The SBS-1 Multilateration Trials Webpages

Introduction

What do we miss from our beloved SBS-1?

The aircraft list is cluttered with flights that do not appear as a radar target. These are mainly older aircraft like the MD80 series, but also modern Regional Jets (CRJ, Embraer) and the fast growing crowd of light type aircraft lack the position ADS-B squitter that is required to show them as a radar target. Not to forget military flights that tend to hide their position.

While ADS-B radar position accuracy can be as good as a few meters when GPS is used or as bad as 5NM when the aircraft's IRS is used, we cannot see those "no-pos" flights at all. Making them visible and bringing the depiction accuracy into the range of a position squittered flight is the task of this trial.

Compared to trials that have been run by others this trial uses a highly scientific calculation scheme that requires four or more receivers to "see" the same no-pos flight. More about the underlying algorithm can be found later during the trial.

What is missed to make this all happen?

There are two very basic issues that need to be resolved for us as SBS-1 users

a) - calculation of the position of the no-pos flight

b) - display of the position as a target on the radar screen, i.e. Basestation or Planeplotter etc.

We will start with question a) and come back to question b) once we have seen results that can be used for display purposes.

Calculating the position of a no-pos flight

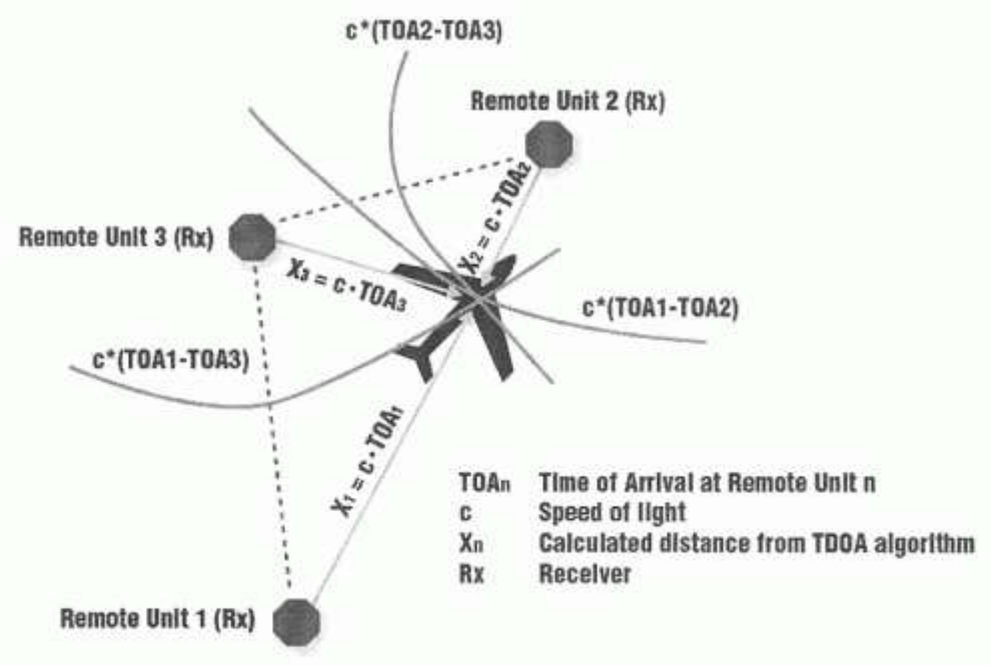
This is not an easy task. We will use the best available technique for this purpose that is called Multilateration. This technique is used in professional ATC systems, too, but under much "easier" circumstances. In general it calculates the position from a set of time differences that are derived when the associated set of receivers "sees" a data signal.

Time difference of arrival (TDOA)

When a no-pos flight emits a Mode-S data signal it (the signal) travels through the atmosphere and towards its receivers with the speed of light in air (approximately, but fair enough for our purpose):

$V_{LS} = c / r_{AIR} = 299{,}552{,}815$ m/s = $300 * 10^6$ m/s = 300 m / µs

This says that the data packet travels 300 m or about 1000 ft within a microsecond. If we can get a result like this we are pretty much within the accuracy area we would be happy about. If the receivers have a common time reference the thing would work like this: There would be a measurement when the signal would arrive at each receiver. The farther away a receiver would be from the no-pos flight the later the data packet would arrive there. One of the receivers involved would be the master receiver and all times of reception would be submitted to a central server (the multilateration controller). This server installation would calculate for each pair of master to another receiver the TDOA. For four stations we would get 3 TDOA values and this is exactly what we need for our purposes.



Indeed, professional receivers work like this and so does the SBS-2M. At least it was advertised that way. The common time reference can be derived from a GPS receiver with an uncertainty of maybe a few nano- or microseconds. But our SBS-1 receiver does not have GPS capability.

Others have tried to utilize the Windows PC clock for this purpose. Not only is this clock generally unsuited to serve as a sync source (updates occur only every few 10 or 100 ms), but also is the USB data stream from the SBS-1 box to the PC interface highly asynchronous. It means that data are buffered and then sent in bulks, when either a timeout has occurred (no more data have arrived for a certain period) or a buffer is full. Also other tasks on the PC may delay processing and time stamping.

