# 注入故障

该功能的主要目的是向XJTU-SY数据集中的正常振动信号中注入故障信号，生成故障信号文件以供进一步分析。故障注入的方式有两种，分别模拟不同类型的振动故障。注入的信号将保存为CSV文件，并包含可视化图像，帮助用户直观了解故障的变化。

具体步骤包括：

* 从指定的信号文件中读取数据。
* 根据参数指定的时间点注入故障信号。
* 将注入故障后的信号保存为CSV文件。
* 生成注入故障前后信号的可视化图像。

## 1. 代码结构

实现注入依靠两个代码文件：

（1）data\generate\_fault\_signals.py

该文件主要实现了向正常信号中注入故障信号的功能，并保存注入后的信号。主要包含以下两个函数：

* **XJTU\_inject**：用于读取指定文件夹中的信号文件，并对信号进行故障注入。它接受多个参数，如信号文件夹路径、使用的信号文件数量、故障注入时间、故障类型和故障强度等。调用generate\_fault\_signal\_content生成注入故障后的信号。
* **params**：一个字典列表，包含不同实验设置的参数，包括信号文件夹路径、使用文件数量、注入时间、故障类型和故障强度。每个字典对应一次故障注入实验。

（2）data\inject.py

该文件包含两个主要函数，用于生成注入故障后的信号并保存到指定文件夹中：

* **inject\_fault**：该函数接受一个正常信号，并根据注入类型和故障强度参数将故障注入到信号中。支持两种注入方式：
  1. **Gause**：加入逐渐增大的高斯噪声，模拟振动故障。
  2. **bias**：加入逐渐增大的偏移量，模拟故障引起的趋势变化。
* **generate\_fault\_signal\_content**：该函数用于将注入故障后的信号按指定长度拆分成多个文件，并保存到指定路径。它还会生成可视化图像（signal.png），展示正常信号和注入故障后的信号。

data\inject.py还包含一个简单的测试函数，用于验证故障信号注入过程是否正确。它通过生成一个简单的正弦信号，注入高斯噪声，然后保存生成的信号文件和可视化图像。

**2.参数说明**

故障注入时，用户需要为函数XJTU\_inject提供以下参数：

* **signal\_cont**：正常信号所在的文件夹路径。
* **file\_nums**：使用的信号文件数量。
* **inject\_time**：注入故障信号的时间点（单位：秒）。
* **type**：注入故障的类型。可以是 'Gause'（加入高斯噪声）或 'bias'（加入偏移量）。
* **p**：故障强度的参数。对于Gause类型，表示噪声的最大标准差；对于bias类型，表示偏移量的最大值。故障从注入时刻开始，逐渐增强，直到最大值。

示例参数配置：

params = [

{

'signal\_cont': r'path\_to\_signal',

'file\_nums': 61,

'inject\_time': 30,

'type': 'Gause',

'p': 1

},

# 更多配置...

]

**故障注入方式说明**

注入故障的方式有两种：

1. **白噪声注入**： 向信号中加入方差逐渐增大的白噪声，模拟振动故障。方差代表故障的程度。通过参数p控制最大方差，故障从注入时间点开始，噪声的方差从0逐渐增大至p。
2. **趋势项注入**： 向信号中加入逐渐增大的趋势项，模拟故障引起的趋势变化。通过参数p控制趋势项的最大值，故障从注入时间点开始，趋势项从0逐渐增大至p。

**3.执行流程**

1. 在params中配置好每次注入实验的参数。
2. 运行XJTU\_inject函数，它会根据参数配置读取信号数据，注入故障并生成新的信号。
3. 注入的信号将被保存为多个CSV文件，每个文件包含一定长度的信号数据。文件将保存在指定路径下。
4. 同时，生成信号的可视化图像，并保存为signal.png。

**4. 可视化图像**

在故障信号注入后，会生成一张图像，展示正常信号与注入故障后的信号。图像中会标出故障发生的时间点，便于直观了解故障的影响。

图像将保存在保存路径下，命名为signal.png。

**5. 使用方法**

在data\generate\_fault\_signals.py中配置好参数之后，直接运行即可。