

## Глава 1. Восстановление метки времени в массивах данных

В процессе обработки данных полученных во время летных испытаний прибора ДЭПРОН было выяснено, что во временных метках массивов информации имеются ошибочные значения, происхождение которых связано с отсутствием календаря в ПО микроконтроллера ДЭПРОН. Так как в ПО не были заложены длительности месяцев года, при наступлении нового месяца метки времени продолжают приходить с номером предыдущего месяца к числу дней прибавляется дополнительный и возникают ошибочные даты: 2016-05-32 и 2016-05-33. На рисунке 1.1 видно наличие пробелов при наступлении нового месяца, так как невозможно автоматическое распознавание меток времени. Наличие таких отклонений должно быть исправлено отправлением метки времени в прибор от БИ в первую минуту нового месяца. Однако пока такая процедура не проводилась был накоплен значительный объем измерений и для их верной привязки к действительным датам был разработан алгоритм и реализован на языке R, листинг кода представлен в 1.1.

При привязке секундных данных к баллистическим данным была обнаружена еще одна проблема с метками времени в данных ДЭПРОН, связанная с постоянным уходом приборных часов. Для решения этой проблемы также был разработан алгоритм 1.2 и успешно применен для восстановления меток времени

### Описание алгоритма восстановления дат

На первом этапе бинарные данные каждого сброса распаковываются в текстовый вид с получением таблицы с колонками: YYYY-MM-DD hh:mm:ss-1s, count.h, count1, count2, count.both, n1, n2, dose1, dose2, filename, timestamp. Далее осуществляется разделение текстового поля с меткой времени — YYYY-MM-DD hh:mm:ss-1s на дату и время, а после полученное поле date на год, месяц и день - обозначенные year, month, day соответственно.

Создается поле dates имеющее тип данных ISOdate исходя только из полей даты года и месяца, а день месяца устанавливается первый. Далее к полю dates

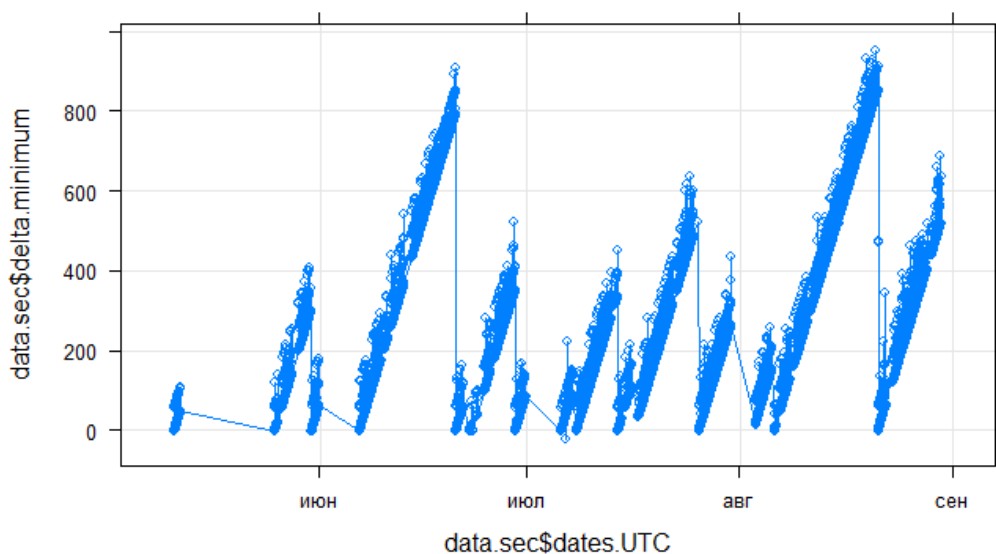


Рис. 1.1 — Временной ряд разницы приборного времени и меток начала записи в файл. Показаны первые шесть месяцев после запуска спутника Ломоносов, отключения прибора соответствуют циклограмме летных испытаний, а пробелы в данных в начале месяца соответствуют ошибочным номерам дня в месяце.

добавляется число дней из поля `day`, минус один день. В последнюю очередь в поле `dates` выставляется приборное время, отделенное в начале алгоритма.

### Описание алгоритма восстановления метки времени

Приборное время ДЭПРОН установлено на третий часовой пояс и соответствует Московскому времени, поэтому для унификации базы данных получено поле `dates.UTC` соответствующее приборному времени смещенному на 3 часа.

