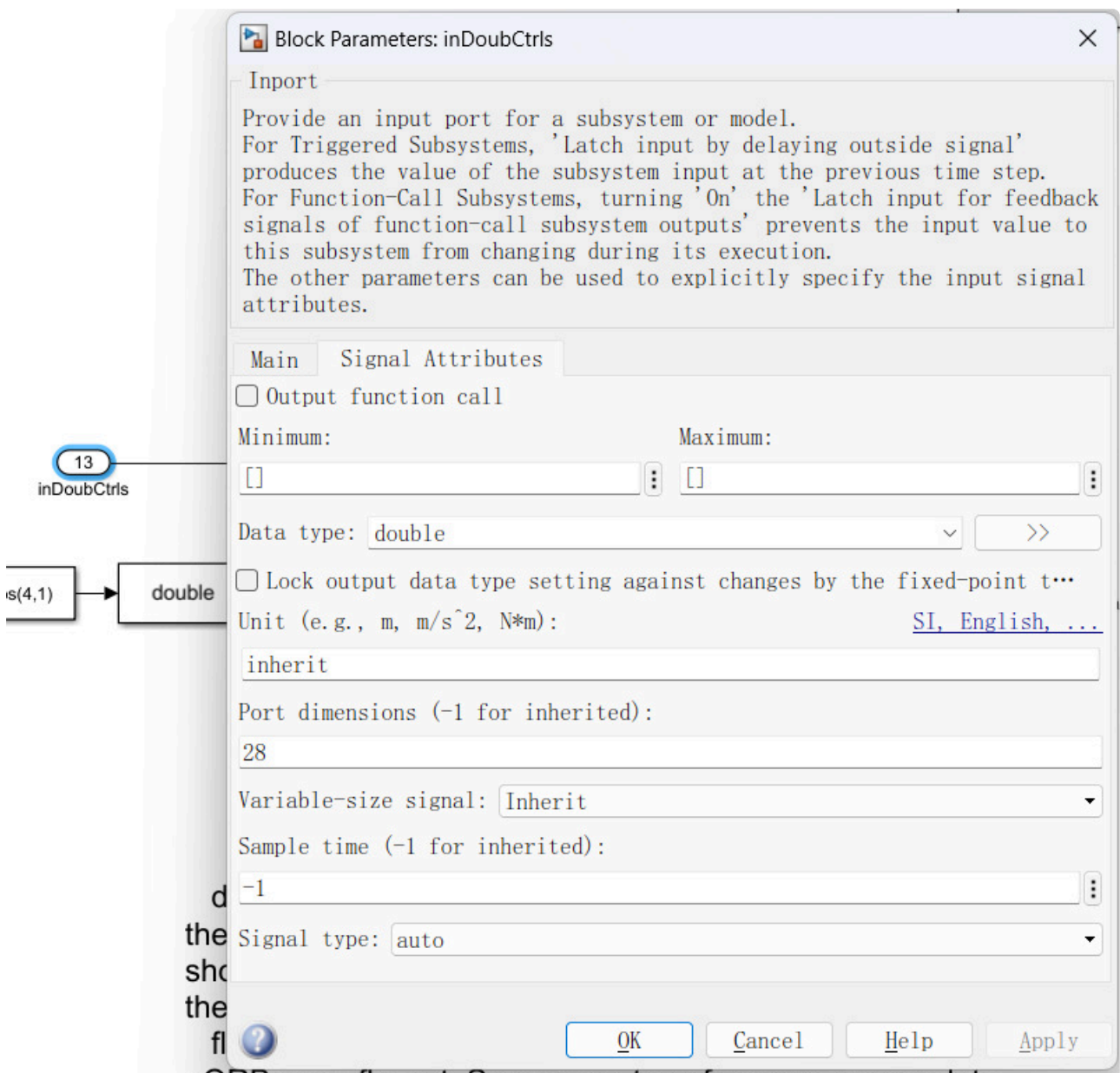
- 调用 dll 对象的 sendSILIntDouble 方法，该方法接受两个参数：一个整数列表和一个浮点数组。

