



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

# UIT-T

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

# G.728

**Annexe I**  
(05/99)

SÉRIE G: SYSTÈMES ET SUPPORTS DE  
TRANSMISSION, SYSTÈMES ET RÉSEAUX  
NUMÉRIQUES

Systemes de transmission numériques – Equipements  
terminaux – Codage des signaux analogiques par des  
méthodes autres que la MIC

---

Codage de la parole à 16 kbit/s en utilisant la  
prédiction linéaire à faible délai avec excitation  
par code

**Annexe I: Masquage des pertes de trame ou de  
paquet dans un décodeur LD-CELP**

Recommandation UIT-T G.728 – Annexe I

(Antérieurement Recommandation du CCITT)

---

RECOMMANDATIONS UIT-T DE LA SÉRIE G  
**SYSTÈMES ET SUPPORTS DE TRANSMISSION, SYSTÈMES ET RÉSEAUX NUMÉRIQUES**

CONNEXIONS ET CIRCUITS TÉLÉPHONIQUES INTERNATIONAUX	G.100–G.199
<b>SYSTÈMES INTERNATIONAUX ANALOGIQUES À COURANTS PORTEURS</b>	
CARACTÉRISTIQUES GÉNÉRALES COMMUNES À TOUS LES SYSTÈMES ANALOGIQUES À COURANTS PORTEURS	G.200–G.299
CARACTÉRISTIQUES INDIVIDUELLES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX À COURANTS PORTEURS SUR LIGNES MÉTALLIQUES	G.300–G.399
CARACTÉRISTIQUES GÉNÉRALES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX HERTZIENS OU À SATELLITES ET INTERCONNEXION AVEC LES SYSTÈMES SUR LIGNES MÉTALLIQUES	G.400–G.449
COORDINATION DE LA RADIODÉLÉPHONIE ET DE LA TÉLÉPHONIE SUR LIGNES	G.450–G.499
<b>EQUIPEMENTS DE TEST</b>	
<b>CARACTÉRISTIQUES DES SUPPORTS DE TRANSMISSION</b>	G.600–G.699
<b>SYSTÈMES DE TRANSMISSION NUMÉRIQUES</b>	
EQUIPEMENTS TERMINAUX	G.700–G.799
Généralités	G.700–G.709
Codage des signaux analogiques en modulation par impulsions et codage	G.710–G.719
<b>Codage des signaux analogiques par des méthodes autres que la MIC</b>	<b>G.720–G.729</b>
Principales caractéristiques des équipements de multiplexage primaires	G.730–G.739
Principales caractéristiques des équipements de multiplexage de deuxième ordre	G.740–G.749
Caractéristiques principales des équipements de multiplexage d'ordre plus élevé	G.750–G.759
Caractéristiques principales des équipements de transcodage et de multiplication numérique	G.760–G.769
Fonctionnalités de gestion, d'exploitation et de maintenance des équipements de transmission	G.770–G.779
Caractéristiques principales des équipements de multiplexage en hiérarchie numérique synchrone	G.780–G.789
Autres équipements terminaux	G.790–G.799
RÉSEAUX NUMÉRIQUES	G.800–G.899
SECTION NUMÉRIQUE ET SYSTÈMES DE LIGNES NUMÉRIQUES	G.900–G.999

*Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.*

## **RECOMMANDATION UIT-T G.728**

### **CODAGE DE LA PAROLE A 16 kbit/s EN UTILISANT LA PREDICTION LINEAIRE A FAIBLE DELAI AVEC EXCITATION PAR CODE**

#### **ANNEXE I**

#### **Masquage des pertes de trame ou de paquet dans un décodeur LD-CELP**

##### **Résumé**

La présente annexe définit une extension pour un fonctionnement robuste des algorithmes de la Recommandation G.728 à des débits de 9,6, 12,8 et 16 kbit/s en présence d'effacements de trame ou de pertes de paquet dans des environnements mobiles ou par paquets.

##### **Source**

L'Annexe I à la Recommandation UIT-T G.728, élaborée par la Commission d'études 16 (1997-2000) de l'UIT-T, a été approuvée le 27 mai 1999 selon la procédure définie dans la Résolution n° 1 de la CMNT.

## AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

## NOTE

Dans la présente Recommandation, le terme *exploitation reconnue (ER)* désigne tout particulier, toute entreprise, toute société ou tout organisme public qui exploite un service de correspondance publique. Les termes *Administration*, *ER* et *correspondance publique* sont définis dans la *Constitution de l'UIT (Genève, 1992)*.

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT avait été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2000

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

## TABLE DES MATIÈRES

	<b>Page</b>
Annexe I – Masquage des pertes de trame ou de paquet pour le décodeur LD-CELP .....	1
I.1    Domaine d'application .....	1
I.2    Références normatives .....	1
I.3    Introduction.....	1
I.4    Principes de fonctionnement.....	2
I.4.1    Extrapolation d'excitation.....	2
I.4.2    Filtre LPC .....	3
I.4.3    Adaptation en boucle de codage LPC et prédicteurs de gain.....	4
I.4.4    Post-filtre .....	5
I.4.5    Limitation de l'accroissement de gain après un effacement de trame.....	6
I.5    Pseudo-code pour le masquage d'effacement de trame.....	6
I.5.1    Nouvelle boucle de programme principal pour le décodeur.....	6
I.5.2    Bloc 31SF – Positionner les fanions et le facteur de normalisation pour l'effacement de trame .....	9
I.5.3    Bloc 31FE – Extrapolation du signal d'excitation .....	10
I.5.4    Bloc 31E – Mise à jour du signal d'excitation .....	11
I.5.5    Bloc 43FE – Module de fenêtre hybride.....	12
I.5.6    Bloc 47AF – Limiteur de gain logarithmique après un effacement de trame .....	13
I.5.7    Bloc 49FE – Module de fenêtre hybride pour le filtre de synthèse .....	13
I.5.8    Bloc 51FE – Module d'expansion de largeur de bande pour des effacements de trame .....	17
I.5.9    Bloc 97FE – Mise à jour de GSTATE pendant les effacements de trame.....	17
I.5.10   Bloc 98AF – Limiteur de gain logarithmique après un effacement de trame .....	19
I.6    Paramètres de code et variables supplémentaires .....	19
Appendice I.I – Valeurs utilisées pour la normalisation de ETPAST pendant les effacements de trame .....	20



## Recommandation G.728

### CODAGE DE LA PAROLE A 16 kbit/s EN UTILISANT LA PREDICTION LINEAIRE A FAIBLE DELAI AVEC EXCITATION PAR CODE

#### ANNEXE I

#### Masquage des pertes de trame ou de paquet pour le décodeur LD-CELP

(Genève, 1999)

##### I.1 Domaine d'application

La présente annexe traite des procédés de masquage de la perte d'informations de flux binaire occasionnée par des pertes de paquets ou de trames dans le canal de communication. En fonctionnement normal, le décodeur se comporte d'une manière identique à celle décrite dans le texte principal de la Recommandation G.728 ou dans l'Annexe G de la Recommandation G.728 pour un fonctionnement à 16 kbit/s ou dans l'Annexe H de la Recommandation G.728 pour un fonctionnement à 12,8 ou 9,6 kbit/s. La modification proposée dans la présente annexe porte uniquement sur le décodeur durant les phases d'indisponibilité de flux binaire. On fait l'hypothèse que cette perte de flux binaire est signalée au décodeur par un moyen externe. La présente annexe ne joue pas de rôle crucial pour le fonctionnement normal en mode G.728.

##### I.2 Références normatives

- Recommandation CCITT G.728 (1992), *Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code.*
- Recommandation UIT-T G.728, Annexe G (1994), *Spécification d'un dispositif à virgule fixe fonctionnant à 16 kbit/s.*
- Recommandation UIT-T G.728, Annexe H (1997), *Fonctionnement de l'algorithme LD-CELP à débit variable, principalement pour équipements DCME à des débits inférieurs à 16 kbit/s.*

##### I.3 Introduction

La présente annexe décrit les modifications nécessaires pour permettre au décodeur G.728 de traiter des effacements de trame dans le flux binaire reçu. Ces modifications incluent l'extrapolation d'excitation, l'expansion de largeur de bande pour le filtre LPC, la prolongation d'une partie de l'adaptation en boucle utilisant l'excitation extrapolée, la prolongation du post-filtrage et de l'adaptateur de post-filtre utilisant l'excitation extrapolée et la limitation du taux d'accroissement du gain adaptatif en boucle après des effacements de trame. De telles modifications de décodeur préservent la compatibilité de flux binaire avec la Recommandation G.728 tout en introduisant une complexité supplémentaire négligeable.

Toutes les modifications sont confinées au niveau du décodeur afin de préserver la compatibilité de flux binaire avec la Recommandation G.728. Elles prennent effet uniquement durant les trames effacées et les quelques premières "bonnes" trames succédant aux effacements de trame. Le système proposé fonctionne par ailleurs exactement de la même façon que dans le cas de la Recommandation G.728 pendant toutes les autres "bonnes" trames.

La présente annexe fait l'hypothèse que le lecteur est familiarisé avec le document de spécification de la Recommandation G.728. Il n'analysera pas l'algorithme G.728 et se limitera à la description des modifications apportées à la Recommandation G.728. Pour éviter toute confusion, nous réserverons

désormais le terme "*trame*" pour désigner la taille de bloc associé à l'effacement de trame (en général un multiple de 2,5 ms) et le terme "*cycle d'adaptation*" pour désigner le bloc de 2,5 ms d'adaptation des paramètres de filtre G.728.

Sur la base des informations en provenance de l'UIT-R (précédemment le CCIR) portant sur le cas d'un canal avec effacement de trame, une trame de bits reçue peut être considérée, soit comme totalement correcte (sans erreur de bit), soit comme totalement erronée (avec un taux d'erreur de bit dans la trame voisin de 50%). On fait par ailleurs l'hypothèse qu'un code de détection d'erreurs a été utilisé, de sorte que le récepteur peut distinguer de manière fiable les bonnes trames des trames erronées et fournir ces informations au décodeur de parole G.728.

