ETSI TS 146 082 V16.0.0 (2020-09)

TECHNICAL SPECIFICATION

Digital cellular telecommunications system (Phase 2+) (GSM); Voice Activity Detector (VAD) for Enhanced Full Rate (EFR) speech traffic channels (3GPP TS 46.082 version 16.0.0 Release 16)

GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS

Reference
RTS/TSGS-0446082vg00

Keywords
GSM

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The present document specifies the Voice Activity Detector (VAD) to be used in the Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels within the digital cellular telecommunications system.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   or greater indicates TSG approved document under change control.

y   the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z   the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document specifies the Voice Activity Detector (VAD) to be used in the Discontinuous Transmission (DTX) as described in GSM 06.81 [5] Discontinuous transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels.

The requirements are mandatory on any VAD to be used either in GSM Mobile Stations (MS)s or Base Station Systems (BSS)s that utilize the enhanced full-rate speech traffic channel.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] GSM 01.04: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".

[2] GSM 06.53: "Digital cellular telecommunications system (Phase 2+); ANSI-C code for the GSM Enhanced Full Rate (EFR) speech codec".

[3] GSM 06.54: "Digital cellular telecommunications system (Phase 2+); Test vectors for the GSM Enhanced Full Rate (EFR) speech codec".

[4] GSM 06.60: "Digital cellular telecommunications system (Phase 2+); Enhanced Full Rate (EFR) speech transcoding".

[5] GSM 06.81: "Digital cellular telecommunications system (Phase 2+); Discontinuous transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

noise: signal component resulting from acoustic environmental noise.

mobile environment: any environment in which mobile stations may be used.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

3.2.1 Variables

aav1   filter predictor values, see clause 5.2.3
acf   the ACF vector which is calculated in the speech encoder (GSM 06.60 [4])

adaptcount	secondary hangover counter, see clause 5.2.6
av0	averaged ACF vector, see clause 5.2.2
av1	a previous value of av0, see clause 5.2.2
burstcount	speech burst length counter, see clause 5.2.8
den	denominator of left hand side of equation 8 in annex B, see clause 5.2.5
difference	difference between consecutive values of dm, see clause 5.2.4
dm	spectral distortion measure, see clause 5.2.4
hangcount	primary hangover counter, see clause 5.2.8
lagcount	number of subframes in current frame meeting periodicity criterion, see clause 5.2.9
lastdm	previous value of dm, see clause 5.2.4
lags	the open loop long term predictor lags for the two halves of the speech encoder frame (GSM 06.60 [4])
num	numerator of left hand side of equation 8 in annex B, see clause 5.2.5
oldlagcount	previous value of lagcount, see clause 5.2.9
prederr	fourth order short term prediction error, see clause 5.2.5
ptch	Boolean flag indicating the presence of a periodic signal component, see clause 5.2.9
pvad	energy in the current filtered signal frame, see clause 5.2.1
rav1	autocorrelation vector obtained from av1, see clause 5.2.3
rc	the first four unquantized reflection coefficients calculated in the speech encoder (GSM 06.60 [4])
rvad	autocorrelation vector of the adaptive filter predictor values, see clause 5.2.6
smallag	difference between consecutive lag values, see clause 5.2.9
stat	Boolean flag indicating that the frequency spectrum of the input signal is stationary, see clause 5.2.4
thvad	adaptive primary VAD threshold, see clause 5.2.6
tone	Boolean flag indicating the presence of an information tone, see clause 5.2.5
vadflag	Boolean VAD decision with hangover included, see clause 5.2.8
veryoldlagcount	previous value of oldlagcount, see clause 5.2.9
vvad	Boolean VAD decision before hangover, see clause 5.2.7

