

# DAS 时间戳读取异常

## 一、概述

semi-DAS 的地震数据文件中包含了文件记录时的时间信息，可以通过不同的方式读取：

- 读取连续记录模式自动生成的文件名，获得设备的系统时间，简称为**文件名时间**；
- 使用 matlab 读取 tdms 文件，得到的时间元数据，简称为**matlab 时间**；
- 使用 segy 格式转换程序，获得写入 segy 文件中的时间，也即为**文件时间戳**。

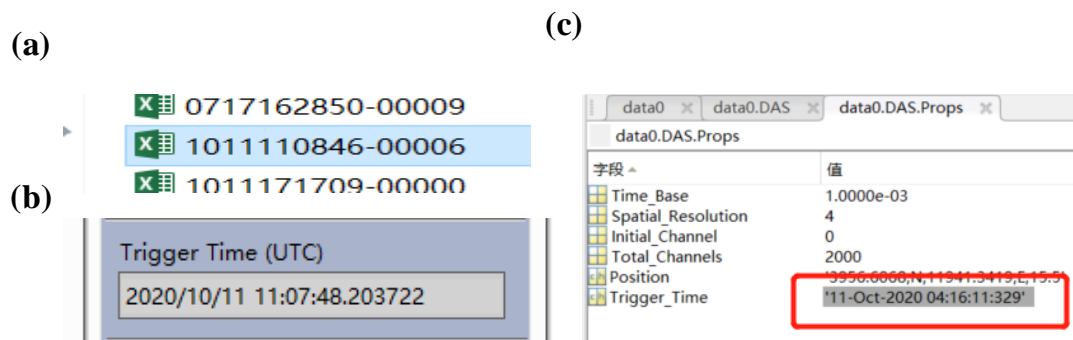


图 1 (a) 连续记录模式下 DAS 根据系统时间自动生成文件名； (b) 文件时间戳； (c) matlab 时间

在数据分析时，关于时间信息的解释应该具备统一性，即满足下列几点：

- ① **文件名时间**是设备的系统时间，接近时区时间。本文中的时区时间默认为北京时间，慢于 UTC 时间 8 小时；
- ② **文件时间戳** 是通过 GPS 天线获得的 UTC 时间；
- ③ **matlab 时间** 应该与 **文件时间戳**相同。



图 2 能够进行统一解释的地震记录时间体系

## 二、问题描述

从北京交通大学蔡云鹏那获得示例文件“1011110846-00006.tdms”，该文件的matlab时间与文件名时间相差13小时。在对文件时间戳进行检查后发现，使用matlab读取得到的matlab时间，存在13个小时误差。并且文件时间戳与文件名时间相同，即此文件中的文件时间戳实际记录的是北京时间。如下图所示：

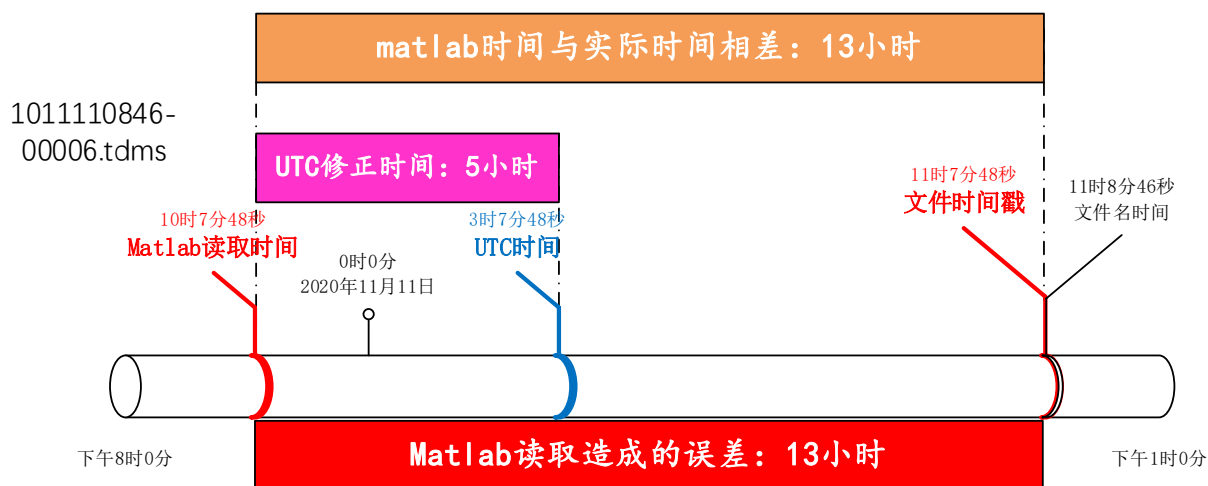


图 3 文件 1011110846-00006.tdms 的时间信息体系，其中 UTC 时间是根据 segyio 时间和文件时间推导的参考时间。

考虑到示例文件在 2020 年的试验中获得，期间 DAS 程序经过更改，遂对 2022 年获得的数据进行检查。选取 2022 年 5 月海试中，JZ07 潜次获得的数据：文件时间戳得到修正，即为 UTC 时间；但由于 matlab 读取时造成的固有误差，这使得 matlab 时间的 UTC 修正时间变为了 13 小时。