La présente annexe est structurée de la manière suivante: le paragraphe I.4 "principes de fonctionnement", contient des sous-paragraphes qui décrivent les modifications du décodeur pendant le masquage de l'effacement de trame. Le paragraphe I.5 "pseudo-code pour le masquage de l'effacement de trame", contient la totalité du pseudo-code nécessaire pour les modifications apportées au décodeur. Le pseudo-code est donné, à des fins d'exhaustivité, aussi bien pour les spécifications en virgule flottante que pour les spécifications en virgule fixe. Enfin, le paragraphe I.6 contient des tableaux pour les paramètres et les variables de codeur supplémentaires ajoutés pour les opérations de masquage d'effacement de trame.

## **I.4 Principes de fonctionnement**

Les modifications de la Recommandation G.728 pour les trames effacées portent sur l'extrapolation des coefficients d'excitation et de filtre, la prolongation de certaines opérations d'adaptation en boucle, les opérations pour le codage LPC et les prédicteurs de gain ainsi que sur la limitation du taux de croissance du gain après l'effacement de trame. Ces modifications sont décrites séparément dans les paragraphes qui suivent.

### **I.4.1 Extrapolation d'excitation**

Examinons d'abord la Figure 3/G.728. Aucune extrapolation n'est faite pour la sortie du répertoire d'excitation VQ (bloc 29). On effectue à la place l'extrapolation de la sortie de l'unité de normalisation du gain (bloc 31), ou *excitation avec gain normalisé*, qui est représentée par le tableau ET() dans le pseudo-code du 5.14/G.728. On conserve uniquement dans la mémoire du décodeur G.728 d'origine 5 échantillons de l'excitation ET(). Il est maintenant nécessaire, pour extrapoler l'excitation ET(), de maintenir un tableau supplémentaire ETPAST() mémorisant les vecteurs ET() actuels et passés.

Se référer au pseudo-code du 5.14/G.728. Dans le décodeur G.728, le module d'extraction de période de hauteur du son (bloc 82) fournit une période de hauteur du son KP. Le calculateur de dérivation de prédicteur de hauteur du son (bloc 83) détermine la pondération de dérivation optimale pour un prédicteur de hauteur du son avec une seule dérivation pour la parole décodée. Cette pondération de dérivation PTAP est utilisée comme un indicateur sommaire du caractère voisé ou non du cycle actuel. Si la pondération PTAP est supérieure au seuil PPFTH (avec PPFTH = 0,6) le cycle d'adaptation actuel est alors considéré comme voisé et le post-filtre à long terme est appliqué; le post-filtre à long terme est désactivé dans le cas contraire. Comme on dispose déjà de la hauteur du son et d'un indicateur de voisage dans le décodeur G.728 d'origine, il est alors possible d'utiliser directement ces informations dans l'extrapolation d'excitation sans avoir à les recalculer.

La valeur de l'indicateur de voisage PTAP obtenue lors du dernier cycle d'adaptation de la dernière bonne trame est utilisée au début de la première trame manquante d'un effacement pour déterminer si la parole est voisée ou non. L'excitation est extrapolée de manière différente pour les trames voisées et non voisées. Un seuil de voisage de valeur  $V_{TH} = PPFTH/1,4$  inférieur à PPFTH est utilisé à cette fin.



La trame est considérée comme voisée si  $PTAP > VTH$ . L'extrapolation de l'excitation  $ET()$  est faite dans ce cas par reproduction périodique d'une version atténuée des KP derniers échantillons du tableau  $ETPAST()$ . On peut, en variante, présenter cette opération de la manière suivante: soit une fenêtre rectangulaire avec des valeurs d'échantillons unité, située à l'emplacement correspondant au vecteur  $ETPAST()$  actuel. Cette fenêtre rectangulaire est décalée vers l'arrière dans le temps de KP échantillons et les 5 échantillons sont extraits puis normalisés par multiplication. Le résultat fournit le vecteur  $ET()$  souhaité. Ce vecteur ( $ET$ ) extrapolé est alors copié vers l'emplacement adéquat dans le tableau  $ETPAST()$  afin de mettre à jour le vecteur actuel dans  $ETPAST()$ . Le processus se répète pour tous les vecteurs au sein de la trame effacée. Cette opération produit, pendant les trames effacées, une série de valeurs d'excitation  $ET()$  avec une enveloppe périodique et des amplitudes décroissantes. La période est égale à KP échantillons. Cette extrapolation périodique du tableau  $ETPAST()$  se poursuit jusqu'à la réception d'une bonne trame. Le facteur de normalisation  $FESCALE$  est égal à 0,8 pendant les premières 20 ms d'un effacement et décroît ensuite d'un facteur 0,2 toutes les 10 ms suivantes. Ces facteurs de normalisation sont mémorisés dans le tableau  $VOICEDFEGAIN()$ . Le facteur  $FESCALE$  est positionné sur 0 si l'effacement dure plus de 50 ms afin d'éviter l'introduction de signaux synthétiques de longue durée dans le signal de parole de sortie.

Le dernier cycle d'adaptation est considéré comme "non voisé" si  $PTAP \leq VTH$ . Il s'agit, au sens strict, de tout ce qui diffère d'un cycle "voisé" pouvant être éventuellement un silence, une transition, etc. On n'effectue pas d'extrapolation périodique dans un tel cas parce qu'il n'est pas souhaitable d'introduire une périodicité artificielle qui n'était pas présente à l'origine dans le signal de parole. La périodicité est évitée par la répétition aléatoire de 5 échantillons consécutifs pris parmi les 140 derniers échantillons du tableau  $ETPAST()$ . Un générateur de nombres aléatoires est utilisé pour fournir un nombre entier aléatoire compris entre 5 et 140. Ce nombre aléatoire détermine le nombre d'échantillons de décalage vers l'arrière de la fenêtre rectangulaire glissante utilisée pour l'extraction des 5 échantillons du tableau  $ETPAST()$  qui constituent l'excitation  $ET()$  extrapolée. La série de nombres aléatoires peut, si on le souhaite, être calculée à l'avance et mémorisée.

L'amplitude moyenne de chaque vecteur d'excitation extrapolée est mise en correspondance avec l'amplitude moyenne de l'excitation qui était présente pendant les 5 dernières millisecondes (40 échantillons) avant l'effacement, afin de garantir que le niveau moyen du signal généré pendant un effacement de trame non voisée reste semblable à son historique récent. Si on a déterminé au début de la première trame effacée que la parole n'est pas voisée, on effectue alors le calcul de l'amplitude moyenne  $AVMAG$  des 8 derniers vecteurs de  $ETPAST()$ . On calcule d'abord, lors de l'extrapolation de chaque vecteur d'excitation, l'amplitude ( $MAG$ , *magnitude*) du vecteur constitué de 5 échantillons consécutifs pris au hasard parmi les 140 derniers échantillons dans  $ETPAST()$ . L'amplitude  $AVMAG$  est divisée par l'amplitude  $MAG$  pour obtenir un facteur de normalisation. Ce facteur est utilisé pour multiplier chacun des 5 échantillons du vecteur d'excitation extrapolée actuel. Cette opération garantit que tout vecteur extrapolé avec un gain normalisé aura une amplitude moyenne égale à  $AVMAG$ . L'amplitude  $AVMAG$  est réduite de 20% toutes les 10 ms si l'effacement dépasse 20 ms, afin d'éviter la génération de signaux artificiels résultant d'effacements de très longue durée. Les facteurs d'atténuation sont stockés dans le tableau  $UNVOICEDFEGAIN()$ .

#### **I.4.2 Filtre LPC**

L'excitation avec gain normalisé  $ET()$  est extrapolée comme décrit ci-dessus pendant les trames effacées, mais un filtre de synthèse LPC reste nécessaire pour produire la parole en sortie. On utilise une version "adoucie" du dernier ensemble valide de coefficients LPC de la dernière bonne trame. Cet "adoucissement" est réalisé par une expansion de largeur de bande, telle qu'elle est faite dans le module d'expansion de largeur de bande du bloc 51 (5.6/G.728), avec la différence que le facteur d'expansion de largeur de bande  $FAC$  (5.1/G.728) est égal à 0,97 et non  $253/256 \approx 0,9883$ . Dans la première trame effacée qui suit une bonne trame, cette opération d'expansion de bande passante se fait à partir du troisième vecteur du premier cycle d'adaptation, au moment exact où les coefficients LPC sont mis à jour dans le cas normal sans effacement de trame.

Soient  $a_i$  les coefficients LPC d'ordre  $i$  utilisés dans le dernier cycle d'adaptation de la dernière bonne trame. Le nouvel ensemble de coefficients LPC adoucis est obtenu par les formules  $a'_i = (0,97)^i a_i$ ,  $i = 1, 2, \dots, 50$ . Ce nouvel ensemble  $\{a'_i\}$  est utilisé pendant les 10 premières millisecondes de l'effacement (si cette durée est atteinte), même si des cycles d'adaptation suivants sont présents dans la même trame. Si l'effacement dure plus de 10 ms, cet ensemble  $\{a'_i\}$  subit ensuite une expansion de bande avec un facteur égal à 0,97. En d'autres termes, si l'effacement dure plus de 10 ms, les coefficients LPC sont de nouveau mis à jour au moment du troisième vecteur du cinquième cycle d'adaptation et le coefficient d'adaptation de rang  $i$  est donné par l'égalité  $a''_i = (0,97)^i a'_i = (0,97)^{2i} a_i$ . Cet ensemble  $\{a''_i\}$  est de nouveau utilisé pour la totalité des 10 ms suivantes de l'effacement. De même, si l'effacement se poursuit pendant une durée de  $(k * 10 \text{ ms})$ , les coefficients LPC correspondent alors à une version avec expansion de largeur de bande du dernier ensemble valide de coefficients LPC, avec un facteur d'expansion de largeur de bande égal à  $(0,97)^k$ . L'expansion de largeur de bande croît avec l'éloignement de la dernière bonne trame; ce processus se poursuit jusqu'à la réception de la bonne trame suivante. Par la suite, lorsqu'une mauvaise trame est de nouveau reçue, le processus redémarre avec un facteur d'expansion de largeur de bande égal à 0,97.