3.2.2 Constants

adp	number of frames of hangover for secondary VAD, see clause 5.2.6
burstconst	minimum length of speech burst to which hangover is added, see clause 5.2.8
dec	determines rate of decrease in adaptive threshold, see clause 5.2.6
fac	determines steady state adaptive threshold, see clause 5.2.6
frames	number of frames over which av0 and av1 are calculated, see clause 5.2.2
freqth	threshold for pole frequency decision, see clause 5.2.5
hangconst	number of frames of hangover for primary VAD, see clause 5.2.8
inc	determines rate of increase in adaptive threshold, see clause 5.2.6
lthresh	lag difference threshold for periodicity decision, see clause 5.2.9
margin	determines upper limit for adaptive threshold, see clause 5.2.6
nthresh	frame count threshold for periodicity decision, see clause 5.2.9
plev	lower limit for adaptive threshold, see clause 5.2.6
predth	threshold for short term prediction error, see clause 5.2.5
pth	energy threshold, see clause 5.2.6
thresh	decision threshold for evaluation of stat flag, see clause 5.2.4

3.2.3 Functions

+	addition
-	subtraction
*	multiplication
/	division
$\|x\|$	absolute value of x
AND	Boolean AND
OR	Boolean OR
$\text{MULT}(x(i))$ with $i=a$ to b	the product of the series x(i) for i=a to b
$\text{SUM}(x(i))$ with $i=a$ to b	the sum of the series x(i) for i=a to b

i=a

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACF	Autocorrelation function
ANSI	American National Standards Institute
DTX	Discontinuous Transmission
LTP	Long Term Predictor
TX	Transmission
VAD	Voice Activity Detector

For abbreviations not given in this clause, see GSM 01.04 [1].

4 General

The function of the VAD is to indicate whether each 20 ms frame produced by the speech encoder contains speech or not. The output is a Boolean flag (vadflag) which is used by the Transmit (TX) DTX handler defined in GSM 06.81 [5].

The present document is organized as follows.

Clause 5 describes the principles of operation of the VAD. Clause 6 provides an overview of the computational description of the VAD. The computational details necessary for the fixed point implementation of the VAD algorithm are given in the form of ANSI C program contained in GSM 06.53 [2].

The verification of the VAD is based on the use of digital test sequences which are described in GSM 06.54 [3].

5 Functional description

The purpose of this clause is to give the reader an understanding of the principles of operation of the VAD, whereas GSM 06.53 [2] contains the fixed point computational description of the VAD. In the case of discrepancy between the two descriptions, the description in GSM 06.53 [2] will prevail.

5.1 Overview and principles of operation

The function of the VAD is to distinguish between noise with speech present and noise without speech present. This is achieved by comparing the energy of a filtered version of the input signal with a threshold. The presence of speech is indicated whenever the threshold is exceeded.

The detection of speech in a mobile environment is difficult due to the low speech/noise ratios which are encountered, particularly in moving vehicles. To increase the probability of detecting speech the input signal is adaptively filtered (see clause 5.2.1) to reduce its noise content before the voice activity decision is made (see clause 5.2.7).

The frequency spectrum and level of the noise may vary within a given environment as well as between different environments. It is therefore necessary to adapt the input filter coefficients and energy threshold at regular intervals as described in clause 5.2.6.

5.2 Algorithm description

The block diagram of the VAD algorithm is shown in figure 1. The individual blocks are described in the following clauses. The variables shown in the block diagram are described in table 1.

Table 1: Description of variables in figure 1

Var	Description
acf	The ACF vector which is calculated in the speech encoder (GSM 06.60 [4]).
av0	Averaged ACF vector.
av1	A previous value of av0.
lags	The open loop long term predictor lags for the two halves of the speech encoder frame (GSM 06.60 [4]).
ptch	Boolean flag indicating the presence of a periodic signal component.
pvad	Energy in the current filtered signal frame.
rav1	Autocorrelation vector obtained from av1.
rc	The first four reflection coefficients calculated in the speech encoder (GSM 06.60 [4]).
rvad	Autocorrelation vector of the adaptive filter predictor values.
stat	Boolean flag indicating that the frequency spectrum of the input signal is stationary.
thvad	Adaptive primary VAD threshold.
tone	Boolean flag indicating the presence of an information tone.
vadflag	Boolean VAD decision with hangover included.
vvad	Boolean VAD decision before hangover.