Далее в результате ручного анализа данных было найдено что за сутки внутренние часы ДЭПРОН уходят вперед на 57 секунд, что хорошо видно на графике 1.1 поэтому был введена поправка  $kt$ :

$$kt = (56.77315002)/86400$$

Далее с использованием полученной поправки из времени UTC получено скорректированное приборное время, хотя это время смещено относительно действи-

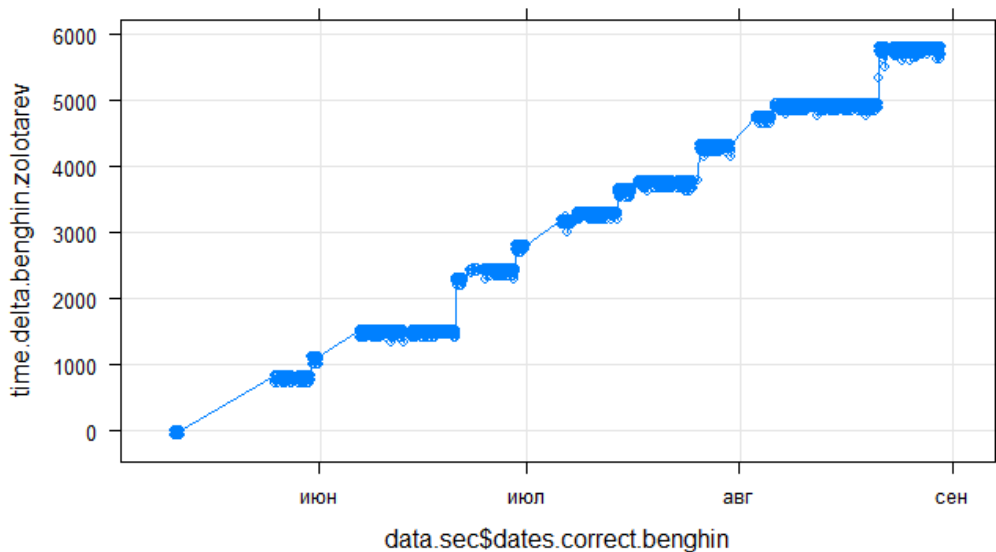


Рис. 1.2 — Временной ряд разницы калиброванного приборного времени и меток начала записи в файл.

тельного всемирного времени это смещение меняется каждый раз при выключениях прибора в ходе программы исследований [1.2](#).

Для автоматического определения этого смещения необходимо привязать его к независимому источнику точного времени - в качестве такого рассмотрены метки времени начала записи в бинарный файл данных БИ а также метки окончания записи файловой системы. Оказалось что разница времени меток последней записи сильно разнится относительно всех других меток, видимо по причине буферизации записи в файл данных, поэтому в качестве реперного времени выбрано время создания файла, которое записано в названии каждого файла как POSIXtime в виде шестнадцатеричного числа.

После получения разницы “горизонтального” приборного времени с временем начала записи в файл (поле `time.delta.file.start`) мы рассчитываем минимум (`delta.minimum`) этой разницы для каждого файла бинарных данных. Анализ распределения минимумов показал что моменты “перескоков” приборного времени из-за выключений приводят к разницам более двух минут, таким образом времена выключений были отсеяны от минутных или двухминутных пропусков в при записи в бинарных файлах данных. На основе данных о перескоках времени составлен массив `data.sec.switches`, который записывается в отдельный файл и также исходный массив секундных данных разбивается на участки без выключений прибора.

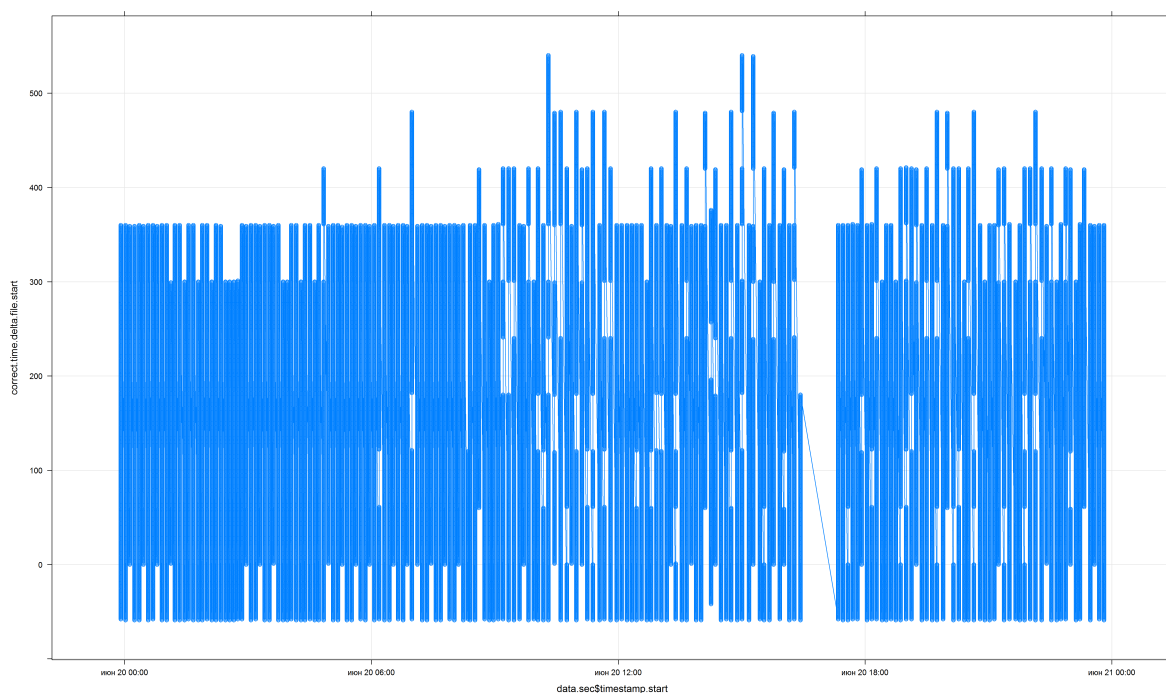


Рис. 1.3 — Пример восстановления меток времени 20 июня 2016 года

Для каждого участка непрерывной работы были найдены наиболее часто встречающиеся значения смещений `mf.v.delta` — мода разниц. И в соответствии с этими значениями скорректировано приборное время.