Une action spéciale est nécessaire pour la première trame d'un effacement afin d'obtenir les coefficients LPC les plus récents pour l'expansion de largeur de bande. En effet, si l'expansion de largeur de bande est faite directement sur le tableau A() résultant de la dernière bonne trame, ce tableau correspond alors à la fonction d'autocorrélation calculée sur la base de la parole synthétisée *avant* la dernière bonne trame; il est donc périmé au moment de la première trame effacée. Les procédures de fenêtre hybride et la boucle récursive de Durbin pour l'analyse LPC d'ordre 50 sont effectuées normalement lors de la première trame effacée, afin d'obtenir le tableau A() de coefficients de prédiction LPC pour l'expansion de largeur de bande. Ce tableau A() subit ensuite une expansion de largeur de bande avec un facteur de 0,97. Les coefficients LPC qui en résultent sont alors plus à jour, parce qu'ils sont obtenus en utilisant la parole synthétisée jusqu'au dernier vecteur de la dernière bonne trame. Le traitement spécial s'arrête si l'effacement dure plus de 10 ms et l'expansion de largeur de bande se fait alors à partir du dernier ensemble de tableaux A() présent en mémoire, qui a fait au préalable l'objet de l'expansion de largeur de bande décrite dans le paragraphe précédent.

Il convient de noter qu'une telle expansion de largeur de bande du prédicteur de gain n'est pas nécessaire pendant les trames effacées. Comme l'excitation ET() avec gain normalisé a déjà été extrapolée, elle est utilisée directement en entrée du filtre de synthèse LPC pour effectuer la synthèse de la parole. Il n'est donc pas nécessaire de calculer le gain d'excitation avec adaptation en boucle ou de mettre à jour les coefficients du prédicteur de gain pendant les trames effacées.

### **I.4.3 Adaptation en boucle de codage LPC et prédicteurs de gain**

Les valeurs de sortie de l'adaptateur de filtre de synthèse en boucle et de l'adaptateur de gain vectoriel en boucle ne sont pas nécessaires pendant les trames effacées (blocs 33 et 30 dans la Figure 3/G.728). Il est toutefois nécessaire de poursuivre le traitement des fonctions vitales du processus d'adaptation en boucle dans ces deux blocs afin de minimiser les discontinuités des états internes lors de l'arrivée des bonnes trames suivantes. Les "fonctions vitales" sont les parties des opérations d'adaptation en boucle qui affecteront les sorties des blocs 30 et 33 dans des trames futures.

Analysons le bloc 30 (adaptateur de gain vectoriel en boucle) qui est identique au bloc 20 (5.7/G.728 et Figure 6/G.728). Dans le cas du bloc 30 (adaptateur de gain vectoriel en boucle), les fonctions vitales concernent les blocs 67, 39, 40, 41, 42 (qui coopèrent pour le calcul de gain logarithmique sans décalage) ainsi que certaines parties du bloc 43 (module de fenêtre hybride) et du bloc 46 (prédicteur linéaire de gain logarithmique). Les blocs 67 et 39 à 42 utilisent l'excitation ET() extrapolée pour le calcul du gain logarithmique en l'absence de décalage qui est utilisé pour la mise à jour du tableau GTMP() constituant l'entrée du bloc 43 et du tableau GSTATE() constituant la

mémoire de filtre du prédicteur linéaire de gain logarithmique. Il est nécessaire d'utiliser le tableau GTMP() dans le bloc 43 (module de fenêtre hybride) pour la mise à jour du tampon SBLG(); il faut également mettre à jour le tableau REXPLG() contenant la composante récursive de la fonction d'autocorrélation. Ces opérations maintiennent les états internes qui affecteront les sorties futures du bloc 30. Toutes les autres opérations, telles que la boucle récursive de Levinson-Durbin (bloc 44), l'expansion de largeur de bande (bloc 45), le calcul d'exponentiation (bloc 48), etc. ne sont pas nécessaires pendant l'effacement de trame parce que leurs résultats ne sont pas conservés dans le cas normal.

Il convient de noter que les coefficients du prédicteur de gain GP() ne sont pas mis à jour pendant un effacement de trame et que le premier vecteur après l'effacement de trame utilisera le dernier tableau GP() calculé dans la dernière bonne trame avant l'effacement de trame. Ceci résulte du fait que le tableau GP() n'est pas mis à jour avant le deuxième vecteur du cycle d'adaptation G.728. On ne tente pas d'effectuer une mise à jour plus rapide du premier vecteur du tableau GP() qui compliquerait inutilement la structure de commande de l'algorithme. L'écrêtage de gain décrit ci-dessous au I.4.5 fournira également une protection contre une excursion de gain élevée pour ce premier vecteur. L'utilisation d'un prédicteur de gain légèrement périmé pour un seul vecteur a de ce fait peu de chance d'entraîner une différence perceptible de la qualité de la parole.

Considérons ensuite le bloc 33 (adaptateur de filtre de synthèse en boucle du décodeur au 5.14/G.728) qui est identique au bloc 23 (5.6/G.728 et Figure 5/G.728). Il existe deux opérations vitales pour le bloc 33:

- 1) mise à jour du tampon SB() dans le module de fenêtre hybride (bloc 49) au moyen de la parole synthétisée (qui est obtenue en faisant passer la dernière excitation ET() extrapolée à travers la version avec expansion de largeur de bande du dernier filtre LPC valide;
- 2) calcul normal du tableau REXP() au moyen du tampon SB().

La plupart des autres parties du bloc 33 sont inutiles pendant les trames effacées. Ceci résulte du fait que les coefficients LPC ne sont pas mis à jour par les sorties des blocs 50 et 51 mais par une version avec expansion de largeur de bande des coefficients LPC de la dernière bonne trame. Il reste toutefois nécessaire de traiter une autre partie du bloc 49 et une petite partie du bloc 50 en vue du post-filtrage. Ce traitement est analysé dans le paragraphe I.4.4.

Comme nous venons de le voir, un grand nombre d'opérations de calcul intensif des blocs 30 et 33 n'est pas nécessaire pendant les trames effacées. Il est possible d'utiliser la puissance de calcul économisée (ou le temps de processeur DSP) pour effectuer l'extrapolation d'excitation et l'expansion de largeur de bande du dernier bon filtre LPC. Ces deux calculs impliquant une charge très faible, il en résulte que les modifications apportées au décodeur G.728 n'augmentent pas la complexité générale des calculs.

#### **I.4.4 Post-filtre**

Le fonctionnement du post-filtre (5.14/G.728) et de l'adaptateur de post-filtre (blocs 34 et 35 dans la Figure 3/G.728) est le même dans le cas de bonnes trames et effacées, afin de préserver la simplicité du déroulement général et du programme. Plusieurs conséquences en résultent.

En premier lieu, l'adaptateur de post-filtre continue à avoir besoin des coefficients de prédicteur d'ordre 10 et du premier coefficient de réflexion de l'adaptateur de filtre de synthèse en boucle (bloc 33 dans la Figure 3/G.728). Il en résulte qu'il est nécessaire de poursuivre, dans le bloc 49 (module de fenêtre hybride, Figure 5/G.728), le calcul des valeurs des coefficients RTMP(1) à RTMP(11), sans qu'il soit toutefois nécessaire de calculer les coefficients RTMP(12) à RTMP(51). Il est nécessaire, de même, d'effectuer dans le bloc 50 la boucle récursive de Durbin d'ordre 1 à 10, mais non les boucles d'ordre 11 à 50. Il convient de noter que durant les trames effacées les coefficients LPC d'ordre 10 et le premier coefficient de réflexion sont effectivement obtenus à partir de la parole synthétisée qui est pilotée en fin de compte par l'excitation extrapolée ET().

En deuxième lieu, le post-filtre met à jour durant les trames effacées la totalité de ses coefficients à partir de la parole synthétisée et utilise les coefficients mis à jour pour filtrer cette parole synthétisée afin de produire la parole synthétisée finale. Contrairement au cas du filtre LPC, il n'est tenté aucune opération de "gel" des coefficients du post-filtre ou d'utilisation d'une version de post-filtre avec expansion de largeur de bande de la dernière bonne trame. Le post-filtre est laissé intentionnellement dans un état "flottant" avec la parole qui est synthétisée à partir de l'excitation extrapolée. Ceci garantit que les pics et les creux du spectre de réponse en fréquence du post-filtre correspondent à tout instant à ceux de la parole synthétisée. Ceci permet d'éviter une suppression ou une atténuation éventuelle des pics du spectre par les creux du spectre de la réponse en fréquence du post-filtre résultant d'un discordance entre pics et creux.

Le post-filtre poursuit la mise à jour de la période de hauteur du son KP et de la dérivation de prédicteur de hauteur du son PTAP pendant un effacement, mais il n'est pas nécessaire d'utiliser les valeurs mises à jour pour modifier, pendant la durée de l'effacement, le fanion VOICED de décision voisée/non voisée et la période de hauteur du son FEDELAY mémorisée. Ces valeurs sont positionnées une fois au début d'un effacement et ne sont plus modifiées pendant la durée de l'effacement.

#### **I.4.5 Limitation de l'accroissement de gain après un effacement de trame**

On a observé qu'après un effacement de trame de très longue durée (pouvant aller jusqu'à 100 ms) ou lorsqu'un effacement de trame était précédé par une région d'une période de parole de faible amplitude et suivi d'une période de forte amplitude, la parole décodée après l'effacement de trame présentait parfois un "pop" de forte amplitude ou un dépassement important de gain. Cet effet est dû au fait que le tampon de gain logarithmique SBG() dans la fenêtre hybride contenait des gains de très faible niveau au moment de l'effacement de trame et que ceux-ci ont été suivis de gains avec des niveaux considérablement plus élevés lors des quelques premières bonnes trames après l'effacement. Il en résulte que l'analyse LPC a produit un prédicteur de gain qui tente de compenser rapidement la variation brusque du niveau de gain. Ceci produit un accroissement excessif et trop rapide du gain prédit (c'est-à-dire adapté en boucle) avec un dépassement important du gain qui est à l'origine du "pop".

Le problème peut être évité en limitant le taux d'accroissement du gain adapté en boucle pendant les quelques premières bonnes trames après un effacement de trame. On limite en conséquence le gain après un effacement de trame à une valeur n'excédant pas FEGAINMAX = +2 dB par vecteur à 5 échantillons. Si l'accroissement de la prédiction de gain logarithmique est supérieur à +2 dB par vecteur, on écrête alors ce gain à un niveau de 2 dB au-dessus du gain logarithmique le plus récent. La durée de cet "écrêtage de gain" dépend de la durée de l'effacement de trame. La durée de l'écrêtage de gain est égale à la durée de l'effacement de trame si cette dernière est inférieure ou égale à AFTERFEMAX = 16 (40 ms). La durée de l'écrêtage de gain est limitée aux 40 premières millisecondes qui suivent l'effacement si la durée est supérieure à 40 ms. Les constantes FEGAINMAX et AFTERFEMAX sont des paramètres réglables du codeur.