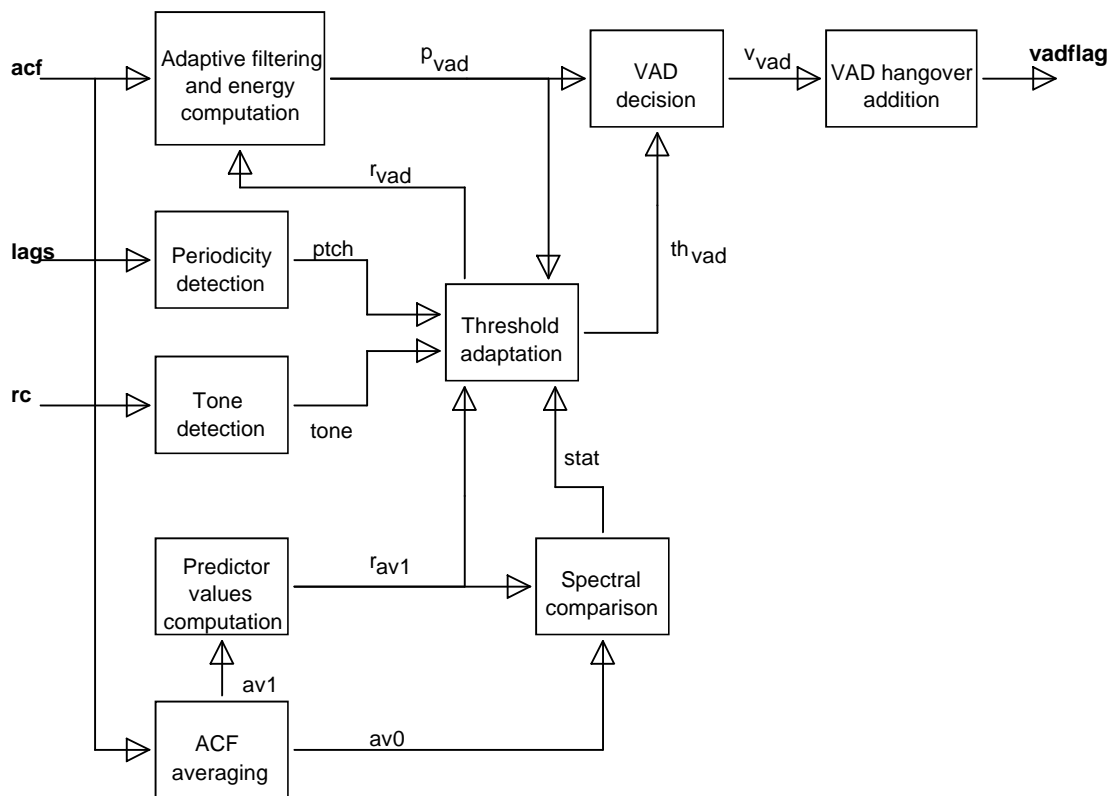**Figure 1: Functional block diagram of the VAD**

5.2.1   Adaptive filtering and energy computation

The energy in the current filtered signal frame (pvad) is computed as follows:

$$pvad = rvad[0] * acf[0] + 2 * \sum_{i=1}^{8} (rvad[i] * acf[i]) \qquad (1)$$

This corresponds to performing an 8th order block filtering on the filtered input samples to the speech encoder. This is explained in annex A.

5.2.2 ACF averaging

Spectral characteristics of the input signal have to be obtained using blocks that are larger than one 20 ms frame. This is done by averaging the ACF (autocorrelation function) values for several consecutive frames. The averaging is given by the following equations:

```
          frames-1
  av0{n}[i] = SUM  (acf{n-j}[i])        ; i = 0..8                                    (2)
          j=0

  av1{n}[i] = av0{n-frames}[i]          ; i = 0..8                                    (3)
```

where (n) represents the current frame, (n-1) represents the previous frame. The values of constants are given in table 2.