Therefore, we need a different solution. The timestamp must be attached to the data packet somewhere inside the SBS-1. At the end of our introduction we have learned that the calculation of a position of a no-pos flight is very much a question of the proper the synchronization of the clocks of the SBS-1 receivers.

The SBS-1 data stream counter

This is a delicate matter and at the center of our first trials. To learn what we try to achieve we must know a few bits about the SBS-1 data stream. We know that most of it is encrypted and only a few were able to convert it to clear data. We don't anticipate this poses a problem here. Each packet of flight data comes embedded into a very professional set of additional data, amongst these there is a 24 bit counter that is very interesting for our purposes.

10 02	Header (STX)
01	SBS-1 messagetype
00	fill byte, always 00
1B 29 56	SBS-1 24-bit Counter
8F 71 BE 01 60 BF 00 B3 1C 5C A1 00 00 00	Raw Mode-S message (DF=17)
3E 49	SBS-1 CRC checksum
10 03	Trailer (ETX)

Table: Typical SBS-1 data packet as decrypted. Other messagetypes than 01 (ADS-B squitter) are: 05 (short Mode-S no-pos message), 07 (long Mode-S no-pos message).

Some experiments have shown that the counter is most likely driven by a 20 MHz clock, i.e. its basic interval is 0.05 µs or 50 ns. This corresponds with half a cycle of a 40 Mhz crystal oscillator chip that can be located on the SBS-1 motherboard. We can assume that this is the main clock for the digital data processing unit. Not more than that is known about this counter but there are no indications that it is used by Basestation at all except to add additional data to the packet stream. As the counter changes quickly the encryption will then let each data packet look very differently, although the raw message may not have changed.

We do not know at what stage the clock counter value is added to the data stream, but as all our receivers have the same hardware and we are only looking for time differences we can safely assume that this is not relevant. The delay from the data reception on the front end of our SBS-1 to the addition of the value is the same for all SBS-1 and subsequently the time difference is zero and can be ignored.

The resolution of the counter applied to the speed of light Vl provides a resolution of

tl = 0.05 µs * 299.552815 m / µs = 15 m

I.e. with every step of the counter the Mode-S data signal travels 15 m, which is a quite comfortable resolution.

But there is more tricky stuff around our counter that we have to deal with. Namely

- it is a 24 bit counter only and as such it can take values from 0 to 2^{24}-1 = 16,777,215 only before it overflows and restarts with 0. The associated length of a full 24 bit counter period is only about 0.83886 s. This means that we cannot process time differences that are greater than 839 ms (not a big deal for the calculation itself as this equals more than 250,000 km), but we also have to provide certain measures that can unambiguously identify to which counter cycle a data packet belongs.

- all our SBS-1 receivers have a different counter value at a specific time, we call it *offset*

- all clocks of our SBS-1 receivers run at *different speeds* at a specific time

- as with all crystal clocks the clock speeds of our SBS-1 receivers change over time, temperature, age and whatsoever. We call it *drift*. The only thing we can safely assume is that a change in speed will not occur abrupt but smoothly.

Clock synchronization with ADS-B flights

Before we can chase no-pos flights we need to know what characteristics the clock of each of our SBS-1 receiver chain has. We will declare one of our receivers the master receiver. We have learned that three values are important, the offset, the speed difference and the drift between the master and the other receivers.

To calculate these values we use a method that is probably used for the first time here and that is surprisingly simple. It makes use of those ADS-B squitters that not only carry exact position data of the transmitting flight but also the same counter value as described above for no-pos flights. Because we know the exact position of the SBS-1 receivers, too, we can calculate the distance between the target and the receiver and, after application of the rule of the speed of light, we also know what the travel time of the data packet is.

This will be our first stage of trial.

Data acquisition

Acquisition tool

Data are acquired through an offline tool that generates a standardized log file over 2 minutes. Earlier trials have shown that an online tool is more comfortable, but the high data rate from some stations together with an increased requirement for configuration changes on the user side (routers, firewalls) deemed it more appropriate to use a tool that can operate completely on its own. The tool can be programmed in advance so to start data acquisition at a specific time, therefore an unmanned operation is possible.

Logfile formats

Log files are ASCII formatted with the PC timestamp added. This is for reference only and will not be used for the position calculation. Here are some examples:

```
20:00:00.412 - 07 - 00 2C 68 F4 - 5D A9 D1 E4 00 00 00 - 3FFD
20:00:00.413 - 01 - 00 6A 8A F4 - 8D A9 D1 E4 99 01 D2 0A 28 08 00 00 00 00 - 0424
20:00:00.415 - 05 - 00 C8 00 F8 - A0 00 16 91 FF F4 75 42 FF FC EE A2 BE BB - 7FDE
20:00:00.417 - 05 - 00 39 25 F8 - A0 00 13 1C 80 1D AD 31 60 0C E3 4C A0 BB - E9C6
20:00:00.418 - 01 - 00 33 96 F8 - 8D 40 06 8D 99 04 A5 21 80 5F 00 00 00 00 - 1047
```

Data are preformatted so that the clock, message type, counter value (LSB first), raw message and the checksum (in that order) appear separated by a hyphen. Each of these data blocks can be processed by other applications as is required.