### **I.5 Pseudo-code pour le masquage d'effacement de trame**

Le pseudo-code pour le programme principal du décodeur est donné ci-dessous en premier. On indique uniquement la succession d'exécution des blocs sans donner les détails de transmission des paramètres. Les paragraphes suivants fournissent les pseudo-codes pour les nouveaux sous-programmes.

#### **I.5.1 Nouvelle boucle de programme principal pour le décodeur**

Les boucles de programme principal en virgule fixe et en virgule flottante ne diffèrent que pour l'adaptateur de gain vectoriel en boucle qui a été modifié dans la spécification en virgule fixe. Une seule version du pseudo-code est fournie dans la mesure où les modifications de la boucle principale sont relativement minimales. On utilisera les expressions de pré-processeur C **#ifdef**, **#else** et **#endif**

pour indiquer quels sont les segments appartenant à chacune des spécifications en virgule fixe et en virgule flottante. Le pseudo-code figurant entre une expression `#ifdef FIXPT` et l'expression `#else` suivante concerne la spécification en virgule fixe, alors que le pseudo-code figurant entre cette expression `#else` et l'expression `#endif` suivante concerne la spécification en virgule flottante.

Positionner toutes les variables de décodeur à leurs valeurs initiales.

```
ILLCOND=.FALSE.
ILLCONDG=.FALSE.
FERROR=.FALSE      | VRAI si la trame actuelle est effacée
FESIZE=4           | Nombre de cycles d'adaptation de 2,5 ms dans un
                  | effacement de trame (FE)
                  | par ex.: 1 pour un effacement FE de 2,5 ms, 4
                  | pour un effacement FE de 10 ms
                  | Un effacement FE peut être un multiple quelconque de 2,5 ms.
                  | Positionné par l'utilisateur.
ICOUNT=4          | Compteur de vecteur, 1,2,3,4,1,2,..., .625 ms chacun
ADDCOUNT=FESIZE  | Compteur de cycle d'adaptation, 2,5 ms chacun
                  | comptage à partir de 1,2,...,FESIZE,1,2,...,FESIZE ,...;
FECOUNT=0        | Nombre de cycles d'adaptation de 2,5 ms supprimés de manière
                  | consécutive
AFTERFE=0       | Nombre de cycles d'adaptation de 2,5 ms après un effacement FE pour la
                  | limitation du gain. Comptage dégressif.
OGAINDB=-32     | Dernière valeur de prédiction de gain en dB
```

VEC\_LOOP:

```
  If ICOUNT = 4, do the next 14 lines
    ICOUNT = 0          | reset vector counter
                      | La réinitialisation du compteur de vecteur, si le dernier cycle
                      | d'adaptation n'était pas
                      | effacé et si le gain n'était pas écrêté, permet de réduire le nombre
                      | de cycles restants
                      | qui sont nécessaires pour l'écrêtage du gain.
    If FERROR = .FALSE. and AFTERFE > 0, do AFTERFE = AFTERFE - 1
                      | Vérifier si la trame suivante est effacée
    If ADCOUNT = FESIZE, do the next 7 lines
      ADCOUNT = 0      | réinitialiser le compteur de cycle d'adaptation
      read FERROR     | lire si une nouvelle erreur est survenue
                      | positionner AFTERFE lors de la première bonne trame après un effacement
                      | de manière à écrêter le gain pour un petit nombre de prochaines
                      | bonnes trames.
      If FERROR = .FALSE. and FECOUNT > 0, do the next 4 lines
        AFTERFE = AFTERFE + FECOUNT | ajouter la longueur de FE à AFTERFE
        If AFTERFE > AFTERFEMAX, do the next line
          AFTERFE = AFTERFEMAX    | écrêter le gain pendant 40 ms au maximum
          FECOUNT = 0             | Réinitialisation de FECOUNT
      If FERROR = .TRUE., do the next 2 lines
        FECOUNT = FECOUNT + 1    | Mise à jour de FECOUNT
                      | Appeler le bloc 31SF au début d'un effacement FE et ensuite
                      | toutes les 10 ms
                      | de manière à initialiser les fanions FE et mettre à jour les
                      | gains d'excitation FE
                      | par exemple, lorsque FECOUNT est égal à 1, 5, 9, 13, ...
        If (FECOUNT & 3) = 1, do block31SF
          ADCOUNT = ADCOUNT + 1   | mise à jour du compteur de cycle d'adaptation
          ICOUNT = ICOUNT + 1    | mise à jour du compteur de vecteur
```

Obtenir ICHAN du vecteur actuel à partir du tampon d'entrée.

Obtenir l'index de forme IS et l'index de gain IG à partir de ICHAN.

```
                      | vérifier s'il est nécessaire de mettre à jour
                      | les coefficients LPC.
```

If ICOUNT = 3, do the next 2 lines

If FERROR = .FALSE. and ILLCOND = .FALSE., do block 51.

If FERROR = .TRUE. and (FECOUNT & 3) = 1, do block 51FE

If ICOUNT = 2 and ILLCONDG=.FALSE. and FERROR = .FALSE., do block 45.

do block 46

```
                      | GSTATE(1:9) est décalé d'une position vers le
                      | bas
```

`#ifdef FIXPT`

If FERROR = .FALSE., do the next 3 lines

```

do blocks 98AF, 99 and 48 | obtenir le gain adapté en boucle
OGAINDB = output of block 98AF | pour la limitation de la croissance du gain
                                | après effacement FE
do blocks 29 and 31         | vecteur de code d'excitation avec
                                | sélection d'échelle
#else
  If FERROR = .FALSE., do the next 3 lines
  do blocks 47AF and 48     | obtenir le gain adapté en boucle
  OGAINDB = output of block 47AF | pour la limitation de la croissance du gain
                                | après effacement FE
  do blocks 29 and 31       | vecteur de code d'excitation avec
                                | sélection d'échelle
#endif
  If FERROR = .TRUE., do the next line
  do block 31FE             | obtenir l'excitation extrapolée

do block 31E               | mise à jour ETPAST()
do block 32                | obtenir ST()
If ICOUNT = 1, do block 85 | mise à jour des coefficients à court terme
                            | du post-filtre
do block 81                | filtrage LPC inverse d'ordre 10
If ICOUNT = 3, do the next 3 lines
  do block 82               | extraction de la période de hauteur du son
  do block 83               | calcul de dérivation de la période de hauteur
                            | du son
  do block 84               | mise à jour des coefficients à long terme
                            | du filtre
do block 71                | post-filtre à long terme
do block 72                | post-filtre à court terme
do blocks 73 and 74        | calcul des sommes des valeurs absolues
do block 75                | rapport des sommes des valeurs absolues
do block 76                | facteur de normalisation du filtre passe-bas
do block 77                | commande du gain de sortie du post-filtre

#ifdef FIXPT
  If FERROR = .FALSE., do the next 2 lines
  do blocks 93, 94, 96, and 97 | mise à jour du gain logarithmique
  GSTATE(1) = output of block 97 | mise à jour de la mémoire du prédicteur de gain
#else
  If FERROR = .FALSE., do the next 2 lines
  do blocks 39, 40 and 42     | mise à jour du gain logarithmique
  GSTATE(1) = output of block 42 | mise à jour de la mémoire du prédicteur de gain
#endif

  If FERROR = .TRUE., do the next 3 lines
  do block 97FE              | calculer le gain logarithmique du tableau ET()
                            | extrapolé
  OGAINDB = output of 97FE   | sauvegarder pour limiter l'accroissement de gain
                            | après FE
  GSTATE(1) = output of block 97FE | puis mettre à jour la mémoire de prédicteur
                            | de gain

  I=(ICOUNT-1)*IDIM          | I=adresse de départ du tableau STTMP()
  copy ST(1:5) to STTMP(I+1:I+5) | mise à jour du tableau STTMP()
  NLSSTTMP(ICOUNT)=NLSST     | mise à jour du tableau NLSSTTMP()
                            | fin du traitement unitaire par vecteur

                            | début du traitement unitaire par cycle
  If ICOUNT = 4, do the next 6 lines
  do block 49FE              | output ill-condition flag = ILLCOND
  do block 50, order 1 to 10 | output pred coef= ATMP() with NLSATMP
                            | output ill-condition flag = ILLCOND
  NLSAPF=NLSATMP            | sauvegarder le prédicteur d'ordre 10 pour
  copy ATMP(2:11) to APF(2:11) | une utilisation ultérieure par le post-filtre
  If FERROR = .FALSE., do the next line
  continue block 50,        |
  order 11 to 50           | continuer pour terminer le bloc 50
                            | output pred coef= ATMP() with NLSATMP
                            | output ill-condition flag = ILLCOND

```

```

If ICOUNT = 1, do the next 7 lines
  GTMP(1)=GSTATE(4) | mettre à jour le tableau GTMP() en une
                    | seule fois
  GTMP(2)=GSTATE(3)
  GTMP(3)=GSTATE(2)
  GTMP(4)=GSTATE(1)
  do block 43FE | output ill-condition flag = ILLCONDG
  If FERROR = .FALSE., do the next line
    do block 44 | output pred. coeff. = GPTMP()
                | output ill-condition flag = ILLCONDG
                | fin du traitement unitaire par trame
Go to VEC_LOOP

```

## I.5.2 Bloc 31SF – Positionner les fanions et le facteur de normalisation pour l'effacement de trame

Ce nouveau bloc est appelé au début de l'effacement de trame, puis toutes les 10 ms durant un effacement. Il positionne le fanion voisé/non voisé au début d'un effacement en se basant sur la valeur de PTAP. Si la dernière bonne trame était voisée, il sauvegarde alors la période de hauteur du son KP dans le FEDELAY pour une extrapolation périodique. Si cette trame était non voisée, il calcule la somme des amplitudes des 40 derniers échantillons du tableau ETPAST() dans la dernière bonne trame puis divise le résultat par 8. Ces valeurs sont calculées une fois au début d'un effacement et sont utilisées ensuite pour la totalité de l'effacement. Le bloc 31SF est appelé à nouveau (toutes les 10 ms) si l'effacement dure plus de 10 ms et le facteur de normalisation de l'extrapolation d'excitation FESCALE est atténué en fonction de la durée de l'effacement et du caractère voisé ou non de ce dernier. Lorsque la parole est voisée et que l'effacement dure 60 ms ou plus, le facteur FESCALE est alors positionné à 0 pour supprimer les "bips" synthétiques. Le facteur FESCALE est positionné à 0 au bout de 70 ms si l'effacement est non voisé. Le code de traitement en virgule flottante est donné en premier.

```

VOICEDFEGAIN(0:4) = .8, .8, .6, .4, .2
UNVOICEDFEGAIN(0:5) = 1., 1., .8, .6, .4, .2

N10MSEC = FECOUNT >> 2
If N10MSEC == 0, do the following indented lines
  If PTAP > VTH, do the next 2 indented lines
    FEDELAY = KP
    VOICED = .TRUE.
  Otherwise, do the following indented lines
    VOICED = .FALSE.
    AVMAG = 0.
    For I = -39, -38, ..., 0, do the next line
      AVMAG = AVMAG + |ETPAST(I)| | somme des amplitudes du tableau ETPAST()
      AVMAG = 0.125*AVMAG | division par 8
  If VOICED = .TRUE., do the following indented lines
    If N10MSEC < 5, do FESCALE = VOICEDFEGAIN(N10MSEC)
    Otherwise, do FESCALE = 0.
  Otherwise, do the following indented lines | non voisé
    If N10MSEC < 6, do FESCALE = AVMAG * UNVOICEDFEGAIN(N10MSEC)
    Otherwise, do FESCALE = 0.