Table 2: Constants and variables for ACF averaging

Constant	Value	Variable	Initial value
frames	4	previous ACF's, av0 & av1	All set to 0

5.2.3 Predictor values computation

The filter predictor values aav1 are obtained from the autocorrelation values av1 according to the equation:

$$a = R^{-1} p \qquad\qquad (4)$$

where:

$$
R = \begin{bmatrix}
av1[0] & av1[1] & av1[2] & av1[3] & av1[4] & av1[5] & av1[6] & av1[7] \\
av1[1] & av1[0] & av1[1] & av1[2] & av1[3] & av1[4] & av1[5] & av1[6] \\
av1[2] & av1[1] & av1[0] & av1[1] & av1[2] & av1[3] & av1[4] & av1[5] \\
av1[3] & av1[2] & av1[1] & av1[0] & av1[1] & av1[2] & av1[3] & av1[4] \\
av1[4] & av1[3] & av1[2] & av1[1] & av1[0] & av1[1] & av1[2] & av1[3] \\
av1[5] & av1[4] & av1[3] & av1[2] & av1[1] & av1[0] & av1[1] & av1[2] \\
av1[6] & av1[5] & av1[4] & av1[3] & av1[2] & av1[1] & av1[0] & av1[1] \\
av1[7] & av1[6] & av1[5] & av1[4] & av1[3] & av1[2] & av1[1] & av1[0]
\end{bmatrix}
$$

and:

$$
p = \begin{bmatrix}
av1[1] \\
av1[2] \\
av1[3] \\
av1[4] \\
av1[5] \\
av1[6] \\
av1[7] \\
av1[8]
\end{bmatrix}
\qquad
a = \begin{bmatrix}
aav1[1] \\
aav1[2] \\
aav1[3] \\
aav1[4] \\
aav1[5] \\
aav1[6] \\
aav1[7] \\
aav1[8]
\end{bmatrix}
$$

```
aav1[0] = -1
```

av1 is used in preference to av0 as the latter may contain speech. The autocorrelated predictor values rav1 are then obtained:

```
           8-i
 rav1[i] =  SUM (aav1[k] * aav1[k+i])            ; i = 0..8                    (5)
           k=0
```

5.2.4   Spectral comparison

The spectra represented by the autocorrelated predictor values rav1 and the averaged autocorrelation values av0 are compared using the distortion measure (dm) defined below. This measure is used to produce a Boolean value stat every 20 ms, as shown in the following equations:

```
                          8
 dm  = (rav1[0] * av0[0] + 2*SUM (rav1[i]*av0[i])) / av0[0]                    (6a)
                         i=1

 difference = |dm - lastdm|                                                    (6b)

 lastdm = dm                                                                   (6c)

 stat = (difference < thresh)                                                  (6d)
```

The values of constants and initial values are given in table 3.

Table 3: Constants and variables for spectral comparison

Constant	Value	Variable	Initial value
thresh	0.056	lastdm	0

5.2.5   Information tone detection

Information tones and noise can be classified by inspecting the short term prediction gain, information tones resulting in a higher prediction gain than noise. Tones can therefore be detected by comparing the prediction gain to a fixed threshold. By limiting the prediction gain calculation to a fourth order analysis, information signals consisting of one or two tones can be detected whilst minimizing the prediction gain for noise.

The prediction gain decision is implemented by comparing the normalized short term prediction error with the short term prediction error threshold (predth). This measure is used to produce a Boolean value, tone, every 20 ms. The signal is classified as a tone if the prediction error is less than predth. This is equivalent to a prediction gain threshold of 13.5 dB.

Vehicle noise can contain strong resonances at low frequencies, resulting in a high prediction gain. A further test is therefore made to determine the pole frequency of a second order analysis of the signal frame. The signal is classified as noise if the frequency of the pole is less than 385 Hz.

The algorithm for evaluating the Boolean tone flag is as follows:

```
    tone = false
    den = a[1]*a[1]
    num = 4*a[2] - a[1]*a[1]
    if (num <= 0)
      return

    if ((a[1] < 0) AND (num/den < freqth))
      return
            4
```

```
    prederr = MULT (1 - rc[i] * rc[i])
            i=1

    if (prederr < predth)
      tone = true

    return
```

rc[1..4] are the first four unquantized reflection coefficients obtained from the speech encoder short term predictor. The coefficients a[0..2] are transversal filter coefficients calculated from rc[1..2] using the step up routine. The pole frequency calculation is described in annex B.