Overview of Mode-S data types

During the various stages of data processing we will be required to identify and decode the raw data messages. The SBS-1 message type (01, 05, or 07) is not sufficient to properly identify the contents of a message. So we have a closer look onto the raw message itself and see how we can come to a quick check what the contents would be:

Raw data	Raw data	Download	SBS-1	Basestation TCP	Name of message and squitter rate	Useful data	Aircraft ID
Byte 1	Byte 5	format	Message type	socket message type			24-bit
20..27		04	07	MSG,5	Mode-S altitude reply	Altitude	end
28..2F		05	07	MSG,6	Mode-S id reply	Squawk	end
58..5F		11	07	MSG,8	All call reply/ACAS Squitter (every 1 s	none	Byte 2-4

					OPEN)		
88..8F		17	01		Extended Squitter		Byte 2-4
	00..07	17	01			Altitude	
	08..27	17	01	ID or MSG,1	Aircraft Identification Squitter (BDS 0,8) (every 4.8 to 5.2 s)	Callsign	
	28..47	17	01	MSG,2	Surface Position Squitter (BDS 0,6)	Position, TTrack, GndSpd	
	48..97	17	01	MSG,3	Airborne Position Squitter (BDS 0,5) (every 0.4 to 0.6 s)	Position, Altitude	
	48..4F					*Precision < 3m*	
	50..57					*Precision < 10m*	
	58..5F					*Precision < 0,05 NM*	
	60..67					*Precision < 0,1 NM*	
	68..6F					*Precision < 0,25 NM*	
	70..77					*Precision < 0,5 NM*	
	78..7F					*Precision < 1 NM*	
	80..87					*Precision < 5 NM*	
	88..8F					*Precision < 10 NM*	
	90..97					*Precision > 10 NM*	
	99 (for subsonic)	17	01	MSG,4	Airborne Velocity Squitter (BDS 0,9) (every 0.4 to 0.6 s)	TTrack, GndSpd, VertRate, GNSS AltDiff	
A0..A7		20	05	MSG,5	Comm-B altitude reply	Altitude	end
A8..AF		21	05	MSG,6	Comm-B id reply	Squawk	end

As it appears a look to the first and fifth byte of the raw message lets us quickly sort out, which data are interesting and which should be ignored. With the examples from above we can decode:

20:00:00.412 - 07 - 00 2C 68 F4 - 5D A9 D1 E4 00 00 00 - 3FFD

An all call reply or squitter (byte 1 = 5D) from aircraft **A9D1E4** (bytes 2 to 4) without any further usable data. This is a message that just says, I am here. The message was received at SBS-1 counter value F4682C which translates to 16,017,452.

20:00:00.413 - 01 - 00 6A 8A F4 - 8D A9 D1 E4 99 01 D2 0A 28 08 00 00 00 00 - 0424

This is an ADS-B squitter message (byte 1 = 8D) from the same aircraft that provides velocity data (byte 5 = 99). The velocity is encoded in the sequence 01 D2 0A 28 08. The counter value is F48A6A, which is 16026218. This is 8766 counts later than the previous message which in turn is 8,766 * 0.05 µs = 438 µs.

20:00:00.417 - 05 - 00 39 25 F8 - A0 00 13 1C 80 1D AD 31 60 0C E3 4C A0 BB - E9C6

Here we have a no-pos message (byte 1 = A0) from aircraft **4CA0BB** (the last three bytes). Message type 05 packets are only sent on specific request from a ground interrogator. This message was processed by SBS-1 at F82539 (16,262,457), which is 245,005 counts or 12.25 ms after the first message above.

What data can we use for what?

We will have to expect two stages of calculation that require known and unknown data.

The clock synchronization requires the following packets to decode a precisely known inertial position of a reference flight:

- ADS-B squitter high precision position and barometric altitude (Flight level) i.e. byte 1 = 88..8F and byte 5 = 48..77

- ADS-B squitter velocity data to correct the barometric altitude by the GNSS altitude difference, if available, i.e. byte 1 = 88..8F and byte 5 = 99