```

Le code de traitement en virgule fixe est le suivant:

```

                    | Les constantes ci-dessous sont en
                    | format Q15
VOICEDFEGAIN(0:4) = 26214, 26214, 19661, 13107, 6554
UNVOICEDFEGAIN(0:5) = 32767, 32767, 26214, 19661, 13107, 6554

N10MSEC = FECOUNT >> 2
If N10MSEC == 0, do the following indented lines
  If PTAP > VTH, do the next 2 indented lines
    FEDELAY = KP
    VOICED = .TRUE.
  Otherwise, do the following indented lines
    VOICED = .FALSE.
    AA0 = 0

```

```

For I = -39, -38, ..., 0, do the next line
  AA0 = AA0 + |ETPAST(I)|           | somme de 40 amplitudes du tableau ETPAST
Call VSCALE(AA0,1,1,30,AA0,NLS)   | normaliser le numérateur
AVMAG = RND(AA0)
NLSAVMAG = NLS - 16 + 3           | "+3" indique "division par 8"
If VOICED = .TRUE., do the following indented lines
  If N10MSEC < 5, do FESCALE = VOICEDFEGAIN(N10MSEC)
  Otherwise, do FESCALE = 0
Otherwise, do the following indented lines | non voisé
  If N10MSEC < 6, do the following indented lines
    FESCALE = UNVOICEDFEGAIN(N10MSEC) | facteur de normalisation en format Q15
    AA0 = FESCALE * AVMAG              | normalisation de AVMAG par FESCALE
    Call VSCALE(AA0,1,1,30,AA0,NLSFESCALE) | normaliser FESCALE
    FESCALE = RND(AA0)                 | sauvegarder le résultat dans FESCALE
    NLSFESCALE = NLSFESCALE + NLSAVMAG - 1 | facteur NLS pour FESCALE
                                           | -1 provient de (Q15 - 16)
  Otherwise, do FESCALE = 0

```

### I.5.3 Bloc 31FE – Extrapolation du signal d'excitation

Ce nouveau bloc est utilisé uniquement durant les trames effacées. Il extrapole le signal d'excitation de manière différente selon que la dernière bonne trame est considérée comme voisée ou non. Dans le cas d'une parole voisée, le signal est extrapolé périodiquement avec une valeur décroissante. Le facteur de normalisation FESCALE part de la valeur 0,8 et décroît toutes les 10 ms pendant la durée d'un effacement dans le bloc 31SF.

On utilise, pour tout autre signal (non voisé, silence, etc.), un segment pris au hasard parmi les 140 échantillons précédents. Ce segment aléatoire est normalisé de manière à posséder la même amplitude moyenne que les 40 derniers échantillons. Il est nécessaire, pour le calcul des amplitudes moyennes du segment aléatoire à 5 échantillons et des derniers 40 échantillons, de diviser les sommes des amplitudes respectivement par 5 et par 40. La seule valeur utile est toutefois le rapport entre les deux amplitudes moyennes. Il en résulte que l'amplitude MAG est calculée comme somme des amplitudes du segment aléatoire à 5 échantillons et l'amplitude AVMAG comme somme des amplitudes des 40 échantillons précédents divisée par 8. Le rapport des deux fournit le rapport correct recherché. La valeur de AVMAG a été calculée dans le bloc 31SF ci-dessus pendant le premier cycle d'adaptation de la première trame effacée. L'amplitude AVMAG est atténuée dans le bloc 31SF en fonction de la durée de l'effacement et placée dans FESCALE. Il convient de noter qu'il n'est pas nécessaire de disposer d'une série prédéfinie de nombres aléatoires du fait que l'interfonctionnement avec le codeur G.728 n'est pas un problème pendant les effacements de trame. Il n'est pas nécessaire non plus que ces nombres soient effectivement aléatoires. Une succession quelconque d'entiers compris entre 5 et 140 peut être utilisée dans ce cas, tant que cette série n'est pas périodique ou proche d'une telle série. Il en résulte que le présent document ne fournit pas de pseudo-code pour le générateur de nombres aléatoires.

Le code de traitement en virgule flottante est donné en premier.

```

                                           | cas voisé
If VOICED = .TRUE., do the next 2 indented lines
  For I = 1, 2, ..., IDIM, do the next line
    ET(I) = FESCALE * ETPAST(I-FEDELAY) | KP est la période de hauteur du son
                                           | de 82
                                           |
                                           | cas non voisé
If VOICED = .FALSE., do the following indented lines
  set FEDELAY = a random number between 5 and 140
  MAG = 0.
  For I = 1, 2, ..., IDIM, do the next 2 lines
    ET(I) = ETPAST(I-FEDELAY)
    MAG = MAG + |ET(I)|                 | somme des amplitudes dans l'excitation
                                           | ET()
    If MAG = 0 do MAG = 1.              | éviter la division par zéro
  RATIO = FESCALE / MAG

```



```

For I = 1, 2, ..., IDIM, do the next line
  ET(I) = RATIO*ET(I) | normalisation pour obtenir la même
                       | amplitude

```

Les excitations ET et ETPAST utilisent des représentations différentes pour le pseudo-code en virgule fixe. Comme spécifié dans l'Annexe G/G.728, l'excitation ET est représentée par un bloc en virgule flottante à 15 bits avec un zéro en tête. Nous utilisons d'autre part, dans un but de simplification, un format Q2 fixe pour représenter le tableau long d'excitation ETPAST. Il n'est pas nécessaire d'employer un bloc en virgule flottante pour ETPAST parce que l'erreur numérique de la représentation Q2 fixe est notablement inférieure à la différence entre l'excitation extrapolée et l'excitation effective du codeur. Dans le cas extrêmement rare d'un débordement de l'excitation ETPAST, la valeur de cette excitation est écrêtée à un niveau de saturation de +32767 ou 32768 pour éviter un "rebouclage". Le pseudo-code suivant décrit d'abord le vecteur d'excitation extrapolée (en format Q2) dans les éléments ETPAST(1) à ETPAST(5), puis convertit les 5 échantillons ETPAST extrapolés en un tableau de blocs ET à 15 bits en virgule flottante.

```

If VOICED = .TRUE., do the next 4 lines | cas voisé
  TEMP = FESCALE
  NLSTEMP = 15 | FESCALE est en format Q15
  For I = 1, 2, ..., IDIM, do the next line
    ETPAST(I) = ETPAST(I-FEDELAY) | FEDELAY est la période de la hauteur du son

If VOICED = .FALSE., do the next 13 lines | cas non voisé
  set FEDELAY = a random number between 5 and 140
  AA1 = 0
  For I = 1, 2, ..., IDIM, do the next 2 lines
    ETPAST(I) = ETPAST(I-FEDELAY)
    AA1 = AA1 + |ETPAST(I)| | somme des amplitudes ETPAST(1:5)
  If AA1 = 0 or FESCALE = 0, do the next 2 lines
    TEMP = 0
    NLSTEMP = 15
  Otherwise, do the next 4 lines
    Call VSCALE(AA1,1,1,30,AA1,NLSDEN) | normaliser le dénominateur
    DEN = RND(AA1)
    NLSDEN = NLSDEN - 16
    Call DIVIDE(FESCALE,NLSFESCALE,DEN,NLSDEN,TEMP,NLSTEMP)

For I = 1, 2, ..., IDIM, do the next 5 lines
  AA0 = TEMP * ETPAST(I) | AA0 en format Q(2 + NLSTEMP)
  AA0 = AA0 >> NLSTEMP | décaler AA0 pour obtenir le format Q2
  if AA0 > 32767, set AA0 = 32767 | écrêter si nécessaire
  if AA0 < -32768, set AA0 = -32768
  ETPAST(I) = AA0 | ETPAST normalisé au format Q2
  Call VSCALE(ETPAST,IDIM,IDIM,13,ET,NLS) | convertir en excitation ET BFT à 15 bits
  | BFL
  NLSET = NLS + 2 | mise à jour de NLSET

```

#### I.5.4 Bloc 31E – Mise à jour du signal d'excitation

Ce bloc met à jour le tampon de signal d'excitation. Ce bloc doit être exécuté indépendamment du masquage éventuel d'un effacement de trame en cours de manière à garantir que le tampon sera prêt en cas de besoin. Le code de traitement en virgule flottante est donné en premier.

```

For I = -KPMAX+1, -KPMAX+2, ..., -IDIM, do the next line
  ETPAST(I) = ETPAST(I+IDIM) | décaler l'excitation ETPAST de
                              | IDIM échantillons

For I = 1, 2, ..., IDIM, do the next line
  ETPAST(I-IDIM) = ET(I) | copier ET dans ETPAST