The values of the constants are given in table 4.

Table 4: Constants for information tone detection

Constant	Value
freqth	0,0973
predth	0,0447

5.2.6   Threshold adaptation

A check is made every 20 ms to determine whether the VAD decision threshold, (thvad) should be changed. This adaptation is carried out according to the flowchart shown in figure 2. The values of the constants and initial variable values are given in table 5.

Adaptation of thvad takes place in two different situations:

In the first case, the decision threshold (thvad) is set to the lower limit for the adaptive threshold (plev) if the input signal frame energy (acf[0]) is less than the energy threshold (pth). The autocorrelation vector of the adaptive filter predictor values (rvad) remains unchanged.

In the second case, thvad and rvad are adapted if there is a low probability that speech or information tones are present. This occurs when the following conditions are met:

   a) The frequency spectrum of the input signal is stationary (clause 5.2.4).

   b) The signal does not contain a periodic component (clause 5.2.9).

   c) Information tones are not present (clause 5.2.5).

The autocorrelation vector of the adaptive filter predictor values (rvad) is updated with the rav1 values. The step size by which thvad is adapted is not constant but a proportion of the current value and its rate of increase or decrease is determined by constants inc and dec respectively.

The adaptation begins by experimentally multiplying thvad by a factor of (1-1/dec). If thvad is now higher than or equal to pvad times the steady state adaptive threshold constant (fac), then thvad needed to be decreased and it is left at this new lower level. If, on the other hand, thvad is less than pvad times fac then it either needs to be increased or kept constant. In this case, it is multiplied by a factor of (1+1/inc) or set to pvad times fac whichever yields the lower value. Thvad is never allowed to be greater than pvad+upper adaptive threshold limit (margin).

Table 5: Constants and variables threshold adaptation

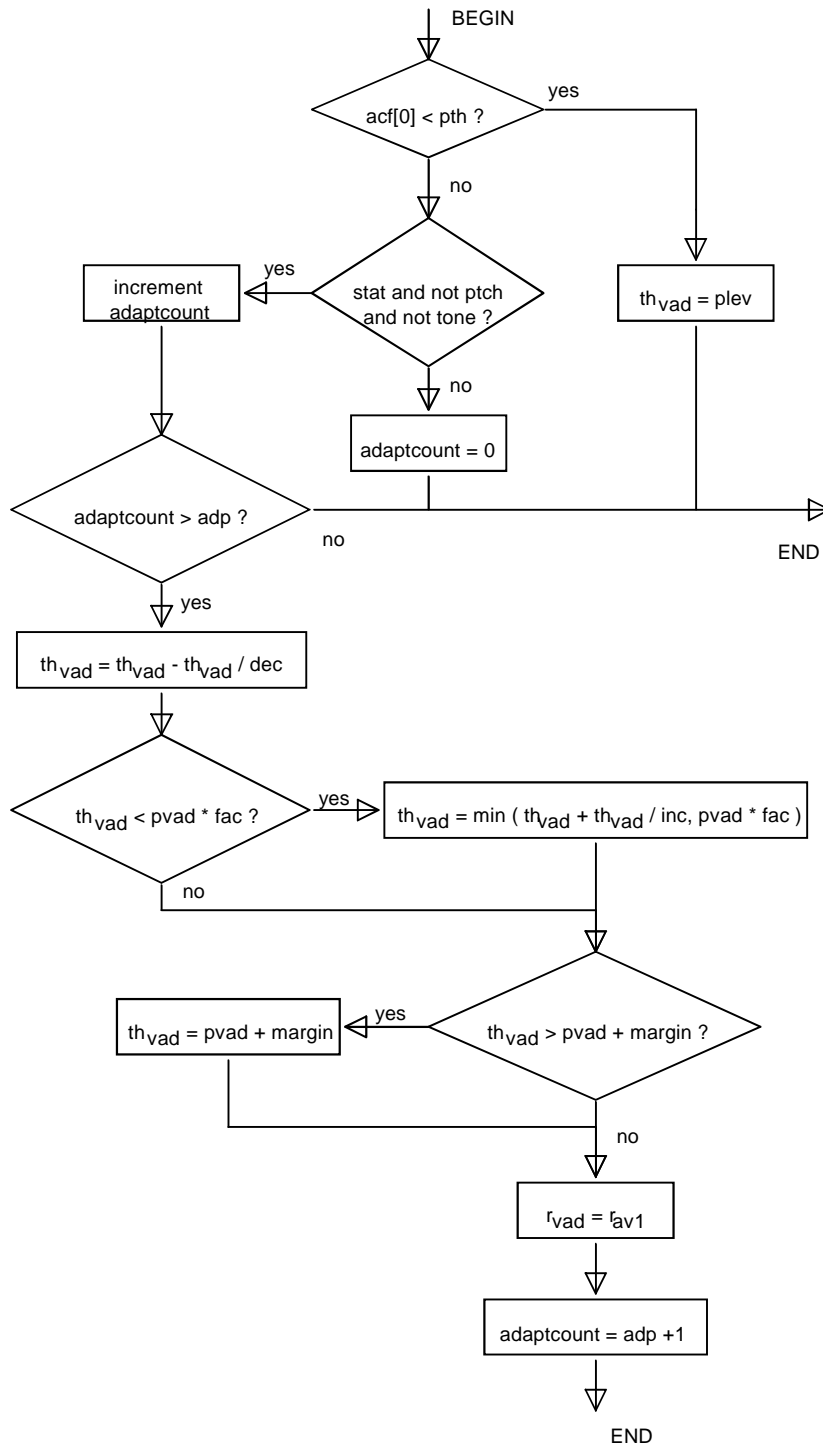| Constant	Value	Variable	Initial value
pth	130000	margin	69333340
plev	346667	adaptcount	0
fac	2,1	thvad	866656
adp	8	rvad[0]	6
inc	16	rvad[1..8]	All set to 0
dec	32		