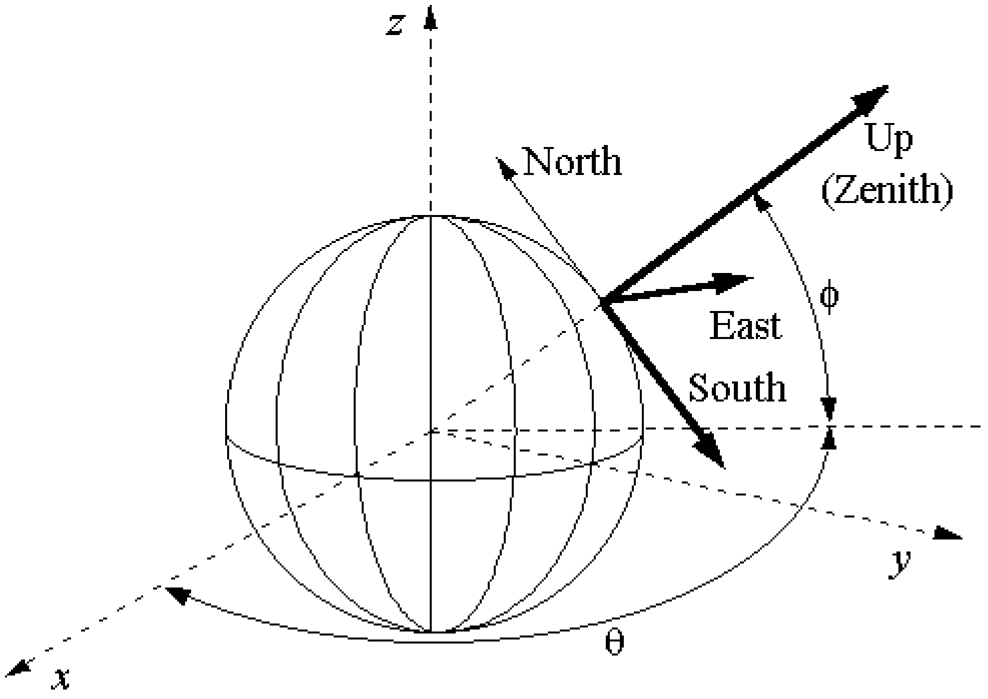
For flights with unknown position the data required can be derived from the other messages if it is secured that the message does not contain garbage. Therefore only those no-pos messages should be processed where the aircraft id had been confirmed by the reception of an all call reply/squitter message 58..5F. These messages, as are messages 88..8F, are CRC secured on the Mode-S downlink and the SBS-1 only lets them through if they have proven to be correct, i.e. the last three bytes are zero.

Inertial position conversion

While we have the luxury that position data are transmitted by ADS-B squitters, these positions are a bit different from what we need. As we will convert signal travel time to distance and v.v. this will assume that a position is described in space by latitude, longitude and altitude over the globe and distances are calculated as straight and slant.

Altitude is therefore very important. As an example, while a flight exactly overhead the SBS-1 would be described as 0 NM distance from it, the travel time of a transponder signal would be a product of the light of speed and the altitude over ground.

To calculate distances we will there for convert the position and altitude data we have to an inertial coordinate system. This Cartesian system has the center of the earth as a midpoint.



The conversion is

[Latitude, Longitude, Height (=Altitude (FL) + GNSS Altitude Difference + radiusEarth)] >> [x, y, z]

The distance between two points x1, y1, z1 and x2, y2, z2 would then be

d12 = sqrt [(x2 - x1)2 + (y2 - y1)2 +(z2 - z1)2]

With a mean radiusEarth = 6371 km the conversion is

x = sin (Longitude) * cos (Latitude) * Height

y = sin (Latitude) * Height

z = cos (Longitude) * cos (Latitude) * Height

all in km.

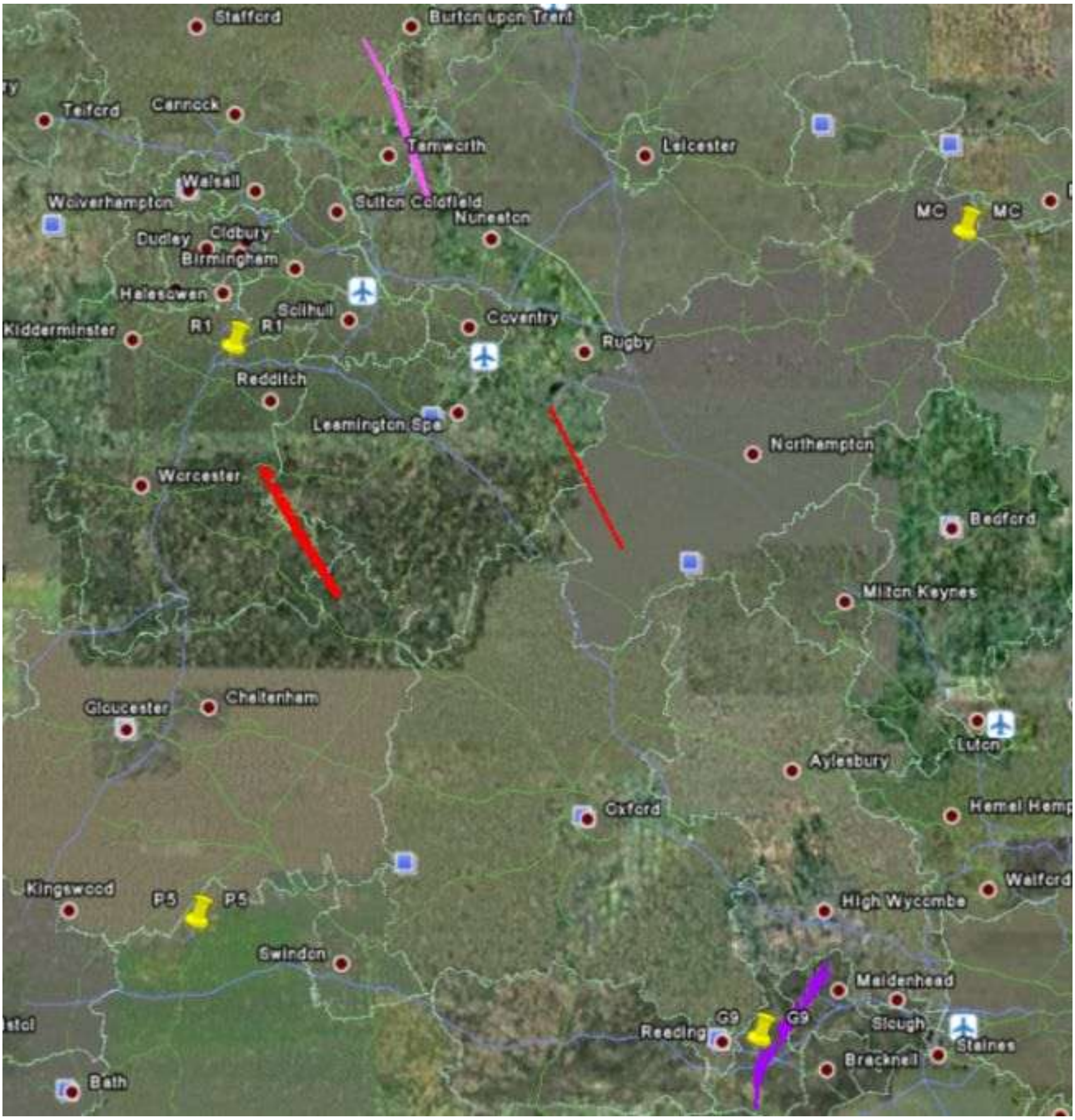
The mean value of radiusEarth = 6371 km shall be subject to discussion later, because this may not be the appropriate value for the position of the flight target.