```

L'excitation ETPAST est stockée dans le format Q2 pour le pseudo-code en virgule fixe et l'excitation ET est stockée dans un bloc en virgule flottante avec NLSET en format Q. Une conversion de format est nécessaire de ce fait lors de la copie de ET vers ETPAST. Le code de traitement en virgule fixe est donné ci-après.

```

For I = -KPMAX+1, -KPMAX+2, ..., -IDIM, do the next line
  ETPAST(I) = ETPAST(I+IDIM) | décaler ETPAST en format Q2
                                | de 5 échantillons
NRS = NLSET - 2                | nombre de décalages à droite pour
                                | le format Q2
For I = 1, 2, ..., IDIM, do the next 6 lines
  AAO = ET(I)                  | obtenir la mantisse à 15 bits de ET
  if NRS > 0, AAO >> NRS        | effectuer le décalage
  if NRS < 0, AAO << -NRS
  if AAO > 32767, set AAO = 32767 | écrêter si nécessaire
  if AAO < -32768, set AAO = -32768
  ETPAST(I-IDIM) = AAO         | ETPAST est maintenant en format Q2

```

### I.5.5 Bloc 43FE – Module de fenêtre hybride

Le présent sous-paragraphe contient le pseudo-code du traitement en virgule flottante et fixe pour le bloc 43FE. Le code a été modifié pour tenir compte du masquage d'erreur de trame et doit remplacer le bloc 43 d'origine décrit au 5.7/G.728. La seule modification dans le code en virgule flottante concerne une ou plusieurs lignes de pseudo-code qui évitent le calcul final de la fonction d'auto-corrélation dans le cas d'un effacement de trame. Le pseudo-code en virgule flottante est donné en premier.

```

N1=LPCLG+NUPDATE                | calculer certaines constantes
N2=LPCLG+NONRLG                 | (pouvant être déterminées à l'avance
                                | et mémorisées)
N3=LPCLG+NUPDATE+NONRLG

For N=1,2,...,N2, do the next line
  SBLG(N)=SBLG(N+NUPDATE)       | décaler l'ancien tampon de signal;
For N=1,2,...,NUPDATE, do the next line
  SBLG(N2+N)=GTMP(N)           | décaler dans le nouveau signal;
                                | SBLG(N3) est l'échantillon le plus
                                | récent
K=1
For N=N3,N3-1,...,3,2,1, do the next 2 lines
  WS(N)=SBLG(N)*WNRLG(K)        | multiplier la fonction de fenêtre
  K=K+1

For I=1,2,...,LPCLG+1, do the next 4 lines
  TMP=0.
  For N=LPCLG+1,LPCLG+2,...,N1, do the next line
    TMP=TMP+WS(N)*WS(N+1-I)
  REXPLG(I)=(3/4)*REXPLG(I)+TMP | mise à jour composant récursif

If FERROR = .TRUE., skip all the lines below
  For I=1,2,...,LPCLG+1, do the next 3 lines
    R(I)=REXPLG(I)
    For N=N1+1,N1+2,...,N3, do the next line
      R(I)=R(I)+WS(N)*WS(N+1-I) | ajouter le composant non récursif

  R(1)=R(1)*WNCF                | correction de bruit blanc

```

Nous donnons ci-dessous la version en virgule fixe pour le même module. La seule modification porte sur le module HWMCOREFE qui remplace le module HWMCORE. Le pseudo-code en virgule fixe remplace le bloc 43 d'origine en virgule du G.3.14/G.728. Le sous-programme HWMCOREFE se trouve dans le bloc 49FE du I.5.7.

```

N1=LPCLG+NUPDATE (=10+4)       | calculer certaines constantes
N2=LPCLG+NONRLG (=10+20)      | (pouvant être déterminées à l'avance et
                                | mémorisées)
N3=LPCLG+NUPDATE+NONRLG (=10+4+20)

For N=1,2,...,N2, do the next line
  SBLG(N)=SBLG(N+NUPDATE)       | décaler l'ancien tampon de signal
For N=1,2,...,NUPDATE, do the next line
  SBLG(N2+N)=GTMP(N)           | SBLG(N3) est l'échantillon le plus
                                | récent
                                | tous les SBLG sont en format Q9 et
                                | sont représentés

```

```

Call FINDNLS(SBLG,N3,N3,14,NLS)          | avec une précision de 16 bits
                                           | déterminer le nombre de décalages
                                           | à gauche
NLSTMP=NLS-1                             | nécessaire dans la boucle suivante
                                           | afin d'avoir par la suite deux bits
                                           | libres en tête

K=1
For N=34,33,...,1, do the next 5 lines
P=SBLG(N)*WNRLG(K)                       | WNRLG est en format Q15
If NLSTMP = -1, set AA0=P >> 1
If NLSTMP > -1, set AA0=P << NLSTMP
WS(N)=RND(AA0)                           | WS(N) a une longueur inférieure ou
                                           | égale à 14 bits

K=K+1

NLSATTLG = 14
Call HWCOREFE(LPCLG,N1,N3,NLSATTLG,WS,NLSTMP,REXPLG,NLSREXPLG,R,ILLCOND)

```

### I.5.6 Bloc 47AF – Limiteur de gain logarithmique après un effacement de trame

Ce nouveau bloc remplace le bloc 47 du 5.7/G.728. Le pseudo-code en virgule flottante est le suivant.

```

If GAIN < 0., set GAIN = 0.               | Correspond à un gain linéaire unité.
If GAIN > 60., set GAIN = 60.            | Correspond à un gain linéaire de 1000.
TMP = GAIN - OGAINDB
If AFTERFE > 0 and TMP > FEGAINMAX, do the next line
GAIN = OGAINDB + FEGAINMAX

```

### I.5.7 Bloc 49FE – Module de fenêtre hybride pour le filtre de synthèse

Ce bloc est le module de fenêtre hybride pour le filtre de synthèse tel qu'il a été modifié pour le cas d'effacements de trame. Ce bloc remplace l'ancien bloc 49 décrit au 5.6/G.728. Le pseudo-code en virgule flottante est donné en premier.

```

N1=LPC+NFRSZ                             | calculer certaines constantes
N2=LPC+NONR                               | (pouvant être déterminées à l'avance
                                           | et mémorisées)

N3=LPC+NFRSZ+NONR

For N=1,2,...,N2, do the next line
SB(N)=SB(N+NFRSZ)                         | décaler l'ancien tampon de signal;
For N=1,2,...,NFRSZ, do the next line
SB(N2+N)=STTMP(N)                         | décaler dans le nouveau signal;
                                           | SB(N3) est l'échantillon le plus
                                           | récent

K=1
For N=N3,N3-1,...,3,2,1, do the next 2 lines
WS(N)=SB(N)*WNR(K)                        | multiplier la fonction de fenêtre
K=K+1

For I=1,2,...,LPC+1, do the next 4 lines
TMP=0.
For N=LPC+1,LPC+2,...,N1, do the next line
TMP=TMP+WS(N)*WS(N+1-I)
REXP(I)=(3/4)*REXP(I)+TMP                 | mise à jour du composant récursif

If FERROR = .TRUE., do the next 4 lines
For I=1,2,...,11, do the next 3 lines
RTMP(I)=REXP(I)
For N=N1+1,N1+2,...,N3, do the next line
RTMP(I)=RTMP(I)+WS(N)*WS(N+1-I)         | ajouter le composant non récursif

If FERROR = .FALSE., do the next 4 lines
For I=1,2,...,LPC+1, do the next 3 lines
RTMP(I)=REXP(I)
For N=N1+1,N1+2,...,N3, do the next line
RTMP(I)=RTMP(I)+WS(N)*WS(N+1-I)         | ajouter le composant non récursif

RTMP(1)=RTMP(1)*WNCF                     | correction de bruit blanc

```

Le pseudo-code en virgule fixe du module de fenêtre hybride pour le filtre de synthèse (Bloc 49FE) est modifié d'une manière similaire. La modification se trouve effectivement dans le module HWMCOREFE. Ce pseudo-code en virgule fixe remplace l'ancien bloc 49 en virgule fixe décrit au G.3.17/G.728.

```

N1=LPC+NFRSZ(=70)           | calculer certaines constantes
N2=LPC+NONR(=85)           | (pouvant être déterminées à l'avance et
mémorisées)
N3=LPC+NFRSZ+NONR(=105)
N4=N3/IDIM(=21)
N5=NFRSZ/IDIM(=4)
N6=N4-N5(=17)

For N=1,2,...,N2, do the next line
  SB(N)=SB(N+NFRSZ)         | décaler l'ancienne partie du tampon SB
For N=1,2,...,N6, do the next line
  NLSSB(N)=NLSSB(N+N5)     | décaler l'ancien NLSSB
For N=1,2,...,NFRSZ, do the next line
  SB(N2+N)=STTMP(N)        | décaler dans la nouvelle partie de SB
For N=1,2,...,N5, do the next line
  NLSSB(N6+N)=NLSSTMP(N)   | décaler dans le nouveau NLSSB

  | Déterminer le NLSSB minimal - ceci détermine NLSTMP
NLSTMP=Min{NLSSB(1),NLSSB(2),...,NLSSB(N4)}

K=1                           | multiplier SB par la
N=N3                           | fonction de fenêtre hybride
For J=1,2,...,N4, do the next 8 lines
  NRSH=NLSSB(J)-NLSTMP-1     | -1 pour compenser le format Q15
  For M=1,2,...,IDIM, do the next 5 lines
    P=SB(K)*WNR(N)           | WNR est en format Q15
    If NRSH = -1, set AA0=P << 1
    If NRSH > -1, set AA0=P >> NRSH
    WS(K)=RND(AA0)           | arrondir le mot supérieur et
                              | mémoriser dans WS

    N=N-1
    K=K+1

NLSATT50 = 14
Call HWMCOREFE(LPC,N1,N3,NLSATT50,WS,NLSTMP,REXP,NLSREXP,RTMP,ILLCOND)

