

The automatic mode algorithm

A set of well-formed audio characteristics (R) is first defined (implementations may be restrictive for practical purposes as above), variables in (R) will henceforth be called regular variables.

The algorithm is based on the two equations on regular variables,

$$\begin{aligned}(1) \quad N - F C B/8 &= 0 \\(2) \quad S - C B/8 &= 0\end{aligned}$$

where N is the number of bytes per second
S the number of bytes per sample (all channels)
C the number of channels
B the number of bits per sample channel
F sampling frequency in Hz

Assumptions on header state are:

- (3) three out of the five above variables are assumed to be correct, and considered as parameters.

Mathematical discussion

Let $D=\{N, S, C, B, F\}$. The above system of equations (1) and (2) form a linear system with two unknown variables if the pair of variables is among this list: $\{N, S\}$, $\{N, C\}$, $\{N, B\}$, $\{S, F\}$, $\{S, C\}$, $\{S, B\}$, as the determinant is not null. In these cases, there is a single solution to the linear system.

However, the determinant is either null, or the system is not linear, for the following pairs of unknown variables: $\{N, F\}$, $\{F, C\}$, $\{F, B\}$, $\{C, B\}$, out of the 10 possible pairs. In these cases yet, S is always known and, following (3), considered as a parameter. As there must be a solution, the problem thus boils down to proving unicity under the set of assumptions. From (2) it can be shown that, for a pair of solutions $\{(N, S, C, B, F), (N', S, C', B', F')\}$:

$$(4) \quad B/B' = k, \text{ where } k = C'/C$$

The $\{F, C\}$ and $\{F, B\}$ cases are straightforward and the solution is unique. However for $\{C, B\}$ the set of three constants $\{F, N, S\}$ is linked by the equation $F = N S$, hence (2) is hyperbolic. For this case we now add the following assumptions on variables, which define a stricter set (R'):

(R')

1. number of channels is strictly positive and not a multiple of 3,
2. bit rate is either 16, 20 or 24 and if 20, channels are different from 4 or 5.

Variables satisfying (R') in this case will be called regular variables equally.

Now, for B, B' in $\{16, 20, 24\}$, and C, C' and $C' > C$, chosen without loss of generality, in $\{1, 2, 4, 5\}$, $B/B' > 1$ hence $\{B', B\} = \{16, 20\}$ or $\{16, 24\}$ or $\{20, 24\}$, entailing $C'/C = 3/2$ or $5/4$ or $6/5$ with C in $\{1, 2, 4, 5\}$ and C' in $\{2, 4, 5\}$, $C' > C$.

This set of constraints cannot be satisfied under (R). Ab absurdo, it ensues that $C' = C$, then $B' = B$ out of (4).

Out of the five cases at hand, N is a known correct parameter except in the four cases, hence $N' = N$, whence $F' = F$ out of (1).

There remains the $\{N, F\}$ case, which should be very rare, and added to the set of header assumptions as below:

(3') *(header assumptions, revised)*

Three parameters are known to be correct, other than $\{S, C, B\}$.

In the $\{S, C, B\}$ case, the algorithm will bail out.

Algorithm

The algorithm first tests whether all five C variables read from the file header are within the bounds of (R), the set of regular values for this mode. If there are fewer than two such variables out of five, fixwav reverts to manual mode. Then setting the channel number, two regular variables are selected other than $\{S, B\}$. If this is not possible, fixwav bails out. The other two variables are calculated out of (1) and (2), then tested to be within the bounds of (R). Should the test fail, fixwav looks for other possible combinations of known parameters. The above theorem ensures that there is just one solution: the first regular values are the only ones under the set of assumptions. In the $\{C, B\}$ case, the linear constraint on constants is checked and the stricter conditions (R') are enforced, bailing out if they are not satisfied. Then the one remaining equation is solved by looping on the number of channels C : the above theorem ensures that the first regular pair is the only one solution. When all options have failed, fixwav bails out to manual mode, otherwise it returns `BAD_header`.

Important note

1. The algorithm assumes that if the constants are regular, then they are correct values. Should this assumption be erroneous, wrong corrections can be made that satisfy all mathematical constraints. User checking is therefore advised when option `-a` is used (please refrain from using silent mode `-q` in conjunction with `-a`). Example of “wrong” correction: $C = 1$, $S = 3$, $B = 24$, $F = 96$ kHz, instead of $C = 2$, $S = 6$, $B = 24$, $F = 48$ kHz.

2. The above theorem holds in the 20-bit case except for one minor subcase ($C = 4$ or $C = 5$). The implementation ignores 20-bit configurations for simplicity.

Added by Fabrice Nicol, May 2008, corrected Aug. 2016