Figure 2: Flow diagram for threshold adaptation

5.2.7   VAD decision

Prior to hangover the Boolean VAD decision is defined as:

```
vvad = (pvad > thvad)
```

5.2.8   VAD hangover addition

VAD hangover is only added to bursts of speech greater than or equal to burstconst blocks. The Boolean variable vadflag indicates the decision of the VAD with hangover included. The values of the constants and initial variable values are given in table 6. The hangover algorithm is as follows:

```
if (vvad)
    increment(burstcount)
else
    burstcount = 0

if (burstcount >= burstconst)
    {
    hangcount = hangconst
    burstcount = burstconst
    }

vadflag = (vvad OR (hangcount >= 0))

if (hangcount >= 0)
    decrement(hangcount)
```

Table 6: Constants and variables for VAD hangover addition

Constant	Value	Variable	Initial value
burstconst	3	burstcount	0
hangconst	10	hangcount	-1

5.2.9   Periodicity detection

The variables thvad and rvad are updated when the frequency spectrum of the input signal is stationary. However, vowel sounds also have a stationary frequency spectrum. The Boolean variable ptch indicates the presence of a periodic signal component and prevents adaptation of thvad and rvad. The variable ptch is updated every 20 ms and is true when periodicity (a vowel sound) is detected. The periodicity detector identifies the vowel sounds by comparing consecutive Long Term Predictor (LTP) lag values lags[1..2] which are obtained during the open loop pitch lag search from the speech codec defined in GSM 06.60 [4]. Cases in which one lag value is near the other are catered for, however the cases in which one lag value is a factor of the other, or in which both lag values have a common factor, are not.

```
lagcount = 0

for (j = 1; j <= 2; j++ )
    {
    smallag = maximum(lags[j],lags[j-1])-minimum(lags[j], lags[j-1])

    if ((smallag - lthresh) < 0)
        increment(lagcount)
    }

veryoldlagcount = oldlagcount
oldlagcount = lagcount

ptch = (oldlagcount + veryoldlagcount >= nthresh)
```

The values of constants and initial values are given in table 7. lags[0] = lags[2] of the previous frame.

ptch is calculated after the VAD decision and when the current LTP lag values lags[1..2] are available. This reduces the delay of the VAD decision.