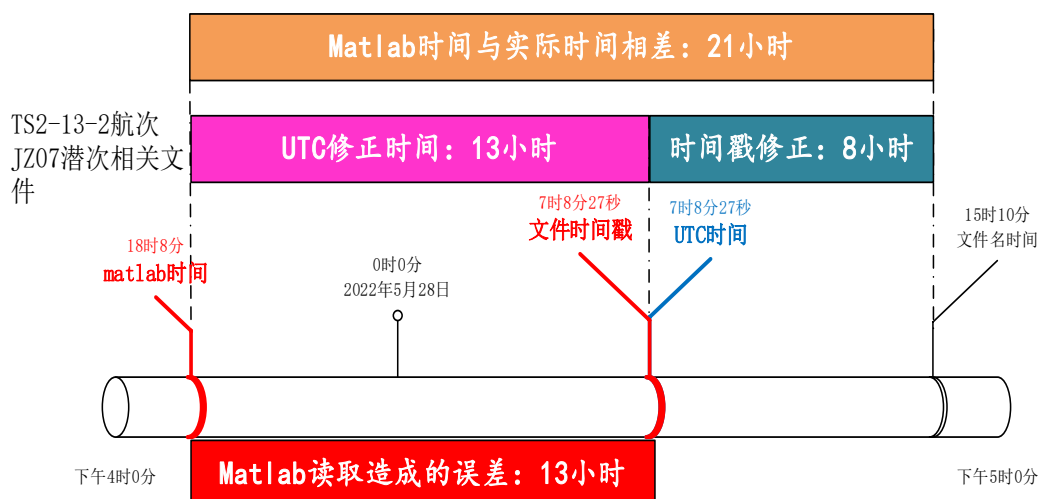


图 4 2022 年 5 月海试数据文件中的时间体系

### 三、应对方案

从图 3 和图 4 可以看出，两种时间记录状态的差别可以由“matlab 时间与文件名时间之差”进行判别。对 2020 年和 2021 年试验数据进行抽样检查，大致定位这两种时间记录状态切换的时间点，如下表所示：

表 1 2020\2021 年试验数据时间体系抽样检查表

试验场地	试验时间	文件名时间	文件时间戳	matlab 时间
三亚	2020.08	08/01 07:12:14	2020/08/01 07:11:18	2020/07/31 18:11:18
长春	2020.11	11/16 12:32:44	2020/11/16 12:31:43	2020/11/15 23:31:43
西藏	2021.04	04/09 13:59:23	2021/04/09 05:59:21	2021/04/08 16:59:21
邢台	2021.07	07/17 16:25:50	2021/07/17 08:25:49	2021/07/16 19:25:49
云南	2021.07	07/09 16:08:44	2021/07/09 08:08:43	2021/07/08 19:08:43
三亚	2021.09	09/27 17:17:15	2021/09/27 09:17:05	2021/09/26 20:17:05

从上表可以看出，直到 2020 年 11 月长春试验，matlab 时间与文件名时间之差为 13 个小时；而在 2021 年 4 月西藏试验之后，matlab 时间与文件名时间之差变为 21 小时。故而可以判断，两种时间记录状态切换的时间点在 2020 年 11 月到 2021 年 4 月之间。基于 matlab 数据读取，可以对批量的数据处理提出以下建议：

- ① 对于 2020 年 11 月之前获得的 tdms 数据，时间戳增加 5 个小时可获得相对于的 UTC 时间，增加 13 个小时可获得相对应的北京时间；

- ② 对于 2021 年 4 月之前获得的 tdms 数据,时间戳增加 13 个小时可获得相  
对于的 UTC 时间, 增加 21 个小时可获得相对应的北京时间;

## 数据修正示例代码

这里记录使用 matlab 对时间戳进行修正的步骤:

% 1. 读取一个 tdms 文件 (1011110846-00006.tdms), 并获得其时间戳

```
date= meta.date;
```

```
==> date = '10-Oct-2020 22:07:48:204'
```

% 2. 将字符串的微秒部分分离, 形成可被 datenum 读取的字符串格式

```
taleTemp= date(end-3:1:end);
```

```
>> taleTemp = ':204'
```

```
date=date(1:end-4);
```

```
>> date = '10-Oct-2020 22:07:48'
```

% 3. 将时间戳字符串转换为时间戳

```
dateNum = datenum(date)
```

% 4. 将时间戳转换为 time 类型

```
date= datetime(dateNum,'ConvertFrom','datenum')
```

```
>> date =(datetime)2020-10-10 22:07:48
```

% 5. 修正时间戳

```
dateTime.UTC=date+hours(5);% 增加 5 小时时延, 修正为 UTC 时间
```

```
>> dateTime.UTC = (datetime)2020-10-11 03:07:48
```

```
dateTime.BJ=date+hours(13); % 增加 13 小时时延, 修正为北京时间
```

```
>> dateTime.BJ = (datetime)2020-10-11 11:07:48
```

% 6. 再转换为字符串格式

```
TimeStr.UTC=datestr(dateTime.UTC)
```

```
>> TimeStr.UTC = '11-Oct-2020 03:07:48'
```

```
TimeStr.BJ=datestr(dateTime.BJ)
```

```
>> TimeStr.BJ = '11-Oct-2020 11:07:48'
```

% 7. 补充微秒部分

```
TimeStr.UTC=[TimeStr.UTC taleTemp]
```

```
>> TimeStr.UTC = '11-Oct-2020 03:07:48:204'
```

```
TimeStr_BJ=[TimeStr_BJ taleTemp]
```

```
>> TimeStr_BJ ='11-Oct-2020 11:07:48:204'
```