```

Le code en virgule fixe du module HWMCOREFE a été modifié de sorte que seules les valeurs de RTMP(1), ..., RTMP(11) sont calculées dans le cas d'un effacement de trame, les valeurs de REXP(1), ..., REXP(51) étant toutefois mises à jour.

```

SUBROUTINE HWMCOREFE(LPO,N1,N3,NLSATT,WS,NLSTMP,
  RREC,NLSRREC,R,ILLCOND)
NLSAA0=2*NLSTMP
AA0=0                           | calculer la partie récursive de RREC(1)
For N=LPO+1,...,N1, do the next 2 lines
  P=WS(N)*WS(N)             | WS a deux bits libres en tête
  AA0=AA0+P                 | AA0 aura 5 bits libres en tête pour
                              | le calcul de l'énergie

                              | Cas 1: NLSRREC > NLSAA0
If NLSRREC > NLSAA0, do the next 22 lines
  AA0=AA0 >> 1
  IR=NLSRREC-NLSAA0+1
  AA1=RREC(1) << NLSATT     | ceci peut se faire par multiplication
  AA1=-AA1+(RREC(1) << 16) | normalisation de RREC par le facteur
                              | d'atténuation
  AA1=AA1 >> IR             | aligner AA0 et AA1
  AA0=AA0+AA1
  Call VSCALE(AA0,1,1,30,AA0,NLSRE) | Trouver NLS pour RREC
  RREC(1)=RND(AA0)         | les 16 bits supérieurs de AA1
                              | sont sauvegardés

NLSRREC=NLSAA0-1+NLSRE
For I=1,2,...,LPO, do the next 11 lines

```

```

AA0=0 | calculer la partie réursive
      | de RREC(I+1)
For N=LPO+1,...,N1, do the next 2 lines
  P=WS(N)*WS(N-I)
  AA0=AA0+P
  AA0=AA0 >> 1
  AA1=RREC(I+1) << NLSATT | normalisation de RREC par
      | un facteur 3/4 ou 1/2
  AA1=-AA1+(RREC(I+1) << 16)
  AA1=AA1 >> IR
  AA0=AA0+AA1
  AA0=AA0 << NLSRE
  RREC(I+1)=RND(AA0) | les 16 bits supérieurs de AA0
      | sont sauvegardés
Go to FIN_RECUR

If NLSRREC = NLSAA0, do the next 21 lines | Cas 2: NLSRREC = NLSAA0
  AA1=RREC(1) << NLSATT | normalisation de RREC par
      | un facteur 3/4 ou 1/2
  AA1=-AA1+(RREC(1) << 16)
  AA0=AA0 >> 1
  AA1=AA1 >> 1
  AA0=AA0+AA1
  Call VSCALE(AA0,1,1,30,AA0,NLSRE) | trouver NLS pour RREC
  RREC(1)=RND(AA0) | les 16 bits supérieurs de AA1
      | sont sauvegardés
  NLSRREC=NLSRREC-1+NLSRE
  For I=1,2,...,LPO, do the next 11 lines
    AA0=0 | calculer la partie réursive
          | de RREC(I+1)
    For N=LPO+1,...,N1, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> 1
      AA1=RREC(I+1) << NLSATT | normalisation de RREC par
          | un facteur 3/4 ou 1/2
      AA1=-AA1+(RREC(I+1) << 16)
      AA1=AA1 >> 1
      AA0=AA0+AA1
      AA0=AA0 << NLSRE
      RREC(I+1)=RND(AA0) | les 16 bits supérieurs de AA0
          | sont sauvegardés
    Go to FIN_RECUR

If NLSRREC < NLSAA0, do the next 21 lines | Cas 3: NLSRREC < NLSAA0
  IR=NLSAA0-NLSRREC+1
  AA0=AA0 >> IR
  AA1=RREC(1) << NLSATT | normalisation de RREC par
      | un facteur 3/4 ou 1/2
  AA1=-AA1+(RREC(1) << 16)
  AA1=AA1 >> 1
  AA0=AA0+AA1
  Call VSCALE(AA0,1,1,30,AA0,NLSRE)
  RREC(1)=RND(AA0) | les 16 bits supérieurs de AA1
      | sont sauvegardés
  NLSRREC=NLSRREC-1+NLSRE
  For I=1,2,...,LPO, do the next 11 lines
    AA0=0 | calculer la partie réursive
          | de RREC(I+1)
    For N=LPO+1,...,N1, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> IR
      AA1=RREC(I+1) << NLSATT | normalisation de RREC par
          | un facteur 3/4 ou 1/2
      AA1=-AA1+(RREC(I+1) << 16)
      AA1=AA1 >> 1
      AA0=AA0+AA1
      AA0=AA0 << NLSRE

```

```

RREC(I+1)=RND(AA0) | les 16 bits supérieurs de AA0
                    | sont sauvegardés

FIN_RECUR:         | le composant récursif a été calculé
                    | une fois ce point atteint

AA0=0              | calculer la partie non récursive de R(1)
For N=N1+1,...,N3, do the next 2 lines
  P=WS(N)*WS(N)
  AA0=AA0+P

LPFE = LPO
If FERROR .=. TRUE, set LPFE = 10

If NLSRREC > NLSAA0, do the next 21 lines | Cas 1: NLSRREC > NLSAA0
  IR=NLSRREC-NLSAA0+1
  AA1=RREC(1) << 16
  AA1=AA1 >> IR
  AA0=AA0 >> 1
  AA1=AA0+AA1
  AA0=AA1 >> 8 | appliquer le facteur de correction
                  | de bruit blanc

  AA1=AA1+AA0
  Call VSCALE(AA1,1,1,30,AA1,NLSRR)
  R(1)=RND(AA1) | les 16 bits supérieurs de AA1
                  | sont sauvegardés

  For I=1,2,...,LPFE, do the next 10 lines | calculer la partie non récursive
  AA0=0 | de R(I+1)
    For N=N1+1,...,N3, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> 1
      AA1=RREC(I+1) << 16
      AA1=AA1 >> IR
      AA1=AA0+AA1
      AA1=AA1 << NLSRR
      R(I+1)=RND(AA1) | sauvegarder les 16 bits supérieurs
    Go to END

If NLSRREC = NLSAA0, do the next 18 lines | Cas 2: NLSRREC = NLSAA0
  AA0=AA0 >> 1
  AA1=RREC(1) << 15 | ceci peut se faire par multiplication
  AA1=AA0+AA1
  AA0=AA1 >> 8 | appliquer le facteur de correction
                  | de bruit blanc

  AA1=AA1+AA0
  Call VSCALE(AA1,1,1,30,AA1,NLSRR)
  R(1)=RND(AA1) | les 16 bits supérieurs de AA1
                  | sont sauvegardés

  For I=1,2,...,LPFE, do the next 9 lines | calculer la partie non récursive
  AA0=0 | de R(I+1)
    For N=N1+1,...,N3, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> 1
      AA1=RREC(I+1) << 15
      AA1=AA0+AA1
      AA1=AA1 << NLSRR
      R(I+1)=RND(AA1) | sauvegarder les 16 bits supérieurs
    Go to END

If NLSRREC < NLSAA0, do the next 18 lines | Cas 3: NLSRREC < NLSAA0
  IR=NLSAA0-NLSRREC+1
  AA0=AA0 >> IR
  AA1=RREC(1) << 15 | ceci peut se faire par multiplication
  AA1=AA0+AA1

```

```

AA0=AA1 >> 8 | appliquer le facteur de correction
                | de bruit blanc

AA1=AA1+AA0
Call VSCALE(AA1,1,1,30,AA1,NLSRR)
R(1)=RND(AA1) | les 16 bits supérieurs de AA1
                | sont sauvegardés

For I=1,2,...,LPFE, do the next 9 lines
  AA0=0 | calculer la partie non récursive
        | de R(I+1)

  For N=N1+1,...,N3, do the next 2 lines
    P=WS(N)*WS(N-I)
    AA0=AA0+P
    AA0=AA0 >> IR
    AA1=RREC(I+1) << 15
    AA1=AA0+AA1
    AA1=AA1 << NLSRR
    R(I+1)=RND(AA1) | sauvegarder les 16 bits supérieurs

END: | un dernier effort, vérification
      | de condition défavorable

ILLCOND=.FALSE.
If AA1 = 0, set ILLCOND=.TRUE. | AA1 contient encore R(LPFE + 1)
                                | avec 32 bits

```

### I.5.8 Bloc 51FE – Module d'expansion de largeur de bande pour des effacements de trame

Ce paragraphe donne le pseudo-code en virgule flottante pour le bloc 51FE qui contient le module d'expansion de largeur de bande du filtre de synthèse LPC dans des situations d'effacement de trame. Ce module est comparable au bloc 51 décrit au 5.6/G.728, mais utilise des coefficients d'élargissement de largeur de bande différents. Une autre différence est que les valeurs d'entrée sont contenues dans le tableau A(I).

```

If FECOUNT = 1 and ILLCOND = .FALSE., do the next 2 lines
  For I=2,3,...,LPC+1, do the next line
    A(I)=FACVFE(I) * ATMP(I) | coefficient de normalisation
  Otherwise, do the next 2 lines
    For I=2,3,...,LPC+1, do the next line
      A(I)=FACVFE(I) * A(I) | coefficient de normalisation

```

Les tableaux pour FACVFE sont donnés dans le format Q14. Les valeurs finales du tableau A peuvent toujours être représentées dans le format Q14. Le pseudo-code en virgule fixe pour le bloc 51 est le suivant.

```

If FECOUNT = 1 and ILLCOND = .FALSE., do the next 4 lines
  For I=2,3,4,...,LPC+1, do the next 3 lines
    AA0 = FACVFE(I) * ATMP(I) | AA0 est au format Q28
    AA0 = AA0 << 2 | mettre AA0 au format Q30
    A(I) = RND(AA0) | arrondir vers le mot supérieur pour
                    | obtenir un format Q14

  Otherwise, do the next 4 lines
    For I=2,3,4,...,LPC+1, do the next 3 lines
      AA0 = FACVFE(I) * A(I) | AA0 est au format Q28
      AA0 = AA0 << 2 | mettre AA0 au format Q30
      A(I) = RND(AA0) | arrondir vers le mot supérieur
                      | pour obtenir un format Q14

```

### I.5.9 Bloc 97FE – Mise à jour de GSTATE pendant les effacements de trame

Nous donnons tout d'abord la version en virgule flottante. Ce code est le même que dans les blocs 67, 39, 40 et 42 décrit au 5.7/G.728.

```

ETRMS = 0.
For K = 1,...,IDIM, do the next line | calcul de l'énergie
  ETRMS = ETRMS + ET(K)*ET(K) | d'excitation
ETRMS = ETRMS * DIMINV
If ETRMS < 1., set ETRMS = 1. | vérifier la limite inférieure
ETRMS = 10.* log10 (ETRMS) | calculer la valeur en dB

GSTATE(1) = ETRMS - GOFF | soustraire le décalage de gain en dB