Последней операцией производится смещение полученного правильно времени на 59 секунд назад, так как

## 1.1 Листинги программного кода комплекса ДЭПРОН

По причине проблем с поддержкой кириллицы (она встречается в комментариях и печатаемых сообщениях), комментарии не отображены **1.1**.

## Листинг 1.1 Алгоритм коррекции даты в начале нового месяца на языке R

```

# date correction

data.sec<-separate(data.sec, 'YYYY-MM-DD hh:mm:ss-1s',c("date", "time"),
5 sep=' ')
data.sec<-separate(data.sec, 'date',c("year", "month", "day"),
sep='-', convert = TRUE)
#                               ,                               12:00

10 data.sec$dates <- ISOdate(data.sec$year, data.sec$month, 1)
#
data.sec$dates <- data.sec$dates + (as.integer(data.sec$day) - 1) * 60*60*
24
# 00:00
data.sec$dates <- data.sec$dates - 60*60*12
15 #
data.sec$dates <- data.sec$dates + parse_time(data.sec$time)

```

## Листинг 1.2 Алгоритм коррекции ухода приборных часов на R

```

\end{ListingEnv}
\begin{lstlisting}[language={Renhanced}, ]
# time correction

5 data.sec <- data.sec%>%
mutate(dates.UTC = data.sec$dates - 60*60*3 )

data.sec <- data.sec[,-17:-22]

10 #
kt = (56.77315002) /86400

#
15 data.sec <- data.sec %>%
mutate(dates.correct.benghin = dates.UTC - ceiling(
kt* (dates.UTC - min(dates.UTC))
))
20 #

```

```

data.sec$timestamp.start <-gsub("depron-", "0x", data.sec$filename)
data.sec$timestamp.start <-gsub(".dat", "", data.sec$timestamp.start)
data.sec$timestamp.start <- as.POSIXct(as.integer(data.sec$timestamp.start),
25 origin="1970-01-01", 'GMT' )

#
data.sec$timestamp.end <-
as.POSIXct(strptime(data.sec$timestamp, format="%d.%m.%Y %H:%M"))

30 #
data.sec$time.delta.file.start <- as.numeric(data.sec$dates.correct.benghin -
data.sec$timestamp.start ,
units = "secs")

35 data.sec <- data.sec %>%
group_by(filename) %>%
distinct(filename) %>%
summarise(delta.minimum = min(time.delta.file.start)) %>%
left_join(data.sec, ., by = 'filename')

40 # table(data.sec$delta.minimum)
data.sec$time.correct.zolotarev <- data.sec$dates.correct.benghin -
data.sec$delta.minimum

45 #
120 -

,
#
data.sec <-mutate(data.sec, lag.delta = delta.minimum - lag(delta.minimum))
table(data.sec$lag.delta)
50 data.sec.switches <-filter(data.sec, abs(lag.delta) >120)

data.sec <- data.sec %>%
mutate(switches = cut(data.sec$dates.UTC,
breaks = c(min(data.sec$dates.UTC),
55 data.sec.switches$dates.UTC,
max(data.sec$dates.UTC) )))

# xy1 <- xyplot( delta.minimum + switches ~ timestamp.start , data = data.sec
,
#
type = c("o", "g"))

```

```

60 # plot(xyl)

# plot(table(data.sec$delta.minimum))
# table(data.sec$delta.minimum)
# median(data.sec$delta.minimum)
65 # mfv(data.sec$delta.minimum)

library('modeest')
if(nrow(data.sec.switches)>0){
  data.sec <- data.sec %>%
70 group_by(switches) %>%
  mutate(mfv.delta =max(mfv(delta.minimum)))
}
if(nrow(data.sec.switches)== 0){
  data.sec <- data.sec %>%
75 mutate(mfv.delta =max(mfv(delta.minimum)))
}

data.sec <- data.sec %>%
mutate(dates.correct = dates.correct.benghin - mfv.delta)
80

#
data.sec$dates.correct <- as.POSIXct(data.sec$dates.correct, 'GMT') - 59

data.sec$dates.correct.copy <- as.POSIXct(data.sec$dates.correct, 'GMT')
85 #
#
data.sec$correct.time.delta.file.start <- as.numeric(data.sec$dates.correct -
data.sec$timestamp.start, units = "secs")

```