Table 7: Constants and variables for periodicity detection

Constant	Value	Variable	Initial value
lthresh	2	ptch	1
nthresh	4	oldlagcount	0
		veryoldlagcount	0
		lags[0]	18

6　Computational description overview

The computational details necessary for the fixed point implementation of the speech transcoding and DTX functions are given in the form of an American National Standards Institute (ANSI) C program contained in GSM 06.53 [2]. This clause provides an overview of the modules which describe the computation of the VAD algorithm.

6.1　VAD modules

The computational description of the VAD is divided into three ANSI C modules. These modules are:

- vad_reset;

- vad_computation;

- periodicity_update.

The vad_reset module sets the VAD variables to their initial values.

The vad_computation module is divided into nine sub-modules which correspond to the blocks of figure 1 in the high level description of the VAD algorithm. The vad_computation module can be called as soon as the acf[0..8] and rc[1..4] variables are known. This means that the VAD computation can take place after the levinson routine of the second half of the frame in the speech encoder (GSM 06.60 [4]). The vad_computation module also requires the value of the ptch variable calculated in the previous frame.

The ptch variable is calculated by the periodicity_update module from the lags[1..2] variable. The individual lag values are calculated by the open loop pitch search routine in the speech encoder (GSM 06.60 [4]). The periodicity_update module is called after the VAD decision and when the current LTP lag values lags[1..2] are available.

6.2　Pseudo-floating point arithmetic

All the arithmetic operations follow the precision and format used in the computational description of the speech codec in GSM 06.53 [2]. To increase the precision within the fixed point implementation, a pseudo-floating point representation of some variables is used. This applies to the following variables (and related constants) of the VAD algorithm:

- pvad:　　　　Energy of filtered signal;

- thvad:　　　　Threshold of the VAD decision;

- acf0:　　　　Energy of input signal.

For the representation of these variables, two 16-bit integers are needed:

- one for the exponent (e_pvad, e_thvad, e_acf0);

- one for the mantissa (m_pvad, m_thvad, m_acf0).

The value e_pvad represents the lowest power of 2 just greater or equal to the actual value of pvad and the m_pvad value represents an integer which is always greater or equal to 16 384 (normalized mantissa). It means that the pvad value is equal to:

$$pvad = 2^{e\_pvad} * (m\_pvad/32768) \tag{7}$$

This scheme provides a large dynamic range for the pvad value and always keeps a precision of 16 bits. All the comparisons are easy to make by comparing the exponents of two variables. The VAD algorithm needs only one pseudo-floating point addition and multiplication. All the computations related to the pseudo-floating point variables require simple 16- or 32-bit arithmetic operations defined in the detailed description of the speech codec.

Some constants, represented by a pseudo-floating point format, are needed and symbolic names (in capital letters) for their exponent and mantissa are used; table 8 lists all these constants with the associated symbolic names and their numerical constant values.

Table 8: List of floating point constants

Constant	Exponent	Mantissa
pth	E_PTH = 17	M_PTH = 32500
margin	E_MARGIN = 27	M_MARGIN = 16927
plev	E_PLEV = 19	M_PLEV = 21667

Annex A (informative):
Simplified block filtering operation

Consider an 8th order transversal filter with filter coefficients a0..a8, through which a signal is being passed, the output of the filter being:

```
        8
  s'[n] = - SUM (a[i]*s[n-i])                                              (1)
        i=0
```

If we apply block filtering over 20 ms segments, then this equation becomes:

```
        8
  s'[n] = - SUM (a[i]*s[n-i])     ; n = 0..167                            (2)
        i=0                 ; 0 <= n-i <= 159
```

If the energy of the filtered signal is then obtained for every 20 ms segment, the equation for this is:

```
        167     8
  pvad = SUM (- SUM (a[i]*s[n-i]))²   ; 0 <= n-i <= 159                   (3)
        n=0     i=0
```

We know that:

```
        159
  acf[i] = SUM (s[n]*s[n-i])        ; i = 0..8                            (4)
        n=0                 ; 0 <= n-i <= 159
```

If equation (3) is expanded and acf[0..8] are substituted for s[n] then we arrive at the equations:

```
                    8
  pvad = r[0]*acf[0] + 2*SUM (r[i]*acf[i])                                (5)
                    i=1
```

Where:

```
        8-i
  r[i] = SUM (a[k]*a[k+i])        ; i = 0..8                              (6)
        k=0
```

Annex B (informative):
Pole frequency calculation

This annex describes the algorithm used to determine whether the pole frequency for a second order analysis of the signal frame is less than 385 Hz.

The filter coefficients for a second order synthesis filter are calculated from the first two unquantized reflection coefficients rc[1..2] obtained from the speech encoder. This is done using the step up routine described in GSM 06.53 [2]. If the filter coefficients a[0..2] are defined such that the synthesis filter response is given by:

$$H(z) = 1/(a[0] + a[1]z^{-1} + a[2]z^{-2}) \tag{1}$$

Then the positions of the poles in the Z-plane are given by the solutions to the following quadratic:

$$a[0]z^2 + a[1]z + a[2] = 0, \quad a[0] = 1 \tag{2}$$

The positions of the poles, z, are therefore:

$$z = re \pm j*sqrt(im), \qquad j^2 = -1 \tag{3}$$

where:

$$re = - a[1] / 2 \tag{4}$$

$$im = (4*a[2] - a[1]^2)/4 \tag{5}$$

If im is negative then the poles lie on the real axis of the Z-plane and the signal is not a tone and the algorithm terminates. If re is negative then the poles lie in the left hand side of the Z-plane and the frequency is greater than 2000 Hz and the prediction error test can be performed.

If im is positive and re is positive then the poles are complex and lie in the right hand side of the Z-plane and the frequency in Hz is related to re and im by the expression:

$$freq = arctan(sqrt(im)/re)*4000/pi \tag{6}$$

Having ensured that both im and re are positive the test for a pole frequency less than 385 Hz can be derived by substituting equations 4 and 5 into equation 6 and re-arranging:

$$(4*a[2] - a[1]^2)/a[1]^2 < tan^2(pi*385/4000) \tag{7}$$

or

$$(4*a[2] - a[1]^2)/a[1]^2 < 0.0973 \tag{8}$$

If this test is true then the signal is not a tone and the algorithm terminates, otherwise the prediction error test is performed.

Annex C (informative):
Change history

Change history					
SMG No.	**TDoc. No.**	**CR. No.**	**Clause affected**	**New version**	**Subject/Comments**
SMG#22				4.0.1	ETSI Publication
SMG#20				5.0.3	Release 1996 version
SMG#27				6.0.0	Release 1997 version
SMG#29				7.0.0	Release 1998 version
				7.0.1	Version update to 7.0.1 for Publication
SMG#31				8.0.0	Release 1999 version
				8.0.1	Update to Version 8.0.1 for Publication

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
03-2001	11				Version for Release 4		4.0.0
06-2002	16				Version for Release 5	4.0.0	5.0.0
12-2004	26				Version for Release 6	5.0.0	6.0.0
06-2007	36				Version for Release 7	6.0.0	7.0.0
07-2007					Makes matrices in §5.2.3 visible	7.0.0	7.0.1
12-2008	42				Version for Release 8	7.0.1	8.0.0
12-2009	46				Version for Release 9	8.0.0	9.0.0
03-2011	51				Version for Release 10	9.0.0	10.0.0
09-2012	57				Version for Release 11	10.0.0	11.0.0
09-2014	65				Version for Release 12	11.0.0	12.0.0
12-2015	70				Version for Release 13	12.0.0	13.0.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
03-2017	SA#75					Version for Release 14	14.0.0
06-2018	SA#80					Version for Release 15	15.0.0
2020-07	-	-	-	-	-	Update to Rel-16 version (MCC)	**16.0.0**

History

Document history		
V16.0.0	September 2020	Publication