- 第一个参数 [1,2,3] 是一个整数列表，表示要发送给 dll 的三个整数值。
- 第二个参数 np.array([4,5,6,7]) 是一个浮点数组，表示要发送给 dll 的四个浮点值。。

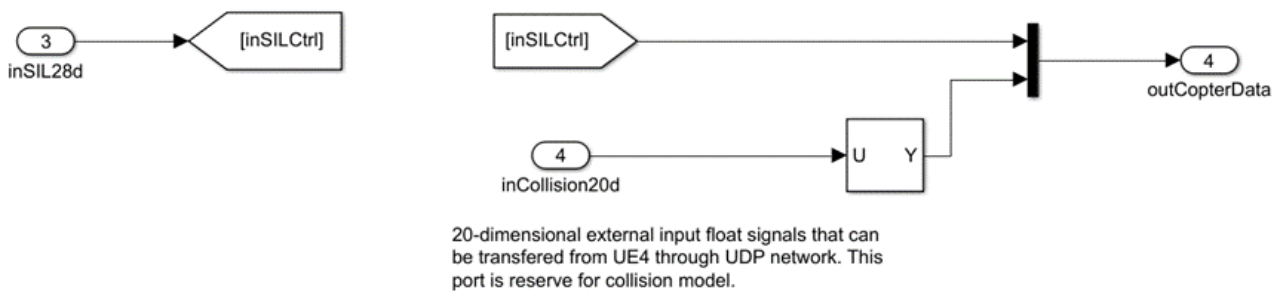
## Simulink监听UDP30101端口

- 详细Simulink代码生成配置参考[2]
- 详细outCopterData接口介绍参考[4]
- 动力学模型的外部UDP通信端口介绍参考[3]

如果想使用DLL模型inSIL28d接口，那么在Simulink模型文件中新建1个输入端口inport，将名称修改为inSIL28d，数据类型设置为double，数据维度设置为28。保存编译并重新生成DLL文件后即可使用该接口。



在Simulink模型中将输入到inSIL28d的数据转发到outCopterData接口。outCopterData接口会将数据通过UDP 30101系列端口转发出去。



本实验中DLL模型将inSIL28d接口的数据转发至outCopterData接口，因此我们在\*:\PX4PSP\CopterSim下创建csv文件后启动仿真，仿真结束关闭CopterSim后即可可以在csv中看到日志。

## sendInDoubCtrls发送

### 发送接口函数sendInDoubCtrls

- 详细解析见[4]中的python控制接口 DllSimCtrlAPI

### 接口使用示例解析

2.sendInDoubCtrls\sendInDoubCtrls.py 关键代码如下：

#### 导入必需的库和仿真API

```
import numpy as np
```

```
import DllSimCtrlAPI
```

numpy：用于创建和操作数值数组，常用于科学计算。

DllSimCtrlAPI：导入DllSimCtrlAPI模块，包含用于传输外部数据给dll模型的多个函数。

#### 初始化DLL控制接口

```
dll = DllSimCtrlAPI.DllSimCtrlAPI()
```

创建 DllSimCtrlAPI 类的一个对象，命名为 dll。

#### 执行发送到dll的任务

dll.sendInDoubCtrls([1,2,3])

方法sendInDoubCtrls接受一个包含三个数字的列表作为28位参数的前3位，并将它们作为双精度控制量发送给dll模型。

## Simulink监听UDP30101端口

- 详细Simulink代码生成配置参考[2]
- 详细outCopterData接口介绍参考[4]
- 动力学模型的外部UDP通信端口介绍参考[3]

同上3.1.3

## 相关文献

1. 第四讲API..\..\API.pdf
2. ..\2.UserDefinedC++\Readme.pdf
3. ..\3.ExtCtrlAPI\Readme.pdf
4. ..\8.OutCopterData\Readme.pdf

## 附加资源

官方文档：RflySim官方文档：<https://rflysim.com/doc/zh/>

社区交流：加入RflySim技术交流群：951534390