A first glance onto trial data

The data acquisition trial was successfully completed on Monday Oct 27, 2008 between 20.00 and 20.02 UTC. We have acquired six sets of data files with around 18 MByte of data packets.

SBS-1 receiver locations

Thanks to your great support a suite of good locations could be used for our trial purposes. Ideally the four SBS-1 locations are grouped like a circle or rectangle around the area that is interesting. From the SBS-1 locations that have recorded data the below selection of stations was made. Also in the picture you can see a few tracks that were recorded during the trial, each of them about 2 minutes long.



The stations depicted R1, MC, P5 and G9 had the highest number of hits and in combination they are ideally located to track flights in the inner of the rectangle they span.

Useable flights for clock synchronization

A few tools are available here that can scan the log files for our purposes. The first step is to identify those flights that have been seen by all four SBS-1 stations at the same time. We do this with the following logic. For a moving target an ADS-B position squitter has a unique byte pattern. This is derived from the position encoding. The following general assumptions apply according to ICAO Annex 10:

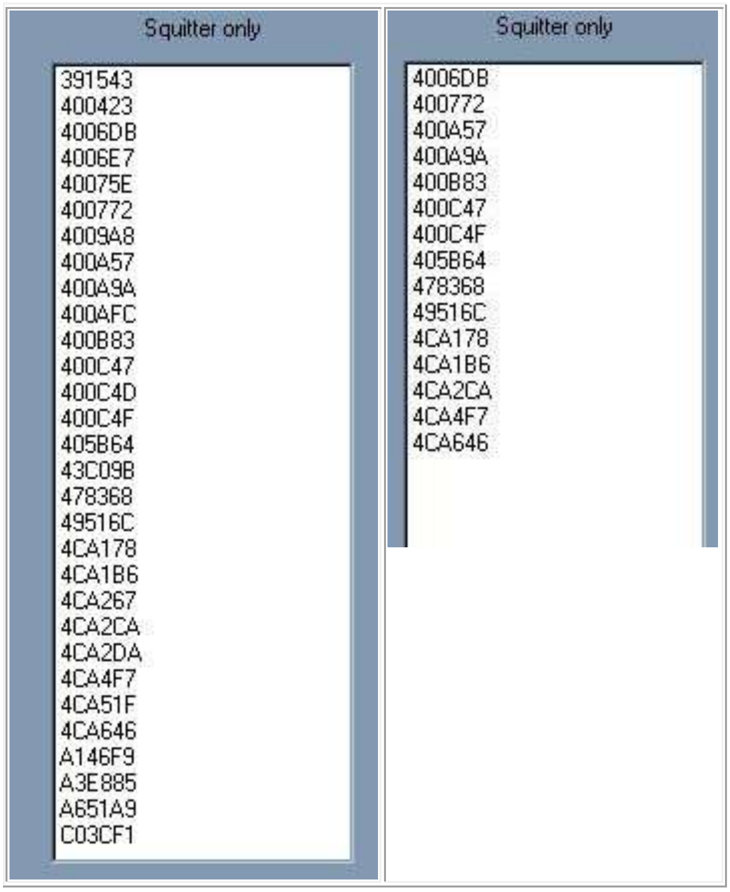
- The position squitter resolution (high precision data) is in the range of a few 10 meters

- The position squitter downlink rate is around 2 per second (ICAO Annex 10 IV 3.1.2.8.6.4.2: "Airborne position squitter transmissions shall be emitted when the aircraft is airborne at random intervals that are uniformly distributed over the range from 0.4 to 0.6 seconds using a time quantization of no greater than 15 milliseconds relative to the previous airborne position squitter...")

- A jet airplane flies around 200 meters per second at cruise, i.e. 100 meters per position squitter

This means that for high precision squitters we can well assume that a different position data packet is transmitted with every squitter.

Now we have to search for those aircraft for which identical positions squitters have been seen by all four stations. Here is the result of this analysis:



```
Squitter only
391543
400423
4006DB
4006E7
40075E
400772
4009A8
400A57
400A9A
400AFC
400B83
400C47
400C4D
400C4F
405B64
43C09B
478368
49516C
4CA178
4CA1B6
4CA267
4CA2CA
4CA2DA
4CA4F7
4CA51F
4CA646
A146F9
A3E885
A651A9
C03CF1
```

```
Squitter only
4006DB
400772
400A57
400A9A
400B83
400C47
400C4F
405B64
478368
49516C
4CA178
4CA1B6
4CA2CA
4CA4F7
4CA646
```

The left window shows those aircraft that position squitters were received from by all four stations during the trial.The right hand list show those aircraft from who identical position squitter packets were received by all four stations. In the end data from 15 airframes are useable for synchronization purposes

For demonstration of the trial results we will continue to use data from airframe 4006DB. This is a First Choice A321 G-OOAE. At this time we do not have more information about this particular flight (and we do not need any). In the above picture of the trial area it is the red track that approaches Coventry from the South.

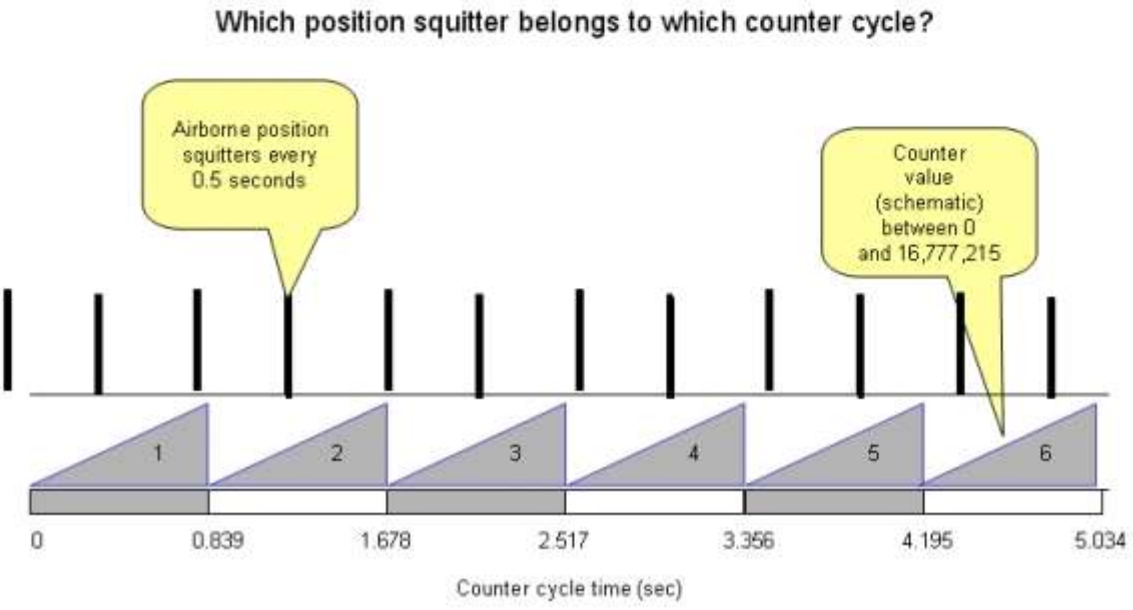
Here is a view onto some data packets after they were sorted to time order:

```
8D 40 06 DB 58 71 B6 43 D7 BE EB 00 00 00 0 B868BA 20:00:24.716
8D 40 06 DB 58 71 B6 43 D7 BE EB 00 00 00 1 D4CA7C 20:00:24.375
8D 40 06 DB 58 71 B6 43 D7 BE EB 00 00 00 2 C89D8B 20:00:26.802
8D 40 06 DB 58 71 B6 43 D7 BE EB 00 00 00 3 3AE78C 20:00:27.110
-----------------
8D 40 06 DB 58 71 F6 43 4F BF 0D 00 00 00 0 5AD76F 20:00:26.736
8D 40 06 DB 58 71 F6 43 4F BF 0D 00 00 00 1 7739A9 20:00:26.625
8D 40 06 DB 58 71 F6 43 4F BF 0D 00 00 00 2 6B0ECD 20:00:29.045
8D 40 06 DB 58 71 F6 43 4F BF 0D 00 00 00 3 DD56D1 20:00:29.589
-----------------
8D 40 06 DB 58 73 06 43 4F BF 0D 00 00 00 0 D7F67A 20:00:26.962
8D 40 06 DB 58 73 06 43 4F BF 0D 00 00 00 1 F458CB 20:00:27.062
8D 40 06 DB 58 73 06 43 4F BF 0D 00 00 00 2 E82E50 20:00:29.496
8D 40 06 DB 58 73 06 43 4F BF 0D 00 00 00 3 5A75F6 20:00:30.262
```

You can see the raw position squitter, then the station number 0,1,2, or 3, the associated 24-bit counter value that came with the data packet (expressed as a hex number) and the PC time stamp from the log file.

We can see that PC time stamps are only a rough estimation of time as much as they differ. Also it is obvious from the data received that there are some difficulties in receiving every position squitter by all four stations. This should happen every ca. 1 second, but in reality this is the exception and there are some gaps.

Given that the 24-bit counter overflows after 0.8 seconds we have to take additional measures to assign the data packets to the correct counter period. Here is a sketch about the problem:


Which position squitter belongs to which counter cycle?
Airborne position squitters every 0.5 seconds
Counter value (schematic) between 0 and 16,777,215
1
2
3
4
5
6
0
0.839
1.678
2.517
3.356
4.195
5.034
Counter cycle time (sec)

We will continue to work on this issue in the next chapter.

Receiver clock synchronization

With the data acquired the following logic was applied to derive the offset and drift of each receiver:

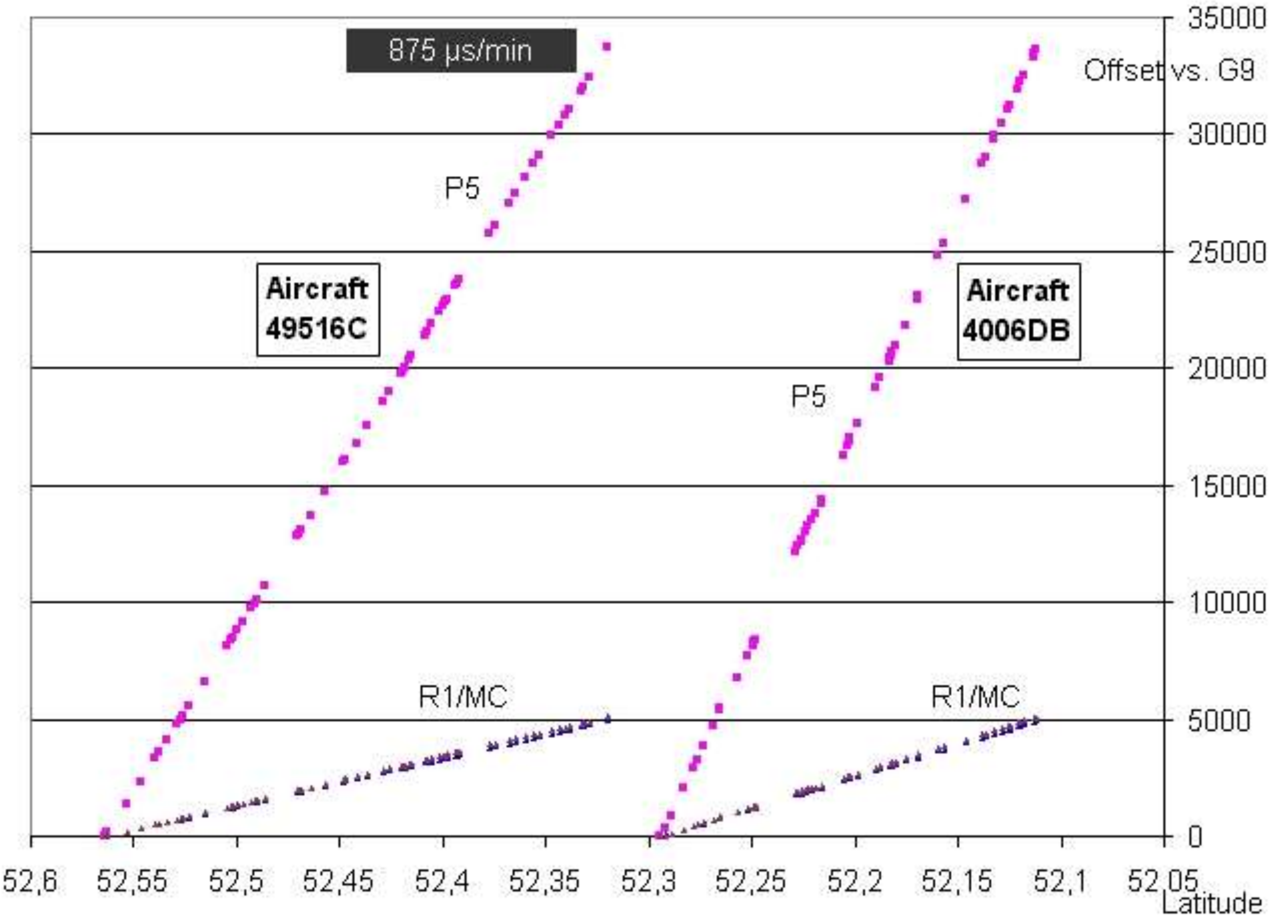
- Decode the 24-bit counter value which represents the time at which the data packet was processed by SBS-1

- Decode the reference flight position and convert it to inertial coordinates
- Calculate the distance between each receiver and the flight, which represents the travel time of the data packet
- Convert the calculated distance into clock counts (50 ns)
- Deduct each counter value by these clock counts
- The result is the counter value which represents the time at which the data packet has left the aircraft's transponder

After we choose G9 to be our reference receiver we now get values for the three offsets

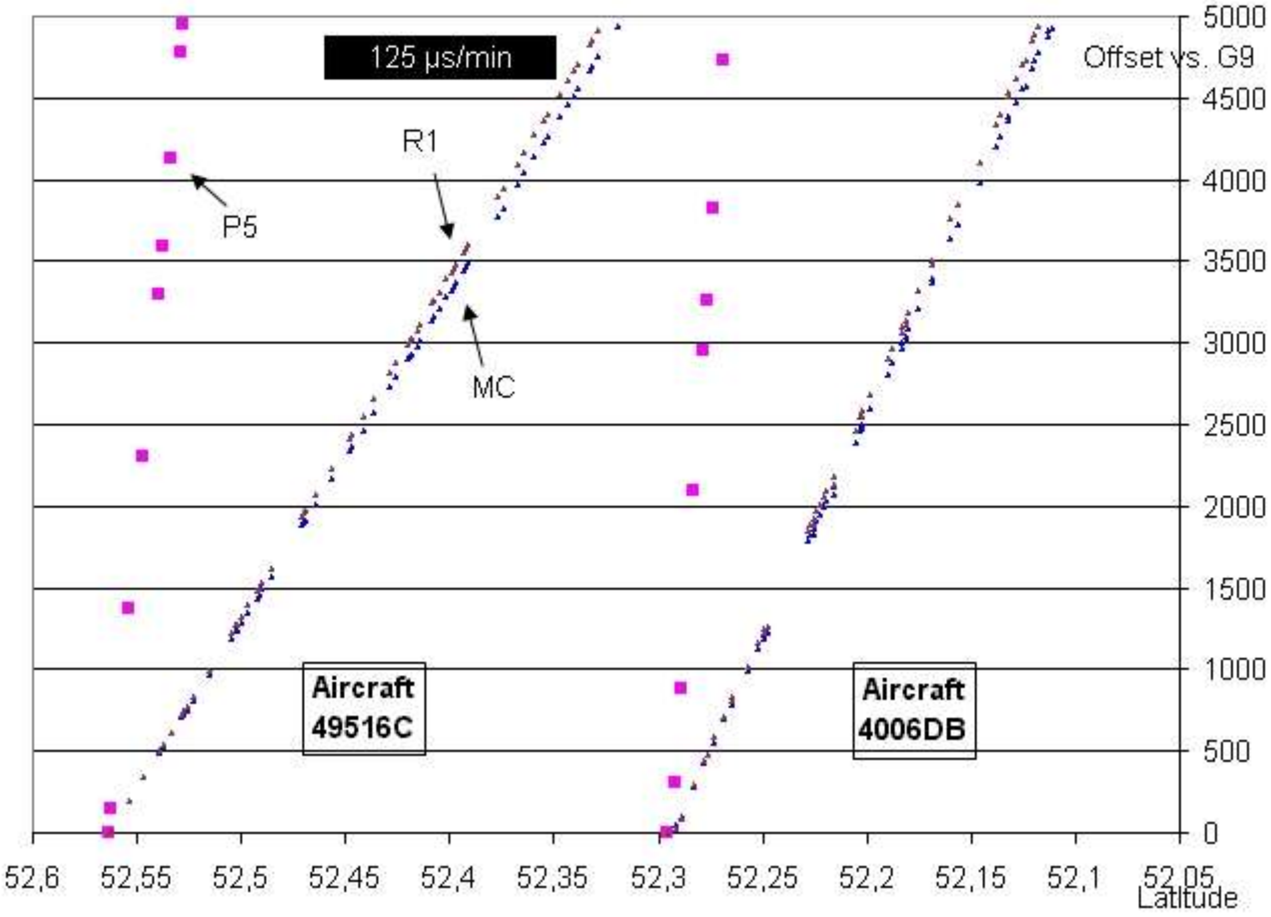
- MC minus G9
- R1 minus G9
- P5 minus G9

over the flight time or distance flown. We have selected two aircraft 4006DB and 49516C that were both on a pretty Southbound track. The result is shown in the two graphs below.



We have drawn the offset for each plotted position over the latitude. The scope is the entire 2 minutes of the trial as both aircraft were seen during the full period.

Compared to station G9 station P5 has a pretty steep drift which equals about 17,000 clocks or 875 µs/min, while R1 and MC have a much smaller drift and almost the same drift value. To have a better view onto their drift the resolution is enhanced in the second graph:



The resulting drift for both R1 and MC is about 2,500 clocks or 125 µs per minute.

Remark: a high drift value does not describe a malfunction. Drift values are just relative to a randomly selected other box and they depend on the crystal accuracy and the prevailing environmental conditions, especially the temperature.

In addition to the drift we have observed the following basic offset values at the beginning of the recording (as hexadecimal values):

- CLKG90 = 12A316
- CLKMC0 = 2F0A90
- CLKP50 = 22C61A
- CLKR10 = 952AEF

For any box and any flight we can now adjust the received clock value to a clock value as if the receiver would behave as box G9:

- CLKG9' = CLKG9.- CLKG90

- CLKMC' = CLKMC.- (CLKMC0-CLKG90) + t * DRIFTMC
- CLKP5' = CLKP5.- (CLKP50-CLKG90) + t * DRIFTP5

CLKR1' = CLKR1.- (CLKR10-CLKG90) + t * DRIFTR1