```

La version en virgule fixe de ce nouveau bloc effectue des opérations équivalentes. La fonction  $\log_{10}$  avait été supprimée dans l'ancienne spécification en virgule fixe de la Recommandation G.728. Il est nécessaire de la réintroduire ici pour effectuer les calculs qui suivent. La précision maximale n'est pas nécessaire pour le calcul de la fonction  $\log_{10}$  parce que nous ne recherchons pas de correspondance exacte avec la valeur fournie par la spécification en virgule flottante. Le code en virgule fixe est donné ci-dessous. Comme une compatibilité exacte au niveau du bit n'est pas requise, on ne donne pas de description exacte à ce niveau pour l'implémentation en virgule fixe de la fonction LIN2DB. Cette fonction convertit une représentation scalaire en virgule flottante de ETRMS en une valeur exprimée en dB dans le format Q9. Il est possible d'utiliser toute implémentation en virgule fixe de la fonction LIN2DB qui fournit une précision convenable. Le code en virgule fixe qui suit donne un exemple d'une telle implémentation.

```

AA0 = 0
For K = 1,...,IDIM, do the next line | calcul de l'énergie d'excitation
  AA0 = AA0 + ET(K)*ET(K)          | ET est en format Q2, de sorte que AA0
                                  | est en format Q4

Call VSCALE(AA0,1,1,30,AA0,NLS)   | normalisation de AA0
ETRMS = AA0 >> 16                 | prendre le mot de fort poids de AA0
NLSETRMS = (4 + NLS) - 16         | facteur NLS de ETRMS
AA0 = ETRMS * DIMINV              | DIMINV = 0,2 en format Q16
Call VSCALE(AA0,1,1,30,AA0,NLS)   | normalisation de AA0
ETRMS = AA0 >> 16                 | prendre le mot de fort poids de AA0
NLSETRMS = NLSETRMS + NLS        | facteur NLS de ETRMS
If NLSETRMS > 14, do the next 2 lines | si ETRMS est en format Q15
                                  | ou inférieur (<1)
  ETRMS = 16384                   | positionner ETRMS à 1 en format Q16
  NLSETRMS = 14

Call LIN2DB(ETRMS,NLSETRMS,GSTATE(1)) | la conversion linéaire en dB
                                       | fomat Q9 fixe)
                                       | du décalage du gain est faite
                                       | dans la fonction LIN2DB

```

Les constantes nécessaires pour le code en virgule fixe de LIN2DB sont indiquées ci-dessous.

```

c(0)    = -3 400 = -0,4150374993 en format Q13
c(1)    = 15 758 = 1,9235933879 en format Q13
c(2)    = -10 505 = -1,2823955919 en format Q13
c(3)    = 9 338 = 1,1399071928 en format Q13
c(4)    = -9 338 = -1,1399071928 en format Q13
c(5)    = 9 961 = 1,2159010057 en format Q13
FAC1    = 24 660 = 3,0102999566 en format Q13
GOFF    = -16 384 = -32 en format Q9
THRQTR  = 24 576 = 0,75 en format Q15

```

Le code de la fonction LIN2DB en virgule fixe est donné ci-dessous. Il calcule un développement limité par série de Taylor pour la fonction  $\log_2$  au voisinage du point 0,75. On fait l'hypothèse que l'entrée X possède une mantisse normalisée positive de 16 bits avec NLSX comme format Q. Le code effectue la normalisation nécessaire pour obtenir la valeur en dB et soustrait le décalage de gain GOFF du nombre résultant à 32 bits avant la conversion vers la sortie Q19 fixe au format Q.

```

LIN2DB(X, NLSX, RET)
AA0 = x | normalisé 0,5 <= x <= 1, Q15
AA0 = AA0 - THRQTR | soustraire 0,75, maintenant -0,25 <= x 0,25, Q15
AA0 = AA0 << 1 | convertir en Q16
AA1 = 0
For I=5,4,...,1 do the next 3 lines
AA1 = AA1 + C(I) | Q13
P = AA1 * AA0 | Q13 * Q16 = Q29
AA1 = P >> 16 | Q29 -> Q13
AA1 = AA1 + C(0) | Q13
AA1 = AA1 >> 3 | convertir vers Q10
AA0 = 15 - NLSX | exposant en Q0

```



```

AA0 = AA0 << 10      | convertir l'exposant en Q10
AA1 = AA0 + AA1      | Ajouter l'exposant au logarithme de la
                     | mantisse, AA1 est maintenant log2(x)
AA1 = AA1 * FAC1     | Q10 * Q13 = Q23, convertir à 10*log10(x)
AA1 = AA1 >> 14     | Q23 -> Q9
AA1 = AA1 - GOFF     | Soustraire GOFF
RET = AA1            | retourner la valeur Q9

```

### I.5.10 Bloc 98AF – Limiteur de gain logarithmique après un effacement de trame

Ce nouveau bloc remplace le bloc 98 décrit au G.3.16/G.728. Le pseudo-code en virgule fixe pour le bloc 98AF est décrit ci-après.

```

If LOGGAIN > 14336, set LOGGAIN=14336
If LOGGAIN < -16384, set LOGGAIN=-16384
TMP = LOGGAIN - OGAINDB
If AFTERFE > 0 and TMP > FEGAINMAX, do the next line
  LOGGAIN = OGAINDB + FEGAINMAX

```

### I.6 Paramètres de code et variables supplémentaires

Le présent paragraphe contient un tableau donnant les nouvelles variables qui ont été ajoutées au décodeur G.728 pour le masquage d'effacement de trame. Le format en virgule fixe est indiqué dans la colonne "format". Dans le cas d'une implémentation en virgule flottante, toutes les variables de bloc en virgule fixe ou en format Q fixe peuvent être représentées en virgule flottante.

**Tableau I.1/G.728 – Paramètres de code supplémentaires nécessaires pour le masquage d'effacement de trame**

Nom	Valeur en virgule flottante	Valeur en virgule fixe	Format Q	Description
AFTERFEMAX	16	16	Q0	Trames de 2,5 ms au maximum pour l'écrêtage du gain
DIMINV	0,2	13 107	Q16	Inverse de la dimension du vecteur
FEGAINMAX	2	1 024	Q9	Accroissement maximal du gain en dB après un effacement de trame
VTH	0,4285714	7 022	Q14	Seuil de voisage pour les effacements de trame
FESCALE	0,8	26 214	Q15	Facteur de normalisation ETPAST pour les effacements de trame

**Tableau I.2/G.728 – Variables supplémentaires de traitement LD-CELP interne nécessaires pour le masquage d'effacement de trame**

Nom	Domaine d'index de tableau	Format en virgule fixe	Description
MAG	1	a)	amplitude moyenne du vecteur d'excitation actuel
ADCOUNT	1	entier	compteur de cycles d'adaptation G.728 (2,5 ms)
AFTERFE	1	entier	compteur d'écrtage de gain après effacement de trame (2,5 ms)
AVMAG	1	SFL	amplitude moyenne des 8 derniers vecteurs d'excitation
ETPAST	-139 à IDIM	Q2	excitation passée (nulle au départ)
ETRMS	1	SFL	énergie du vecteur actuel en dB
FECOUNT	1	entier	longueur de l'effacement de trame actuel (2,5 ms)
FEDELAY	1	entier	hauteur du son (KP) stockée au début de l'effacement de trame
FERROR	1	logique	indicateur d'effacement de trame
FESCALE	1	entier	facteur de normalisation ETPAST pour les effacements de trame
FESIZE	1	entier	nombre de cycles d'adaptation de 2,5 ms dans un effacement de trame
NLSFESCALE	1	entier	facteur NLS pour FESCALE
NLSAVMAG	1	entier	facteur NLS pour AVGMAG
NLSETRMS	1	entier	facteur NLS pour ETRMS
OGAINDB	1	Q9	ancienne prédiction de gain en dB
VOICED	1	logique	fanion voisé/non voisé

a) MAG n'est pas utilisé dans le pseudo-code en virgule fixe.

## APPENDICE I.I

### Valeurs utilisées pour la normalisation de ETPAST pendant les effacements de trame

Les tableaux I.I.1 et I.I.2 donnent les valeurs en virgule fixe et en virgule flottante utilisées pour la normalisation de l'excitation ETPAST pour les signaux voisés (VOICEDFEGAIN) et non voisés (UNVOICEDFEGAIN) pendant des effacements de trames. Le format Q15 est utilisé pour la virgule fixe.

**Tableau I.I.1/G.728 – Valeurs utilisées pour la normalisation de ETPAST avec des signaux voisés (VOICEDFEGAIN)**

Index VOICEDFEGAIN	Valeur en virgule flottante	Valeur Q15 en virgule fixe
0	0,8	26 214
1	0,8	26 214
2	0,6	19 661
3	0,4	13 107
4	0,2	6 554

**Tableau I.I.2/G.728 – Valeurs utilisées pour la normalisation de ETPAST avec des signaux non voisés (UNVOICEDFEGAIN)**

Index UNVOICEDFEGAIN	Valeur en virgule flottante	Valeur Q15 en virgule fixe
0	1,0	32 767
1	1,0	32 767
2	0,8	26 214
3	0,6	19 661
4	0,4	13 107
5	0,2	6 554

**Valeurs utilisées pour l'expansion de largeur de bande du prédicteur LPC pendant les effacements de trame**

Les tableaux suivants donnent les valeurs entières pour le tableau de coefficients d'expansion de gain FACVFE utilisé pour adoucir le prédicteur durant les effacements de trame. Ces valeurs sont en format Q14. Les valeurs équivalentes en virgule flottante s'obtiennent en divisant les valeurs entières par 16 384. Ce tableau doit être lu de la gauche vers la droite et du haut vers le bas en allant de FACVFE(1) à (51).

16384	15892	15416	14953	14505
14069	13647	13238	12841	12456
12082	11719	11368	11027	10696
10375	10064	9762	9469	9185
8910	8642	8383	8131	7888
7651	7421	7199	6983	6773
6570	6373	6182	5996	5816
5642	5473	5309	5149	4995
4845	4700	4559	4422	4289
4161	4036	3915	3797	3683
3573				

## **SERIES DES RECOMMANDATIONS UIT-T**

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
<b>Série G</b>	<b>Systèmes et supports de transmission, systèmes et réseaux numériques</b>
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication