

①9 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

①1 N° de publication : **2 911 227**  
(à n'utiliser que pour les  
commandes de reproduction)

②1 N° d'enregistrement national : **07 00056**

⑤1 Int Cl<sup>8</sup> : *H 03 M 13/35 (2006.01), H 04 L 9/16, G 06 F 21/00, 11/07*

⑫ **DEMANDE DE BREVET D'INVENTION**

**A1**

②2 Date de dépôt : 05.01.07.

③0 Priorité :

④3 Date de mise à la disposition du public de la demande : 11.07.08 Bulletin 08/28.

⑤6 Liste des documents cités dans le rapport de recherche préliminaire : *Ce dernier n'a pas été établi à la date de publication de la demande.*

⑥0 Références à d'autres documents nationaux apparentés :

⑦1 Demandeur(s) : *FRANCE TELECOM Société anonyme — FR.*

⑦2 Inventeur(s) : *KOVESI BALAZS, VIRETTE DAVID et PHILIPPE PIERRICK.*

⑦3 Titulaire(s) :

⑦4 Mandataire(s) : *CABINET PLASSERAUD.*

⑤4 **CODAGE PAR TRANSFORMÉE, UTILISANT DES FENÊTRES DE PONDERATION ET A FAIBLE RETARD.**

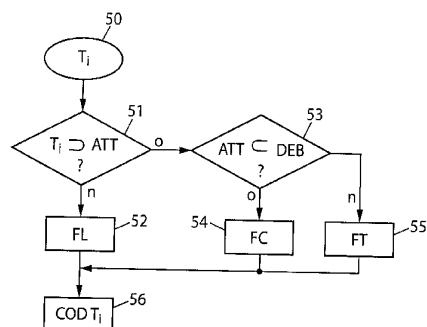
⑤7 La présente invention concerne le codage/décodage, par transformée en utilisant des fenêtres de longueurs différentes, d'un signal audionumérique représenté par une succession de trames. Pour le codage au sens de l'invention :

- on cherche à détecter (51) un événement particulier, tel qu'une attaque, dans une trame courante ( $T_i$ ), et

- au moins si l'événement particulier est détecté en début de trame courante (53), on applique directement une fenêtre courte (54) pour coder (56) la trame courante ( $T_i$ ), sans appliquer de fenêtre de transition.

Le codage est alors de retard réduit par rapport à l'art antérieur.

On applique un traitement ad hoc au décodage pour compenser le passage direct d'une fenêtre longue à une fenêtre courte au codage.



FR 2 911 227 - A1



**Codage par transformée, utilisant des fenêtres de pondération et à faible retard**

La présente invention concerne le codage/décodage de signaux audionumériques.

5 Dans un schéma de codage par transformée, pour une réduction de débit, on cherche habituellement à réduire la précision accordée au codage des échantillons, en s'assurant que l'auditeur ne puisse toutefois percevoir que le moins possible de dégradations.

10 À cette fin, la réduction de précision, effectuée par une opération de quantification, est contrôlée à partir d'un modèle psychoacoustique. Ce modèle, basé sur une connaissance des propriétés de l'oreille humaine, permet de doser le bruit de quantification dans les fréquences auditives les moins perceptibles.

15 Afin d'utiliser les informations issues du modèle psychoacoustique, essentiellement données dans le domaine fréquentiel, il est classique de réaliser une transformation temps/fréquence, la quantification s'effectuant dans ce domaine fréquentiel.

La figure 1 illustre schématiquement la structure d'un codeur par transformée, avec :

- un banc BA de filtres d'analyse FA1, ..., FAn, attaquant le signal d'entrée X,
- 20 - un module de quantification Q, suivi d'un module de codage COD,
- et un banc BS de filtres de synthèse FS1, ..., FSn délivrant le signal codé X'.

25 Pour réduire encore le débit avant transmission, les échantillons fréquentiels quantifiés sont codés, souvent à l'aide d'un codage dit « entropique » (codage sans perte). La quantification est réalisée de façon classique par un quantificateur scalaire, uniforme ou non, ou encore par un quantificateur vectoriel.

Le bruit introduit dans l'étape de quantification est mis en forme par le banc de filtres de synthèse (appelé également « transformation inverse »). La transformation inverse,

liée à la transformée d'analyse, doit donc être choisie afin de bien concentrer le bruit de quantification, fréquemment ou temporellement, afin d'éviter qu'il ne devienne audible.

- 5 La transformation d'analyse doit concentrer au mieux l'énergie du signal afin de permettre un codage aisé des échantillons dans le domaine transformé. En particulier, le gain de codage de la transformée, qui dépend du signal d'entrée, doit être maximisé dans la mesure du possible. On utilise à cet effet une relation du type :

$$SNR = G_{TC} + K \cdot R$$

- 10 où  $K$  est un terme constant qui peut avantageusement valoir 6,02.

Ainsi, le rapport signal à bruit obtenu ( $SNR$ ) est proportionnel au nombre de bits par échantillon sélectionné ( $R$ ) augmenté de la composante  $G_{TC}$  qui représente le gain de codage de la transformée. Plus le gain de codage est important et plus la qualité de reconstruction est bonne.

15

On comprend donc l'importance que revêt la transformation en codage. Elle assure un codage aisé des échantillons, grâce à son aptitude à concentrer à la fois l'énergie du signal (par la partie analyse) et le bruit de quantification (par la partie synthèse).

- 20 Les signaux audio étant notoirement non stationnaires, il convient d'adapter la transformation temps fréquence au cours du temps, en fonction de la nature du signal audio.

On décrit ci-après quelques applications à des techniques de codage usuelles.

25

Dans le cas de transformées modulées, les techniques de codage audio normalisées intègrent des bancs de filtres modulés en cosinus qui permettent de mettre en œuvre ces techniques de codage à l'aide d'algorithmes rapides, basés sur des transformées en cosinus ou des transformées de Fourier rapides.

30

Parmi les transformées de ce type, la transformée la plus communément utilisée (en codage MP3, MPEG-2 et MPEG-4 AAC en particulier) est la transformée MDCT (« *Modified Discrete Cosine Transform* ») dont l'expression est présentée ci-dessous :

$$5 \quad X_k^t = \sum_{n=0}^{2M-1} x_{n+tM} p_k(n) \quad 0 \leq k < M$$

avec les notations suivantes :

- $M$  représente la taille de la transformée.
- $x_{n+tM}$  sont les échantillons du son numérisé à une période  $\frac{1}{F_c}$  (inverse de la fréquence d'échantillonnage) à l'instant  $n+tM$ ,
- 10 •  $t$  est l'indice de trame.
- $X_k^t$  sont les échantillons dans le domaine transformé pour la trame  $t$ ,
- $p_k(n) = \sqrt{\frac{2}{M}} h(n) \cos\left[\frac{\pi}{4M}(2n+1+M)(2k+1)\right]$  est une fonction de base de la transformée dont  $h(n)$  est appelé filtre prototype de taille  $2M$ .

- 15 Pour restituer les échantillons temporels initiaux, la transformation inverse suivante est appliquée afin de reconstituer les échantillons  $0 \leq n \leq M-1$ :

$$\hat{x}_{n+tM} = \sum_{k=0}^{M-1} \left[ X_k^{t+1} p_k(n) + X_k^t p_k(n+M) \right]$$

En référence à la figure 1a, la reconstruction s'effectue comme suit :

- transformée DCT inverse (notée ci-après  $DCT^{-1}$ ) des échantillons  $X_k^t$  donnant
- 20 2M échantillons,
- transformée DCT inverse des échantillons  $X_k^{t+1}$  donnant 2M échantillons, les M premiers échantillons ayant un support temporel identique aux M derniers échantillons de la trame précédente,

- pondération par la fenêtre de synthèse  $h(M+n)$  pour la deuxième moitié de la trame  $T_i$  ( $M$  derniers échantillons), et par la fenêtre de synthèse  $h(n)$  pour la première moitié de la trame suivante  $T_{i+1}$  ( $M$  premiers échantillons), et
- additions des composantes fenêtrées sur le support commun.

5

Afin d'assurer la reconstruction exacte (dite « parfaite ») du signal (selon la condition  $\hat{x}_{n+1M} = x_{n+1M}$ ), il convient de choisir une fenêtre prototype  $h(n)$  satisfaisant quelques contraintes.

Typiquement, les relations suivantes sont satisfaisantes afin de permettre une reconstruction parfaite :

10

$$\begin{cases} h(2M-1-n) = h(n) \\ h^2(n) + h^2(n+M) = 1 \end{cases} \quad (1)$$

les fenêtres étant à symétrie paire par rapport à un échantillon central.

Il est relativement simple de satisfaire ces deux contraintes simples et, à cet effet, un filtre prototype classique est constitué d'une fenêtre sinusoïdale qui s'écrit comme suit :

15

$$h(n) = \sin \left[ \frac{\pi}{2M} (n+0.5) \right]$$

Bien entendu, d'autres formes de filtre prototype existent, telles que les fenêtres définies dans la norme MPEG-4 sous le nom de « Kaiser Bessel Dérivées » (ou KBD), ou encore les fenêtres à faible recouvrement (pour « Low Overlap »).

20

Un exemple de traitement par une transformée MDCT, avec fenêtres longues, est donné sur la figure 1a. Sur cette figure :

- les flèches en traits pointillés illustrent une soustraction,
- les flèches en traits pleins illustrent une addition,

25

- les flèches en traits pointillés mixtes illustrent un traitement DCT au codage et DCT<sup>-1</sup> au décodage DEC, ce terme DCT correspondant au terme en cosinus de la fonction de base présentée ci-avant,
- les échantillons du signal à coder sont dans un flux noté  $x_{in}$ , et on suit l'évolution des traitements de codage/décodage de certains échantillons entourés et référencés **a** et **b** sur la figure 1b et **e** et **f** sur la figure 1c,
- les échantillons  $x_{in}$  sont regroupés par trames, une trame courante est notée  $T_i$ , les trames précédente et suivante étant notées respectivement  $T_{i-1}$  et  $T_{i+1}$ ,
- la référence DEC est propre au traitement réalisé par le décodeur (utilisant des fenêtres de synthèse FS avec addition-recouvrement),
- les fenêtres d'analyse sont notées FA et les fenêtres de synthèse sont notées FS,
- $n$  est la distance entre le milieu de fenêtre et l'échantillon **a**.

La référence calc  $T'_i$  est propre au calcul de la trame codée  $T'_i$  utilisant la fenêtre d'analyse FA et les échantillons respectifs des trames  $T_{i-1}$  et  $T_i$ . Il s'agit simplement ici d'un exemple de convention illustré sur la figure 1a. Il pourrait aussi être décidé, par exemple, d'indexer les trames  $T_i$  et  $T_{i+1}$  pour le calcul d'une trame codée  $T'_i$ . En poursuivant l'exemple de la figure 1a, la référence calc  $T'_{i+1}$  est propre au calcul de la trame codée  $T'_{i+1}$  en utilisant les échantillons respectifs des trames  $T_i$  et  $T_{i+1}$ .

Les termes  $v_1$  et  $v_2$  obtenus avant transformation DCT et transformation inverse DCT<sup>-1</sup> sont obtenus avec des équations du type :

$$v_1 = a * h(M+n) + b * h(2 * M - 1 - n), \text{ et}$$

$$v_2 = b * h(M - 1 - n) - a * h(n)$$

Ainsi, après traitement global DCT/DCT<sup>-1</sup> et fenêtre de synthèse, les termes de reconstruction  $a'$  et  $b'$  s'écrivent:

$$a' = v_1 * h(M+n) - v_2 * h(n) =$$

$$a * h(M+n) * h(M+n) + b * h(2 * M - 1 - n) * h(M+n) - b * h(M - 1 - n) * h(n) + a * h(n) * h(n),$$

et

$$b' = v_1 * h(2 * M - 1 - n) + v_2 * h(M - 1 - n) =$$

$$a^*h(M+n)^* h(2M-n-1) + b^*h(2^*M-1-n-1)^* h(2M-n-1) + b^*h(M-1-n)^* h(M-1-n) - a^*h(n)^* h(M-1-n)$$

et l'on vérifie bien alors que la reconstruction est parfaite ( $a'=a$  et  $b'=b$ ).

(en utilisant les relations (1) et en en déduisant  $h(M-1-n)=h(n+M)$ )

5

Le principe décrit ci-avant d'une transformée MDCT s'étend naturellement à des transformées nommées ELT (« Extended Lapped Transform »), dans lesquelles l'ordre des fonctions de base est plus grand que deux fois la taille de la transformée, avec en particulier :

$$10 \quad X_k^t = \sum_{n=0}^{L-1} x_{n+1M} p_k(n), \text{ où } 0 \leq k < M \text{ et } L = 2KM, K \text{ étant un entier positif supérieur à } 2.$$

Pour la reconstruction, au lieu d'associer deux trames consécutives comme pour une transformée MDCT, la synthèse des échantillons met en œuvre  $K$  trames successives fenêtrées.

15 On indique en outre que la contrainte de symétrie des fenêtres (principe décrit en détail ci-après) peut être relâchée pour une transformée de type ELT. La contrainte d'identité entre les fenêtres d'analyse et de synthèse peut également être relâchée et on parle alors de filtres bi-orthogonaux.

20 Compte tenu de la nécessité d'adapter la transformée au signal à coder, les techniques antérieures autorisent ce qui est appelé une « commutation de fenêtre », c'est-à-dire un changement au cours du temps de la taille de la transformée utilisée.

25 La nécessité de changer de longueur de fenêtre peut se justifier notamment dans le cas suivant. Lorsque le signal à coder, par exemple un signal de parole, comporte un signal transitoire (non stationnaire) caractérisant une attaque forte (par exemple la prononciation d'un son « ta » ou « pa » caractérisant une plosive dans le signal de parole), il convient d'augmenter la résolution temporelle du codage et de là, de diminuer la taille des fenêtres de codage, ce qui nécessite donc de passer d'une fenêtre  
30 longue à une fenêtre courte. Très exactement, dans l'art antérieur connu, on passe dans

ce cas d'une fenêtre longue (figure 2a qui sera décrite plus loin) à une fenêtre de transition (figure 2c décrite plus loin), puis à une série de fenêtres courtes (figure 2b décrite plus loin). Il faut donc anticiper une attaque sur au moins une trame suivante, comme on le verra en détail plus loin, avant de pouvoir décider la longueur de la

5 fenêtre de codage d'une trame courante, et, par conséquent, coder la trame courante.

On décrit ci-après un exemple de changement de longueur de fenêtre au sens de l'art antérieur.

10 Un exemple typique est le changement de taille d'une transformée MDCT de taille  $M$  vers une taille  $M/8$ , comme prévu dans la norme MPEG-AAC.

Afin de conserver la propriété de reconstruction parfaite, l'équation (1) ci-avant doit être remplacée par les formulations suivantes au moment de la transition entre deux

15 tailles :

$$\begin{cases} h^2(n) + h^2(M-n) = 1 & \text{pour } 0 \leq n < M \\ h^2(M+n) + h^2(2M-n) = 1 & \text{sinon} \end{cases}$$

Une relation est donnée en outre pour les filtres prototypes consécutifs de tailles différentes :

20 
$$h_1(M + M/2 - M_s/2 + n) = h_2(M_s - n) \quad 0 \leq n < M_s$$

Il existe donc une symétrie autour de la taille  $M/2$  au moment de la transition.

On a illustré différents types de fenêtre sur les figures 2a à 2e, avec respectivement :

- une fenêtre sinusoïdale (fonction symétrique sinus) de taille  $2M=512$  échantillons pour la figure 2a,
- 25 - une fenêtre sinusoïdale (fonction symétrique sinus) de taille  $2M_s=64$  échantillons pour la figure 2b,
- une fenêtre de transition permettant de passer d'une taille 512 à 64 pour la figure 2c,

- une fenêtre de transition permettant de passer d'une taille de 64 à 512 pour la figure 2d,
- et un exemple de construction effectuée à l'aide des fenêtres de base présentées plus haut, pour la figure 2e.

5

Chaque succession est de « longueur » prédéterminée définissant ce qui est appelé « la longueur de fenêtre ». Ainsi, des échantillons à coder sont combinés, au moins deux à deux, et pondérés, dans la combinaison, par des valeurs respectives de pondération de la fenêtre, comme on l'a vu en référence à la figure 1a.

10

Plus particulièrement, les fenêtres sinusoïdales (figures 2a et 2b) sont symétriques, c'est-à-dire que les valeurs de pondération sont sensiblement égales de part et d'autre d'une valeur centrale, au milieu de la succession de valeurs formant la fenêtre. Une réalisation avantageuse consiste à choisir des fonctions « sinus » pour définir les variations de valeurs de pondération de ces fenêtres. Les autres choix de fenêtre restant

15

possible (par exemple ceux utilisés dans les codeurs MPEG AAC).

20

On relèvera au contraire que les fenêtres de transition (figures 2c et 2d) sont asymétriques et comportent une région « plate » (référence PLA), ce qui signifie que les valeurs de pondération dans ces régions sont maximales et par exemple sont égales à « 1 ». Comme on le verra en référence aux figures 1b et 1c, en utilisant une fenêtre de transition d'une fenêtre longue à une fenêtre courte (figure 2c), deux échantillons (dans l'exemple représenté sur la figure 1b), dont l'échantillon **a**, sont simplement pondérés par un facteur « 1 », tandis que l'échantillon **b** est pondéré par le facteur « 0 »

25

dans le calcul de la trame codée  $T'_i$ , de sorte que les deux échantillons dont l'échantillon **a** sont simplement transmis tels quels (la DCT mise à part) dans la trame codée  $T'_i$ .

30

On décrit ci-après l'utilisation d'une transformée à taille variable dans un système de codage. On décrit aussi les opérations au niveau d'un décodeur qui doit reconstituer les échantillons audio.

Dans les systèmes normalisés, le codeur sélectionne le plus souvent la transformée à utiliser au cours du temps. Ainsi, dans la norme AAC, le codeur transmet deux bits permettant de sélectionner l'une des quatre configurations de taille de fenêtre présentées ci-dessus.

Le traitement par transformée MDCT utilisant les fenêtres de transition (longue - courte) est illustré sur les figures 1b et 1c. Ces figures représentent les calculs menés, de la même manière que pour la figure 1a.

Sur les figures 1b et 1c, on a représenté seulement quelques fenêtres courtes, d'analyse, référencées FA (avec  $M_s=M/2$  dans l'exemple illustré). Néanmoins, en réalité, comme illustré sur la figure 2e, on prévoit une succession de plusieurs fenêtres courtes (avec typiquement  $M_s=M/8$ ). On comprendra donc que chaque fenêtre FA des figures 1b et 1c englobe en réalité une succession de fenêtres courtes.

La fenêtre de transition FTA (figure 1b), pour le calcul de la trame codée  $T'_i$  (référence calc  $T'_i$ ) comporte :

- une demi-fenêtre longue sur  $M$  échantillons, dans son front montant, et,
- dans son front descendant :
  - o une première région plate PLA (avec des valeurs de pondération égales à 1) sur  $(M/2-M_s/2)$  échantillons,
  - o une demi-fenêtre courte descendante sur  $M_s$  échantillons, et
  - o une seconde région plate (avec des valeurs de pondération égales à 0) sur  $(M/2-M_s/2)$  échantillons.

Pour le calcul de la trame codée suivante  $T'_{i+1}$  (référence calc  $T'_{i+1}$ ), les  $(M/2-M_s/2)$  premiers échantillons sont ignorés et donc non traités par les fenêtres courtes, les  $M_s$  échantillons suivants sont pondérés par le front montant de la fenêtre courte d'analyse FA telle qu'illustrée sur les figures 1b et 1c, et les  $M_s$  échantillons suivants sont pondérés par son front descendant.

Dans ce qui suit, les notations suivantes sont utilisées :

- M est la taille de trame longue,
- $M_S$  est la taille de trame courte.

5

Sur la figure 1b, l'échantillon **b** est synthétisé en utilisant uniquement les fenêtres courtes pour respecter l'analogie avec le calcul pour les fenêtres longues. Ensuite, du fait de la forme particulière de la demi-fenêtre de transition longue-courte, l'échantillon **a** est reconstruit directement à partir des fenêtres de transition d'analyse et de synthèse.

10

La fenêtre de transition est référencée FTA sur les figures 1b et 1c.

Sur la figure 1c, les échantillons correspondant à la zone de transition entre la fenêtre longue-courte et la fenêtre courte sont calculés. Par analogie avec le calcul pour les fenêtres longues de la figure 1a, on suit ici le traitement des échantillons notés **e** et **f** (entourés).

15

Deux exemples de situation de transition de fenêtres sont décrits ci-après.

Dans un premier exemple, on a détecté une attaque nécessitant l'utilisation de fenêtres courtes dans le signal audio à un instant  $t=720$  (figure 2e). Le codeur doit signifier au décodeur l'utilisation de fenêtres de transition long-court venant s'interposer entre les fenêtres longues utilisées préalablement et les fenêtres courtes, ultérieures.

20

Ainsi, le codeur indique au décodeur, successivement, les séquences :

25

- fenêtre longue
- fenêtre de transition long-court
- fenêtre courte
- fenêtre de transition court-long
- fenêtre longue.

30

Le décodeur applique alors une relation du type :

$$\hat{x}_{n+M} = \sum_{k=0}^{M-1} [X_k^{t+1} p_k^l(n) + X_k^t p_k^s(n+M)]$$

où  $p_k^l$  et  $p_k^s$  représentent les fonctions de synthèse des transformées aux temps  $t$  et  $t+1$ , qui peuvent être distinctes l'une de l'autre.

5

La reconstruction se fait comme précédemment, à la différence près que si les fonctions de base  $p_k^l$  et  $p_k^s$  sont de « tailles » différentes, alors on effectue, en référence à la figure 1b :

- une transformée DCT inverse de taille  $M$  des échantillons  $X_k^t$  donnant  $2M$  échantillons, 10
- une transformée DCT inverse de taille  $M_s$  des échantillons  $X_k^{t+1}$  donnant  $2M_s$  échantillons, les  $M_s$  premiers échantillons ayant un support temporel commun de longueur  $M_s$  dans une zone de recouvrement comportant la partie montante de la fenêtre courte, avec les échantillons issus de la transformée DCT inverse de taille  $M$  de la partie descendante de la fenêtre de transition FTA, 15
- une multiplication par la fenêtre de synthèse duale de la fenêtre de transition FTA et référencée FTS sur la figure 1b, pour la première moitié, et une multiplication par la fenêtre de synthèse courte FS pour la seconde moitié, et
- les additions de ces composantes fenêtrées sur la zone de recouvrement, le support temporel correspondant à une partie de fin de la trame initiale  $T_i$ . 20

Le décodeur est donc esclave du codeur et applique fidèlement les types de fenêtre décidés par le codeur.

- 25 Dans ce premier exemple, le codeur détecte une transition lors de l'arrivée des échantillons d'une première trame (par exemple la trame 1 de la figure 2e, comportant les échantillons entre les instants  $t=512$  et  $t=767$ ). Le codeur peut alors décider que la

fenêtre courante doit être une fenêtre de transition long-court, encodée, transmise et signalée au décodeur. Ensuite, huit fenêtres courtes sont successivement appliquées entre les échantillons  $t=624$  et  $t=911$ . Ainsi, au moment de la transition ( $t=720$ ), l'encodeur utilise les fenêtres courtes, ce qui permet une meilleure représentation temporelle du signal.

Dans un deuxième exemple, une transition est détectée à l'échantillon  $t=540$ . Lorsque le codeur reçoit les échantillons d'une première trame (la trame 0 de la figure 2e par exemple), il ne détecte pas de transition et sélectionne donc une fenêtre longue. Lors de l'arrivée des échantillons d'une seconde trame (trame 1 dans l'exemple de la figure 2e), suivante, le codeur détecte une attaque (à l'instant  $t=540$ ). Dans ce cas alors, la détection est effectuée trop tardivement et l'utilisation d'une fenêtre de transition ne permet pas de bénéficier de l'utilisation des supports temporels courts (fenêtres courtes) au moment de l'attaque. Le codeur doit alors anticiper l'utilisation de fenêtres courtes et, par conséquent, insérer un délai de codage additionnel correspondant à au moins  $M/2$  échantillons.

On comprendra alors qu'un inconvénient de l'art antérieur connu réside dans le fait qu'il est nécessaire d'introduire un retard supplémentaire à l'encodeur pour permettre de détecter une attaque dans le signal temporel d'une trame suivante et d'anticiper alors le passage aux fenêtres courtes. Cette « attaque » peut correspondre à un signal transitoire de forte intensité, tel qu'une plosive, par exemple, dans un signal de parole, ou encore à l'apparition d'un signal percussif dans une séquence musicale.

Dans certaines applications de télécommunications, le retard supplémentaire, nécessaire pour la détection de signaux transitoires et l'utilisation des fenêtres de transition, n'est pas acceptable. Ainsi par exemple, dans le codeur MPEG-4 AAC « Low Delay » (à faible retard), les fenêtres courtes ne sont pas utilisées, seules les fenêtres longues étant permises.

30

La présente invention vient améliorer la situation.

Elle propose une transition entre fenêtres ne nécessitant pas l'introduction d'un retard supplémentaire.

5 Elle vise à cet effet un procédé de codage/décodage, par transformée, d'un signal audionumérique représenté par une succession de trames, dans lequel :

- on prévoit au moins deux fenêtres de pondération, de longueurs respectives distinctes, et
- 10 - on utilise une fenêtre courte pour coder une trame dans laquelle un événement particulier a été détecté.

Cet événement particulier peut être par exemple un phénomène non stationnaire, tel qu'une attaque forte présente dans le signal audionumérique que contient la trame courante.

15

Plus particulièrement, pour le codage d'une trame courante, on cherche à détecter l'événement particulier dans cette trame courante, et :

- au moins si l'événement particulier est détecté en début de la trame courante, on applique une fenêtre courte pour coder la trame courante,
- 20 - tandis que si l'événement particulier n'est pas détecté dans la trame courante, on applique une fenêtre longue pour coder la trame courante.

On réitère ces étapes pour une trame suivante, de sorte qu'il est possible, au sens de l'invention, de coder une trame donnée en utilisant une fenêtre longue et de coder une trame qui suit immédiatement cette trame donnée en utilisant directement ensuite une

25 fenêtre courte, sans utiliser de fenêtre de transition comme dans l'art antérieur.

En s'offrant la possibilité de passer directement d'une fenêtre longue à une fenêtre courte, la détection de l'événement particulier peut être menée directement sur la trame en cours de codage et non plus sur la trame suivante, comme dans l'art antérieur. Dès

30 lors, un codage réalisé par le procédé au sens de l'invention est effectué sans retard

supplémentaire par rapport à une transformée MDCT de taille fixe, contrairement aux codages de l'art antérieur.

- D'autres caractéristiques et avantages de l'invention apparaîtront à l'examen de la description détaillée ci-après, et des dessins annexés sur lesquels, outre les figures 1, 1a, 1b, 1c, 2a, 2b, 2c, 2d, 2e, relatives à l'art antérieur et décrites ci-avant :
- la figure 3a illustre schématiquement un traitement de codage/décodage au sens de l'invention en suivant l'évolution d'échantillons **a** et **b**, comme sur la figure 1b décrite précédemment,
  - la figure 3b illustre schématiquement un traitement de codage/décodage au sens de l'invention en suivant l'évolution d'échantillons **e** et **f**, comme sur la figure 1c décrite précédemment, et
  - les figures 4a et 4b illustrent des exemples de variation des fonctions de pondération utilisées pour la compensation au décodage, menée dans la mise en œuvre de l'invention,
  - la figure 5a illustre un exemple de traitement qui peut être appliqué dans un codeur au sens de l'invention,
  - la figure 5b illustre un exemple de traitement qui peut être appliqué dans un décodeur au sens de l'invention, et
  - la figure 6 illustre les structures respectives d'un codeur et d'un décodeur et la communication de l'information du type de fenêtre utilisée au codage.

La présente invention permet d'éviter d'appliquer des fenêtres de transition au moins pour le passage d'une fenêtre longue à une fenêtre courte.

- Ainsi, en reprenant le deuxième exemple décrit précédemment en référence à la figure 2e, si un phénomène non stationnaire ou « attaque » est détecté à l'instant  $t=540$ , la présente invention propose d'utiliser une fenêtre longue pour la trame 0 (fenêtre s'étendant de l'instant  $t=256$  à l'instant  $t=511$ ). Ensuite, lors de l'acquisition des échantillons de la trame suivante ( $t=512$  à  $t=767$ ) et de la détection d'une attaque à  $t=540$ , le codeur utilise huit fenêtres courtes pour encoder les échantillons à partir de

l'instant  $t=368$  (correspondant à  $t=512-M/2-M_S/2$ ), jusqu'à l'instant  $t=655$  (correspondant à  $t=512+M/2+M_S/2-1$ , où

-  $2*M=512$  est la taille de la fenêtre longue, et

-  $2*M_S=64$  est la taille de la fenêtre courte, dans l'exemple décrit),

5 et ce, sans fenêtre de transition classique, asymétrique comme celle représentée sur les figures 1b et 1c relatives à l'art antérieur.

Au niveau du décodeur, lors de la réception de la trame encodée avec des fenêtres courtes, le décodeur procède alors aux opérations suivantes :

- 10
- réception d'une information issue du codeur indiquant que des fenêtres courtes doivent être utilisées pour la trame courante,
  - application d'un traitement avantageux pour compenser la transition directe d'une fenêtre longue à une fenêtre courte au codage, un exemple de ce traitement étant décrit en détail plus loin, en référence à la figure 5b.

15

Les figures 3a et 3b illustrent le procédé de codage/décodage au sens de l'invention pour obtenir d'une part les échantillons **a** et **b** qui se trouvent dans une zone sans recouvrement entre les fenêtres longues et courtes (figure 3a), et d'autre part les échantillons **e** et **f** se trouvant dans cette zone de recouvrement (figure 3b). En particulier, cette zone de recouvrement est définie par le front descendant de la fenêtre longue FL et le front montant de la première fenêtre courte FC.

20

Ainsi, en référence aux figures 3a et 3b, au codage, les échantillons des trames  $T_{i-1}$  et  $T_i$  sont pondérés par la fenêtre longue d'analyse FL pour constituer la trame codée  $T'_i$  et les échantillons de la trame suivante  $T_i$  et  $T_{i+1}$  sont pondérés directement par des fenêtres d'analyse courtes FC, sans appliquer de fenêtre de transition.

25

On remarquera aussi, en référence aux figures 3a et 3b, que la première fenêtre courte d'analyse FC est précédée par des valeurs non prises en compte par les fenêtres courtes (pour les échantillons précédant l'échantillon **e** dans l'exemple de la figure 3b). Plus

30

particulièrement, ce traitement s'applique aux  $M/2 - M_s/2$  premiers échantillons de la trame à coder  $T'_{i+1}$ , et ce, de façon analogue aux codeurs/décodeurs de l'art antérieur. De manière générale, on cherche à perturber le moins possible les traitements réalisés au codage, comme au décodage, par rapport à l'art antérieur. C'est ainsi qu'il est choisi  
5 par exemple d'ignorer les premiers échantillons de la trame codée  $T'_{i+1}$ .

Bien entendu, sur les figures 3a et 3b, on a représenté uniquement le cas de deux fenêtres courtes ( $M_s = M/2$ ) d'analyse FC. Néanmoins, on prévoit, comme dans l'art antérieur, une succession de plusieurs fenêtres courtes et chaque succession de fenêtres courtes est illustré sur ces figures 3a et 3b en portant la référence FC.  
10

Dans ce qui suit, on décrit deux exemples de réalisation pour décoder une trame  $T'_{i+1}$  qui a été codée en utilisant une fenêtre courte FC alors qu'une trame immédiatement précédente  $T'_i$  avait été codée en utilisant une fenêtre longue FL.  
15

Dans une première forme de réalisation, on s'affranchit complètement de l'utilisation de fenêtres de synthèse au décodage et l'on démontre que la propriété de reconstruction parfaite est assurée.

Sur la figure 3a, lors de la détection d'une attaque nécessitant un changement de fenêtre (d'une fenêtre longue directement à une fenêtre courte), on synthétise tout d'abord les échantillons à partir des fenêtres courtes uniquement (échantillon **b** sur la figure 3a). Puis, on compense dans la valeur  $v_1$  calculée à partir de la fenêtre longue d'analyse l'effet de l'échantillon **b** calculé au préalable. Le calcul de codage (trame codée  $T'_i$ ) pour l'échantillon **a** s'effectue alors comme suit :  
20  
25

$$v_1 = a \cdot h(M+n) + b \cdot h(2 \cdot M - 1 - n).$$

En revanche, l'échantillon **a** n'est pas pondéré dans la valeur de codage  $v_2$  puisque le calcul de pondération à partir de la fenêtre courte suivie d'une combinaison s'effectue sur un support temporel distinct (trame codée  $T'_{i+1}$ ), et on a après reconstruction à partir des fenêtres courtes :  
30

$$v_2 = b$$

On vérifie avantageusement la reconstruction parfaite dans le codage/décodage au sens de l'invention. En effet :

$$5 \quad a' = (v_1 - v_2 * h(2*M-1-n)) / h(M+n) = a$$

On remarquera aussi, au décodage, que les échantillons tirés des valeurs  $v_2=b$  et les suivants doivent être déterminés en premier, avant la détermination des échantillons du début de la trame (comme l'échantillon **a**). On procède donc à un renversement temporel lors du décodage.

Sur la figure 3b, on calcule les échantillons codés de la zone de transition entre la fenêtre longue FL (front descendant) et la première fenêtre courte FC (front montant), au niveau donc des échantillons **e** et **f**. L'expression des coefficients codés (ou « valeurs  $v_1$  et  $v_2$  » ci-après) est donnée, dans cette zone de recouvrement entre les deux fenêtres FL et FC, par les équations suivantes :

$$v_1 = e * h(M+n) + f * h(2*M-1-n), \text{ et}$$

$$v_2 = f * h_s(M_s-1-m) - e * h_s(m)$$

20 Au décodeur, on doit donc résoudre ce système d'équations à deux inconnues pour retrouver les valeurs des échantillons **e** et **f** :

$$e = [v_1 * h_s(M_s-1-m) - v_2 * h(2*M-1-n)] / [h(M+n) * h_s(M_s-1-m) + h_s(m) * h(2*M-1-n)]$$

$$f = [v_1 * h_s(m) + v_2 * h(M+n)] / [h_s(M_s-1-m) * h(M+n) + h(2*M-1-n) * h_s(m)]$$

On en déduit encore les formules vérifiant avantageusement la propriété de reconstruction parfaite :

$$e' = [v_1 * h_s(M_s+m) - v_2 * h(n)] / [h(M+n) * h_s(M_s+m) + h(2*M-1-n) * h_s(m)] = e,$$

et

$$f' = [v_1 * h_s(2*M_s-1-m) + v_2 * h(M-1-n)] / [h(M+n) * h_s(M_s+m) + h(2*M-1-n) * h_s(m)] = f,$$

avec  $m = n - M/2 + M_s/2$

On relèvera que la valeur  $v_2$  est pondérée par la fenêtre longue  $h$ , et ce, contrairement à ce qui était établi dans l'art antérieur (où  $v_2$  était pondérée par la fenêtre courte  $h_s$ , comme le montre le bas de la figure 1c).

- 5 Dans une seconde forme de réalisation, on conserve des fenêtres de synthèse au décodage. Elles sont de même forme que les fenêtres d'analyse (homologues ou duales des fenêtres d'analyse), telles qu'illustrées sur les figures 3a et 3b et portant la référence FLS pour une fenêtre longue, de synthèse, et la référence FCS pour une
- 10 fenêtre courte, de synthèse. Cette seconde forme de réalisation présente l'avantage d'être conforme au fonctionnement des décodeurs de l'état de l'art, à savoir utiliser une fenêtre de synthèse longue pour décoder une trame qui a été codée avec une
- fenêtre d'analyse longue et utiliser une série de fenêtres de synthèse courtes pour décoder une trame qui a été codée avec une série de fenêtres d'analyse courtes.
- 15 En revanche, on applique une correction de ces fenêtres de synthèse, par « compensation », pour décoder une trame qui a été codée avec une fenêtre longue, alors qu'elle aurait dû être codée avec une fenêtre de transition long-court. En d'autres
- termes, pour compenser l'effet du passage direct d'une fenêtre longue à une fenêtre
- 20 courte, au codeur, on réalise le traitement ci-après pour décoder une trame courante  $T'_{i+1}$  qui a été codée en utilisant une fenêtre courte FC tandis qu'une trame immédiatement précédente  $T'_i$  avait été codée en utilisant une fenêtre longue FL.

Les équations données ci-dessus pour le décodage et liant les échantillons **a**, **b**, **e**, **f** aux valeurs  $v_1$  et  $v_2$ , peuvent être réécrites sous la forme de sommes pondérées de deux

25 termes, comme suit, en procédant en particulier à un renversement temporel.

D'abord, on se place dans les premières fenêtres courtes de synthèse FCS et après la zone de recouvrement précitée (typiquement à l'échantillon  $v_2=\mathbf{b}$  et les suivants dans l'illustration à titre explicatif de la figure 3a). Pour le décodage de cette partie sans

30 recouvrement, à partir de fenêtres de synthèse courtes FCS uniquement, on décode d'abord les « valeurs » de la trame codée  $T'_{i+1}$  à partir de  $v_2=\mathbf{b}$  (figure 3a). Une fois

que les échantillons **b** et les suivants sont décodés, on applique la somme pondérée de deux termes suivante :

$$\hat{x}_n = w_{1,n}\tilde{l}_n + w_{2,n}s_{M-1-n}, \text{ avec } 0 \leq n < M/2 - Ms/2, \text{ où :}$$

- $\hat{x}_n$  représente les échantillons décodés (correspondant aux échantillons initiaux  $x_n$ ,  
5           puisqu'il y a reconstruction parfaite),
- la notation  $\tilde{l}_n$  désigne ce qui correspondrait à des échantillons qui auraient été  
          décodés (application d'une transformation inverse  $DCT^{-1}$ ) en utilisant une fenêtre  
          de synthèse longue FLS, sans correction, et
- $s_n$  représente les échantillons complètement décodés (typiquement l'échantillon **b**  
10           et les échantillons qui le suivent) à l'aide de la succession de fenêtres courtes de  
          synthèse FCS.

Les deux fonctions de pondération  $w_{1,n}$  et  $w_{2,n}$  s'écrivent alors :

$$w_{1,n} = \frac{1}{h^2(M+n)} \text{ et } w_{2,n} = -\frac{h(2M-n-1)}{h(M+n)} = -\frac{h(n)}{h(M+n)}, \text{ avec } 0 \leq n < M/2 - Ms/2$$

15

On comprendra que les « échantillons »  $\tilde{l}_n$  sont en réalité des valeurs incomplètement décodées par synthèse et pondération en utilisant la fenêtre longue de synthèse. Il s'agit typiquement des valeurs  $v_1$  de la figure 3a, multipliées par les coefficients  $h(M+n)$  de la fenêtre FLS, et dans lesquelles interviennent encore des échantillons de  
20           début de la trame  $T_i$ , tels que l'échantillon **a**.

On remarquera aussi que les échantillons **b** et les suivants sont, ici, déterminés en premier et s'écrivent dans la formule donnée ci-avant «  $s_{M-1-n}$  », illustrant ainsi le renversement temporel que propose le traitement de décodage dans cette seconde  
25           forme de réalisation.

On remarque aussi que la pondération effectuée par la fenêtre longue de synthèse FLS est évitée car celle-ci se retrouve retirée du terme  $w_{1,n}$  (du fait de la division par  $h(M+n)$ ).

- 5 Par ailleurs, pour la reconstruction de la portion d'échantillons couverte à la fois par la fenêtre longue FL (front descendant) et la première fenêtre courte FC (front montant), correspondant à la région des échantillons **e** à **f** de la figure 3b, on applique plutôt la combinaison suivante de deux termes pondérés :

$$\hat{x}_n = w'_{1,n} \tilde{s}_m + w'_{2,n} \tilde{l}_n, \text{ avec } m = n - M/2 + M_s/2 \text{ et } M/2 - M_s/2 \leq n < M/2 + M_s/2$$

- 10 Comme précédemment, les termes  $\tilde{l}_n$  constituent les valeurs incomplètement reconstituées par synthèse et pondération par la fenêtre longue de synthèse FLS et les termes  $\tilde{s}_n$  représentent les valeurs incomplètement reconstituées à partir du front montant de la première fenêtre de synthèse courte FCS.

- 15 Les fonctions de pondération  $w'_{1,n}$  et  $w'_{2,n}$  sont données ici par :

$$w'_{1,n} = \frac{h(n) - \frac{h_s(m)h_s(M_s-1-m)}{h(M-1-n)}}{h(M-1-n)h_s(M_s-1-m) + h(n)h_s(m)} \text{ et } w'_{2,n} = \frac{\frac{h_s(M_s-1-m)}{h(M-1-n)}}{h(M-1-n)h_s(M_s-1-m) + h(n)h_s(m)}$$

- 20 Toutes ces fonctions de pondération  $w_{1,n}$ ,  $w_{2,n}$ ,  $w'_{1,n}$  et  $w'_{2,n}$  sont constituées d'éléments figés qui ne dépendent que des fenêtres longues et courtes. Des exemples de variation de telles fonctions de pondération sont présentés sur les figures 4a et 4b. Dans une réalisation avantageuse, les valeurs prises par ces fonctions peuvent être calculées a priori (tabulées) et stockées définitivement en mémoire d'un décodeur au sens de l'invention.

- 25 Ainsi, en référence à la figure 5b, le traitement au décodage d'une trame  $T'_i$  qui a été codée en passant directement d'une fenêtre d'analyse longue à une fenêtre d'analyse courte, peut comporter les étapes suivantes, dans un exemple de réalisation. Pour

décoder la trame  $T'_i$  (étape 60), on applique d'abord une fenêtre de synthèse courte (étape 61) pour décoder la valeur  $v_2=b$  (étape 63) de fin de trame. On s'appuie ici sur une trame codée suivante  $T'_{i+1}$  (étape 62) pour déterminer  $b$ . On applique ensuite une fenêtre de synthèse longue (étape 64) pour décoder les échantillons  $a$  de début de trame  $T'_i$  (étape 65), en appliquant la compensation pour tout  $n$  compris entre 0 et  $M/2-Ms/2$  à l'aide de la relation  $\hat{x}_n = w_{1,n}\tilde{l}_n + w_{2,n}s_{M-1-n}$  (étape 67) et en utilisant les valeurs de pondération pré-calculées et tabulées  $w_{1,n}$  et  $w_{2,n}$  (étape 66).

Le décodage de la région « centrale » de la trame codée  $T'_i$  (entre **e** et **f**), donc pour  $n$  compris entre  $M/2-Ms/2$  et  $M/2+Ms/2$ , peut être mené en parallèle (signe « + » de la figure 5b) en utilisant à la fois les fenêtres de synthèse courte et longue (étape 68) et en appliquant en particulier la compensation (étape 69) à partir de la relation  $\hat{x}_n = w'_{1,n}\tilde{s}_m + w'_{2,n}\tilde{l}_n$ , où  $m = n - M/2 + Ms/2$  et avec les valeurs de pondération  $w'_{1,n}$  et  $w'_{2,n}$  pré-calculées et tabulées (étape 70). On déduit finalement de ce traitement (étape 71) les valeurs de tous les types d'échantillons **a**, **b**, **e** ou **f** de la trame initiale  $T_i$ .

Les première et seconde formes de réalisation décrites ci-avant, au décodage d'une trame  $T'_i$  qui a été codée en passant directement d'une fenêtre d'analyse longue à une fenêtre d'analyse courte, garantissent une reconstruction parfaite et permettent alors, au codage, de passer effectivement d'une fenêtre longue à une fenêtre courte, directement.

On décrit maintenant, en référence à la figure 5a, un exemple de réalisation dans lequel il est proposé de s'affranchir, au codage, de l'application d'une fenêtre de transition long-court, au moins dans certains cas.

Sur réception d'une trame  $T_i$  (étape 50), on cherche dans le signal audionumérique présent directement dans cette trame  $T_i$  la présence d'un phénomène non stationnaire, tel qu'une attaque ATT (test 51). Tant qu'aucun phénomène de ce type n'est détecté

(flèche n en sortie du test 51), on continue l'application de fenêtres longues (étape 52) pour le codage de cette trame  $T_i$  (étape 56). Sinon (flèche o en sortie du test 51), on cherche à déterminer si le phénomène ATT est au début (par exemple dans la première moitié) de la trame courante  $T_i$  (test 53), auquel cas (flèche o en sortie du test 53) on applique directement une fenêtre courte (étape 54), et plus exactement une série de fenêtres courtes, pour le codage de la trame  $T_i$  (étape 56). Cette réalisation permet alors d'éviter une fenêtre de transition et de ne pas attendre la trame suivante  $T_{i+1}$  pour appliquer une fenêtre courte.

Ainsi, on comprendra que, contrairement à l'état de l'art, on peut détecter un événement particulier tel qu'un phénomène non stationnaire directement dans la trame en cours de codage  $T_i$  et non pas dans une trame suivante  $T_{i+1}$ . Le retard du codage au sens de l'invention est alors réduit par rapport à celui de l'art antérieur. En effet, si le phénomène non stationnaire est détecté en début de trame courante, on applique directement une fenêtre courte, alors que dans l'art antérieur, il aurait fallu détecter le phénomène non stationnaire dans une trame suivante  $T_{i+1}$  de manière à pouvoir appliquer une fenêtre de transition à la trame en cours de codage  $T_i$ .

En référence encore à la figure 5a, si le phénomène non stationnaire est détecté en fin (par exemple dans la seconde moitié) de la trame courante  $T_i$  (flèche n en sortie du test 53), on peut avantageusement choisir d'appliquer une fenêtre de transition (étape 55) pour le codage de la trame en cours  $T_i$  (étape 56), avant d'appliquer une succession de fenêtres courtes. Cette réalisation permet notamment de proposer un traitement équivalent à celui de l'état de l'art, tout en assurant un retard de codage réduit.

Ainsi, en termes plus génériques, on prévoit dans ce mode de réalisation au moins trois fenêtres de pondération :

- une fenêtre courte,
- une fenêtre longue, et
- une fenêtre de transition pour passer d'une utilisation de la fenêtre longue à une utilisation de la fenêtre courte,

et si un événement particulier tel qu'un phénomène non stationnaire est détecté en fin de trame courante (étape 53), on applique une fenêtre de transition (étape 55) pour coder (étape 56) la trame courante ( $T_i$ ).

- 5 Dans une variante de ce mode de réalisation, il peut être prévu, pour un passage entre une utilisation de fenêtre longue à une utilisation de fenêtre courte :
- pour une trame courante  $T_i$ , d'utiliser une fenêtre longue FL,
  - et pour une trame immédiatement consécutive  $T_{i+1}$ , d'utiliser directement une fenêtre courte FC, sans utiliser de fenêtre de transition, quand bien même
- 10 l'événement particulier serait détecté en fin de trame courante.

Cette variante présente l'avantage suivant. Comme le codeur doit envoyer au décodeur une information de changement de type de fenêtre, cette information peut être codée sur un seul bit car il n'y a plus à informer le décodeur du choix entre une fenêtre courte

15 et une fenêtre de transition.

On peut néanmoins conserver une fenêtre de transition pour passer d'une fenêtre courte à une fenêtre longue et, notamment pour assurer encore la transmission de l'information de changement de type de fenêtre sur un seul bit, suite à la réception

20 d'une information de passage de la fenêtre longue à la fenêtre courte, le décodeur peut à cet effet :

- utiliser la fenêtre courte,
  - puis, en l'absence de réception d'information de changement de type de fenêtre, utiliser une fenêtre de transition d'une fenêtre courte à une fenêtre longue,
- 25 - puis enfin, utiliser une fenêtre longue.

On a illustré sur la figure 6 la communication d'information du type de fenêtre utilisée au codage, d'un codeur 10 vers un décodeur 20. On rappelle que le codeur 10 comporte un module de détection 11 d'un événement particulier tel qu'une attaque

30 forte dans le signal contenu dans une trame  $T_i$  en cours de codage et qu'il déduit de cette détection le type de fenêtre à utiliser. A cet effet, un module 12 sélectionne le

type de fenêtre à utiliser et transmet cette information au module 13 de codage qui délivre la trame  $T'_i$  codée en utilisant la fenêtre d'analyse FA sélectionnée par le module 12. La trame codée  $T'_i$  est transmise au décodeur 20, avec l'information INF sur le type de fenêtre utilisée au codage (généralement dans un même flux). Le  
5 décodeur 20 comporte un module 22 de sélection de fenêtre de synthèse FS selon l'information INF reçue du codeur 10 et le module 23 applique le décodage de la trame  $T'_i$  pour délivrer une trame décodée  $\hat{T}_i$ .

La présente invention vise aussi un codeur tel que le codeur 10 de la figure 6 pour la  
10 mise en œuvre du procédé au sens de l'invention et plus particulièrement pour la mise en œuvre du traitement illustré sur la figure 5a, ou de sa variante décrite précédemment (transmission de l'information de changement de type de fenêtre sur un seul bit).

La présente invention vise aussi un programme informatique, destiné à être stocké en  
15 mémoire d'un tel codeur et comportant des instructions pour la mise en œuvre d'un tel traitement, ou de sa variante, lorsqu'un tel programme est exécuté par un processeur du codeur. A ce titre, la figure 5a peut représenter l'organigramme d'un tel programme.

On rappelle que le codeur 10 utilise des fenêtres d'analyse FA et le décodeur 20 peut  
20 utiliser des fenêtres de synthèse FS, selon la seconde forme de réalisation précitée, ces fenêtres de synthèse étant homologues des fenêtres d'analyse FA, en procédant toutefois à la correction par compensation décrite précédemment (en utilisant les fonctions de pondération  $w_{1,n}$ ,  $w_{2,n}$ ,  $w'_{1,n}$  et  $w'_{2,n}$ ).

25 La présente invention vise aussi un autre programme informatique, destiné à être stocké en mémoire d'un décodeur par transformée tel que le décodeur 20 illustré sur la figure 6, et comportant des instructions pour la mise en œuvre du décodage selon la première forme de réalisation, ou selon la seconde forme de réalisation décrite ci-avant en référence à la figure 5b, lorsqu'un tel programme est exécuté par un processeur de

ce décodeur 20. A ce titre, la figure 5b peut représenter l'organigramme d'un tel programme.

5 La présente invention vise aussi le décodeur par transformée, lui-même, comportant alors une mémoire stockant les instructions d'un programme informatique pour le décodage.

10 En termes génériques, le procédé de décodage au sens de l'invention, par transformée, d'un signal représenté par une succession de trames qui ont été codées en utilisant au moins deux types de fenêtres de pondération, de longueurs respectives distinctes, se fait comme suit.

En cas de réception d'une information de passage d'une fenêtre longue à une fenêtre courte :

- 15 - on détermine des échantillons (de type **b**) à partir d'un décodage appliquant un type de fenêtre de synthèse courte FCS à une trame donnée  $T'_{i+1}$  qui a été codée en utilisant une fenêtre d'analyse courte FC, et
- on obtient des échantillons complémentaires en :
- décodant partiellement (application d'une transformée inverse  $DCT^{-1}$ ) une
  - 20 trame  $T'_i$  précédant la trame donnée et qui a été codée en utilisant un type de fenêtre d'analyse longue FL, et
  - en appliquant une combinaison de deux termes pondérés faisant intervenir des fonctions de pondération qui peuvent être tabulées et stockées en mémoire d'un décodeur.

25 Dans la seconde forme de réalisation ci-avant, il s'agit des fonctions notées  $w_{1,n}$ ,  $w_{2,n}$ ,  $w'_{1,n}$ ,  $w'_{2,n}$ .

Toutefois, ce traitement générique au décodage s'applique dans les deux cas des première et seconde formes de réalisation.

Dans la seconde forme de réalisation :

- on détermine d'abord (étape 63 de la figure 5b) les échantillons (**b**) issus de la trame donnée ( $T'_{i+1}$ ), et
- on en déduit (étapes 65-67) des échantillons (**a**) correspondant temporellement au début de la trame précédente ( $T'_i$ ), ces échantillons étant issus d'un décodage appliquant une fenêtre de synthèse longue FLS et propre à la seconde forme de réalisation.

Dans ce cas, pour :

- une trame comportant M échantillons,
- une fenêtre longue comportant 2M échantillons,
- une fenêtre courte comportant 2Ms échantillons, Ms étant inférieur à M,

les échantillons  $\hat{x}_n$ , pour n compris entre 0 et  $(M/2 - Ms/2)$ , n=0 correspondant au début d'une trame en cours de décodage, sont donnés par une combinaison de deux termes

pondérés du type :

$$\hat{x}_n = w_{1,n} \tilde{l}_n + w_{2,n} s_{M-1-n}, \text{ où :}$$

- $\tilde{l}_n$  sont des valeurs (v1) issues de la trame précédente  $T'_i$ ,
- $s_{M-1-n}$  sont des échantillons (**b**) déjà décodés en utilisant des fenêtres de synthèse courtes appliquées à la trame donnée  $T'_{i+1}$ , et
- $w_{1,n}$  et  $w_{2,n}$  sont des fonctions de pondération dont les valeurs prises en fonction de n peuvent être tabulées et stockées en mémoire du décodeur.

Sinon, pour n compris entre  $(M/2 - Ms/2)$  et  $(M/2 + Ms/2)$ , les échantillons  $\hat{x}_n$  sont donnés par une combinaison de deux termes pondérés du type :

$$\hat{x}_n = w'_{1,n} \tilde{s}_m + w'_{2,n} \tilde{l}_n, \text{ avec } m = n - M/2 + Ms/2, \text{ où :}$$

- $\tilde{l}_n$  sont des valeurs v1 issues de la trame précédente  $T'_i$ ,
- $\tilde{s}_m$  sont des valeurs v2 issues de la trame donnée  $T'_{i+1}$ , et
- $w'_{1,n}$  et  $w'_{2,n}$  sont des fonctions de pondération dont les valeurs prises en fonction de n peuvent être aussi tabulées et stockées en mémoire du décodeur.

La présente invention permet donc d'offrir la transition entre fenêtres avec un retard réduit par rapport à l'art antérieur tout en conservant la propriété de reconstruction parfaite de la transformée. Ce procédé peut être appliqué avec tous types de fenêtres  
5 (fenêtres non symétriques et fenêtres d'analyse et synthèse différentes) et pour différentes transformées et bancs de filtres.

Les traitements de compensation présentés ci-dessus dans le cas d'une transition d'une fenêtre longue vers une fenêtre de taille plus courte s'étendent naturellement et de  
10 façon analogue au cas d'une transition d'une fenêtre courte vers une fenêtre de taille supérieure. Dans ce cas, l'absence de fenêtre de transition court-long peut être compensée au décodeur par une pondération analogue au cas présenté ci-dessus.

L'invention s'applique alors à tout codeur par transformée, notamment ceux prévus  
15 pour des applications conversationnelles interactives, comme dans la norme MPEG-4 « AAC - Low Delay », mais aussi à des transformées différentes des transformées MDCT, notamment les transformées précitées ELT (pour « Extended Lapped Transform ») et leurs extensions bi-orthogonales.

**REVENDEICATIONS**

1. Procédé de codage/décodage, par transformée, d'un signal audionumérique représenté par une succession de trames,
- 5 dans lequel :
- on prévoit au moins deux fenêtres de pondération, de longueurs respectives distinctes, et
  - on utilise une fenêtre courte pour coder une trame dans laquelle un événement particulier a été détecté,
- 10 caractérisé en ce que, pour le codage d'une trame courante ( $T_i$ ), on cherche à détecter (51) l'événement particulier dans ladite trame courante ( $T_i$ ), et :
- au moins si l'événement particulier est détecté en début de trame courante (53), on applique une fenêtre courte (54) pour coder (56) la trame courante ( $T_i$ ),
  - si l'événement particulier n'est pas détecté dans la trame courante, on applique
- 15 une fenêtre longue (54) pour coder (56) la trame courante ( $T_i$ ),  
et on réitère ces étapes pour une trame suivante ( $T_{i+1}$ ).
2. Procédé selon la revendication 1, caractérisé en ce que ledit événement particulier est un phénomène non stationnaire.
- 20
3. Procédé selon l'une des revendications 1 et 2, dans lequel on prévoit au moins trois fenêtres de pondération :
- une fenêtre courte,
  - une fenêtre longue, et
- 25 - une fenêtre de transition pour passer d'une utilisation de la fenêtre longue à une utilisation de la fenêtre courte,
- caractérisé en ce que si l'événement particulier est détecté en fin de trame courante (53), on applique la fenêtre de transition (55) pour coder (56) la trame courante ( $T_i$ ).
- 30
4. Procédé selon l'une des revendications 1 et 2, caractérisé en ce que, pour un passage entre une utilisation de fenêtre longue à une utilisation de fenêtre courte :

- pour une trame courante, on utilise la fenêtre longue,
- et pour une trame immédiatement consécutive, on utilise directement la fenêtre courte, sans utiliser de fenêtre de transition,

5 et en ce qu'une information de changement de type de fenêtre, envoyée d'un codeur à un décodeur homologue, est codée sur un seul bit.

5. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'un décodeur recevant une information de passage de la fenêtre longue à la fenêtre courte :

- utilise la fenêtre courte,
- 10 - puis, en l'absence de réception d'information de changement de type de fenêtre, utilise une fenêtre de transition d'une fenêtre courte à une fenêtre longue,
- puis utilise une fenêtre longue.

15 6. Procédé selon l'une des revendications précédentes, dans lequel un décodeur utilise des fenêtres de synthèse, duales de fenêtres d'analyse qu'utilise un codeur homologue du décodeur, caractérisé en ce qu'au décodage d'une trame ( $T'_i$ ) codée en passant d'une fenêtre longue (FL) à une fenêtre courte (FC), une compensation est menée et le codage/décodage présente une propriété de reconstruction parfaite de la transformée.

20 7. Procédé de décodage, par transformée, d'un signal représenté par une succession de trames qui ont été codées en utilisant au moins deux types de fenêtres de pondération, de longueurs respectives distinctes,

dans lequel, en cas de réception d'une information de passage d'une fenêtre longue à une fenêtre courte :

- 25 - on détermine (63) des échantillons (**b**) à partir d'un décodage appliquant un type de fenêtre de synthèse courte (61) à une trame donnée ( $T'_{i+1}$ ) qui a été codée en utilisant une fenêtre d'analyse courte, et
- on obtient des échantillons complémentaires (67,69) en :
  - décodant partiellement ( $DCT^{-1}$ ) une trame ( $T'_i$ ) précédant la trame donnée et
  - 30 qui a été codée en utilisant un type de fenêtre d'analyse longue, et

- en appliquant une combinaison de deux termes pondérés faisant intervenir des fonctions de pondération ( $w_{1,n}$ ,  $w_{2,n}$  ;  $w'_{1,n}$ ,  $w'_{2,n}$ ) tabulées et stockées en mémoire d'un décodeur.

5 8. Procédé selon la revendication 7, caractérisé en ce que :

- on détermine d'abord (63) les échantillons (**b**) issus de la trame donnée ( $T'_{i+1}$ ), et
- on en déduit (65-67) des échantillons (**a**) correspondant temporellement au début de la trame précédente ( $T'_i$ ), ces échantillons étant issus d'un décodage appliquant une fenêtre de synthèse longue.

10

9. Procédé selon la revendication 8, dans lequel :

- une trame comporte M échantillons,
- une fenêtre longue comporte 2M échantillons,
- une fenêtre courte comporte 2Ms échantillons, Ms étant inférieur à M,

15 caractérisé en ce que les échantillons  $\hat{x}_n$ , pour n compris entre 0 et (M/2-Ms/2), n=0 correspondant au début d'une trame en cours de décodage, sont donnés par une combinaison de deux termes pondérés du type :

$$\hat{x}_n = w_{1,n} \tilde{l}_n + w_{2,n} s_{M-1-n}, \text{ où :}$$

- $\tilde{l}_n$  sont des valeurs (v1) issues de la trame précédente ( $T'_i$ ),
- 20 -  $s_{M-1-n}$  sont des échantillons (**b**) déjà décodés en utilisant des fenêtres de synthèse courtes appliquées à la trame donnée ( $T'_{i+1}$ ), et
- $w_{1,n}$  et  $w_{2,n}$  sont des fonctions de pondération dont les valeurs prises en fonction de n sont tabulées et stockées en mémoire du décodeur.

25 10. Procédé selon l'une des revendications 7 à 9, dans lequel :

- une trame comporte M échantillons,
- une fenêtre longue comporte 2M échantillons,
- une fenêtre courte comporte 2Ms échantillons, Ms étant inférieur à M,

caractérisé en ce que les échantillons  $\hat{x}_n$ , pour  $n$  compris entre  $(M/2-Ms/2)$  et  $(M/2+Ms/2)$ ,  $n=0$  correspondant au début d'une trame en cours de décodage,, sont donnés par une combinaison de deux termes pondérés du type :

$$\hat{x}_n = w'_{1,n} \tilde{s}_m + w'_{2,n} \tilde{l}_n, \text{ avec } m = n - M/2 + Ms/2, \text{ où :}$$

- 5
- $\tilde{l}_n$  sont des valeurs (v1) issues de la trame précédente ( $T'_i$ ),
  - $\tilde{s}_m$  sont des valeurs (v2) issues de la trame donnée ( $T'_{i+1}$ ), et
  - $w'_{1,n}$  et  $w'_{2,n}$  sont des fonctions de pondération dont les valeurs prises en fonction de  $n$  sont tabulées et stockées en mémoire du décodeur.
- 10
11. Codeur par transformée, d'un signal audionumérique représenté par une succession de trames, le codeur utilisant au moins deux fenêtres de pondération, de longueurs respectives distinctes, caractérisé en ce qu'il comporte au moins :
- un module de détection (11) d'un événement particulier dans une trame courante à coder ( $T_i$ ),
- 15
- un module de sélection (12) :
    - \* d'une fenêtre d'analyse courte pour coder la trame courante ( $T_i$ ) au moins si l'événement particulier est détecté en début de trame courante, et
    - \* d'une fenêtre d'analyse longue pour coder la trame courante ( $T_i$ ) si l'événement particulier n'est pas détecté dans la trame courante,
- 20
- et un module de codage (13), appliquant la fenêtre d'analyse sélectionnée par le module de sélection (12), pour le codage de la trame courante ( $T_i$ ).

25

12. Programme informatique, destiné à être stocké en mémoire d'un codeur selon la revendication 11, caractérisé en ce qu'il comporte des instructions pour la mise en œuvre du procédé selon l'une des revendications 1 à 6, lorsqu'il est exécuté par un processeur d'un tel codeur.

13. Programme informatique, destiné à être stocké en mémoire d'un décodeur par transformée, caractérisé en ce qu'il comporte des instructions pour la mise en œuvre du

procédé de décodage selon l'une des revendications 7 à 10, lorsqu'il est exécuté par un processeur d'un tel décodeur.

14. Décodeur par transformée, caractérisé en ce qu'il comporte une mémoire stockant  
5 les instructions d'un programme informatique selon la revendication 13.

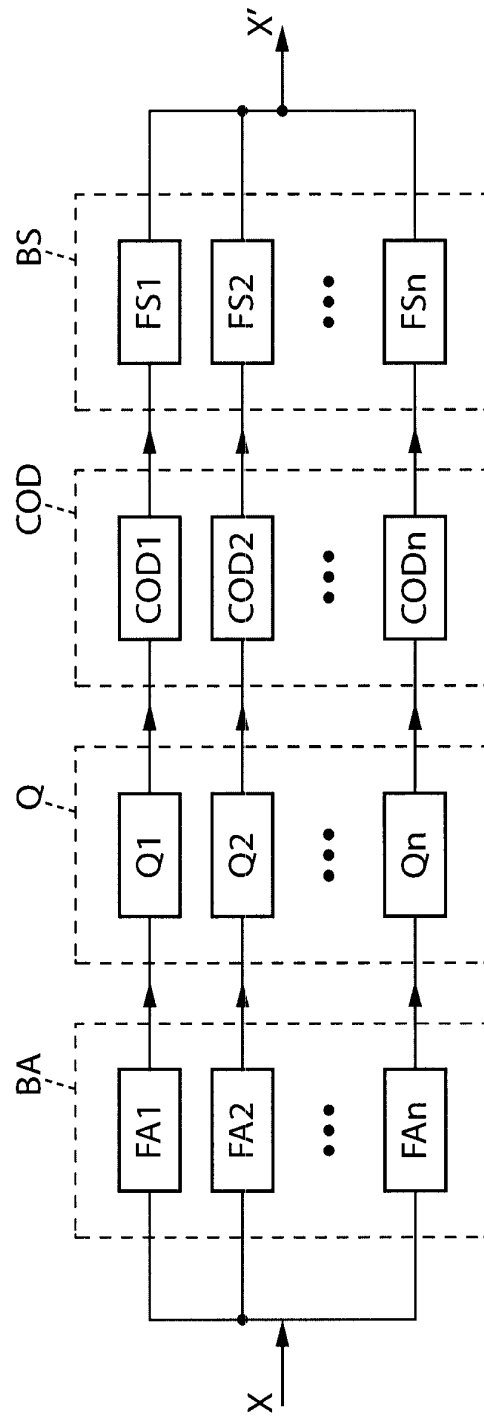


FIG. 1

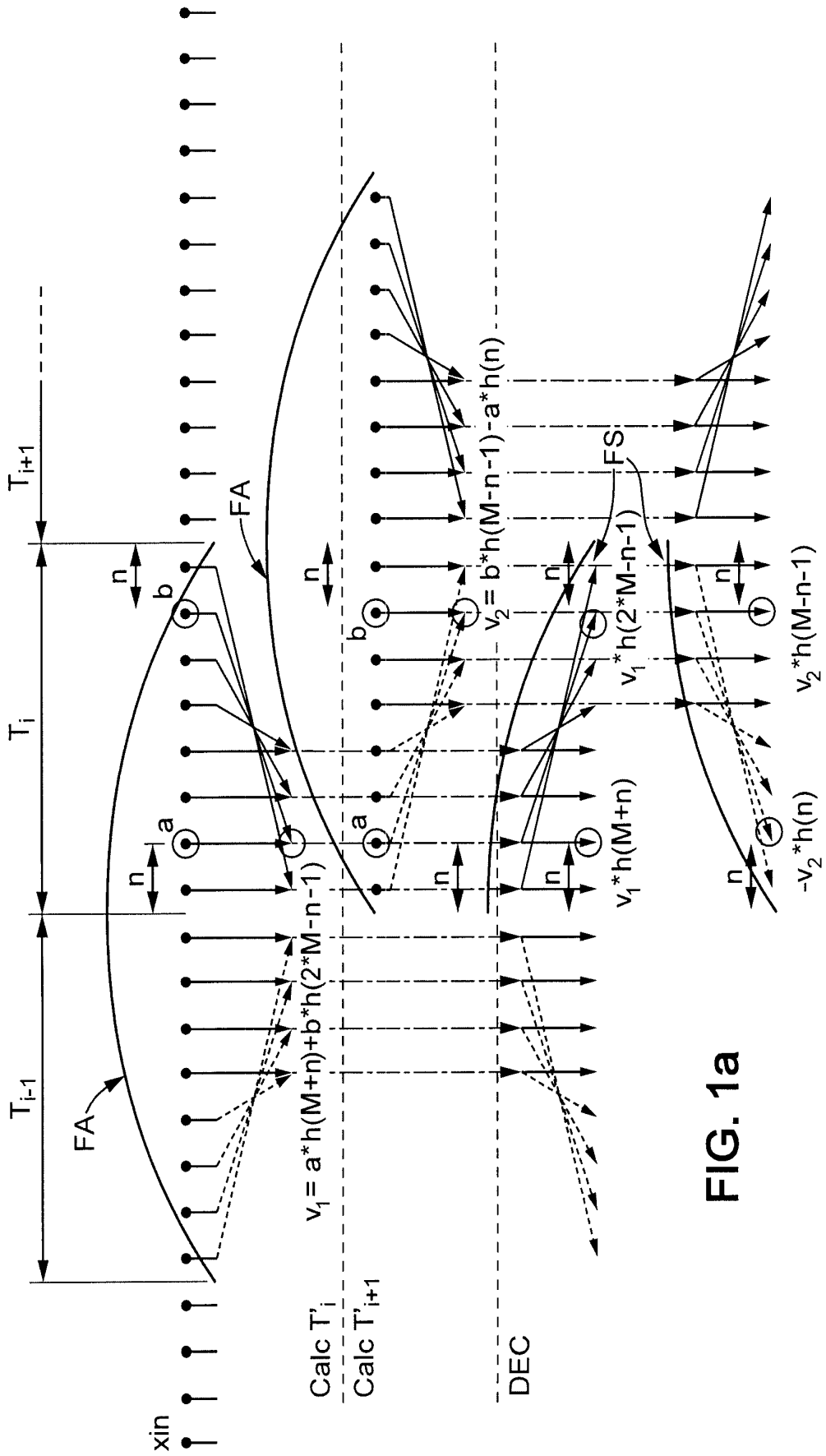


FIG. 1a

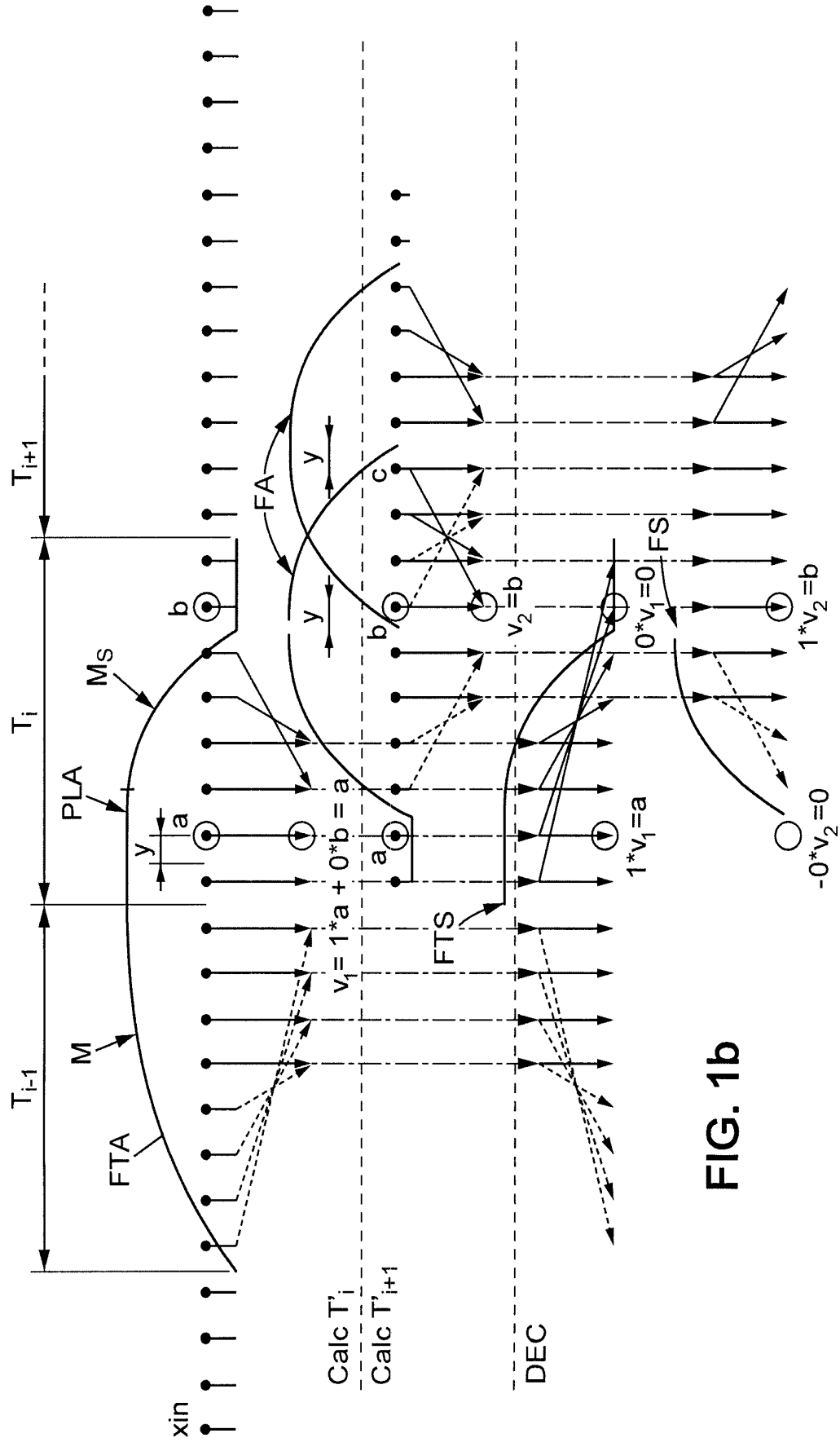


FIG. 1b

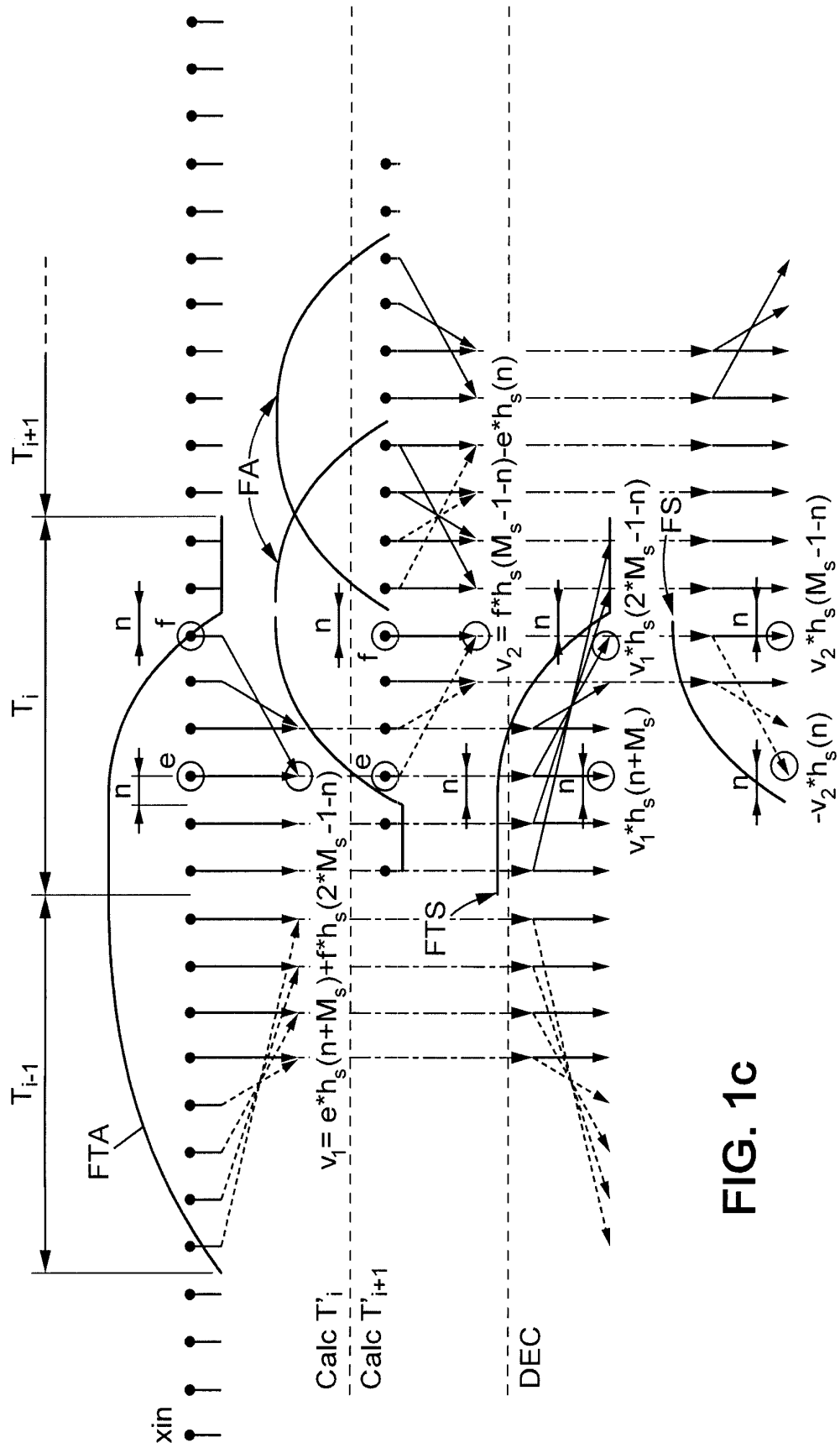


FIG. 1c

5/12

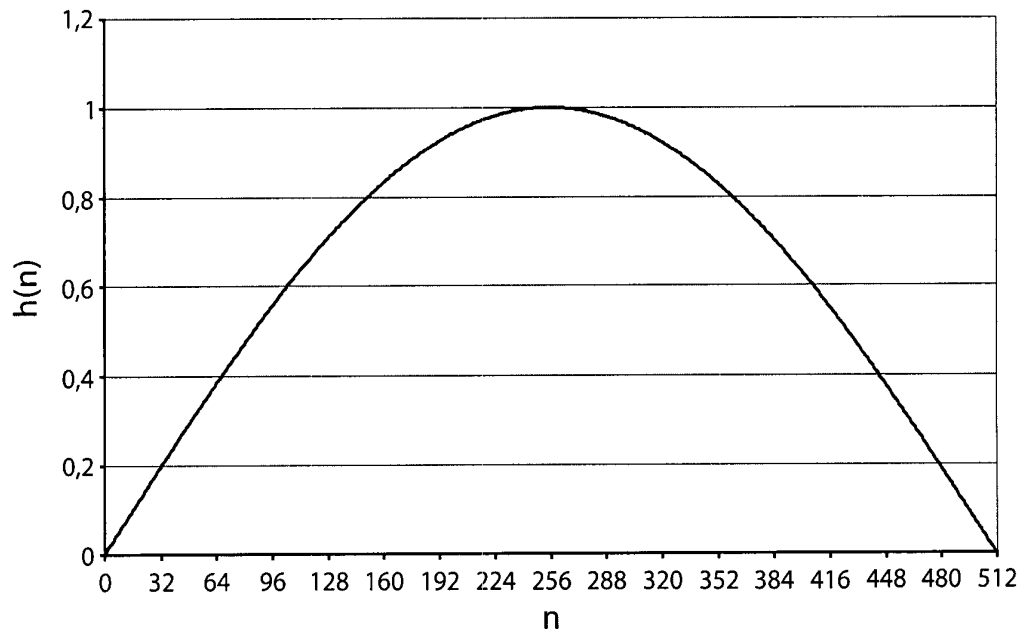


FIG. 2a

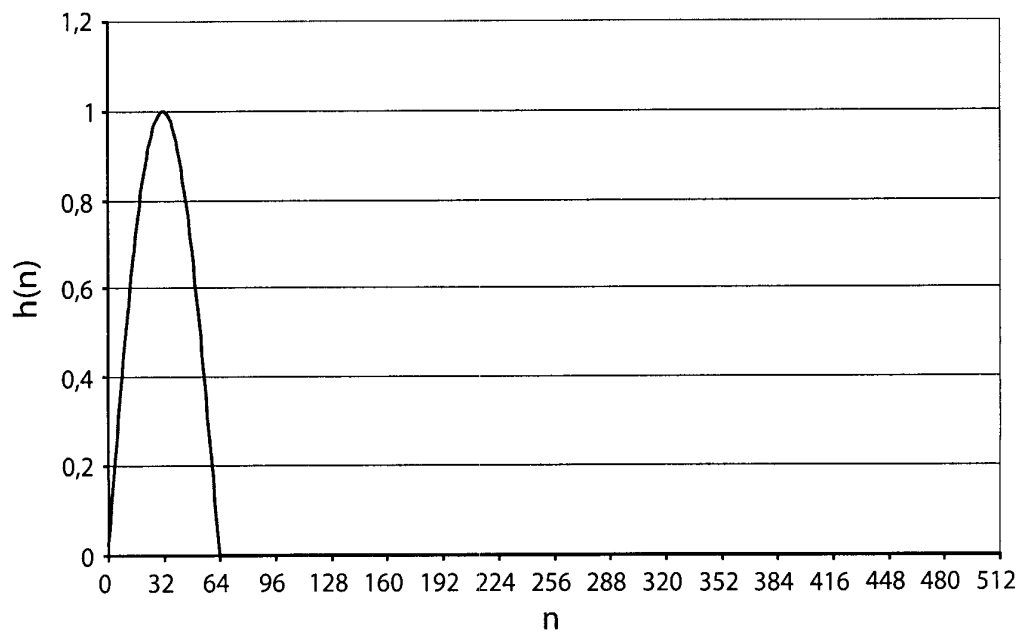


FIG. 2b

6/12

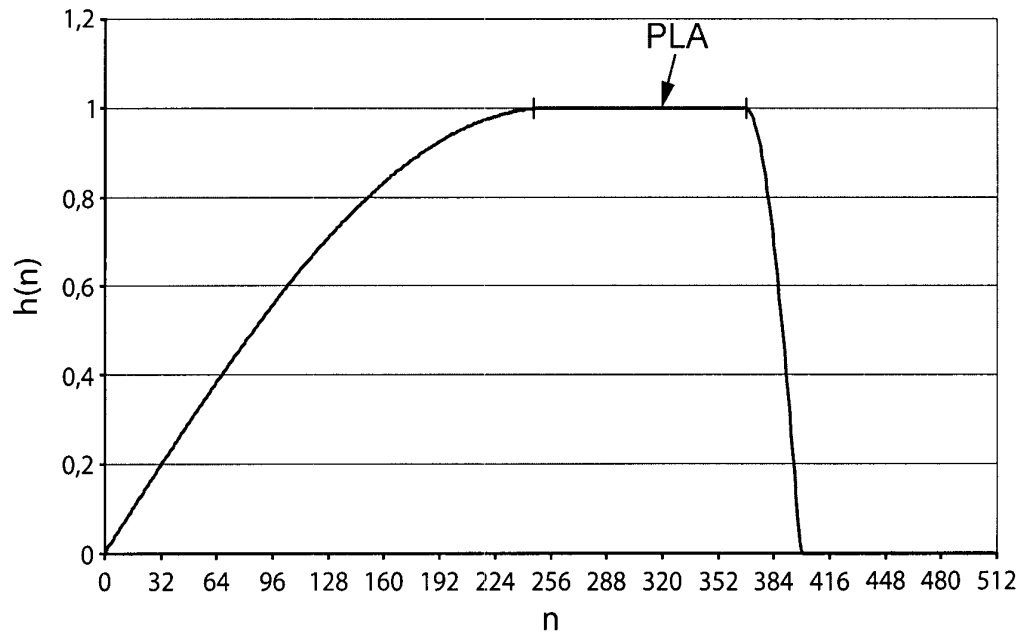


FIG. 2c

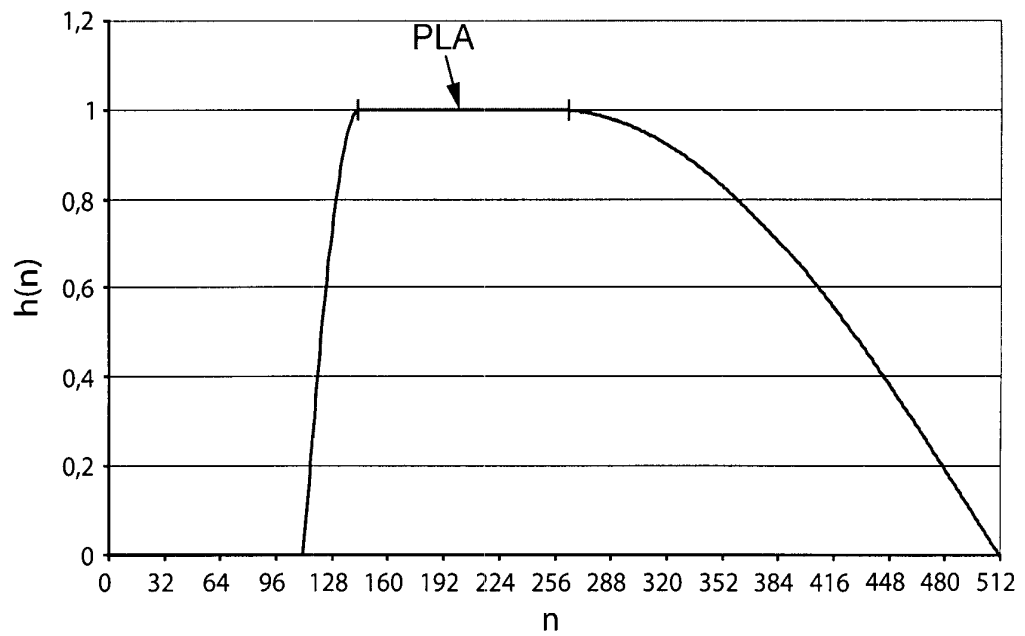


FIG. 2d

7/12

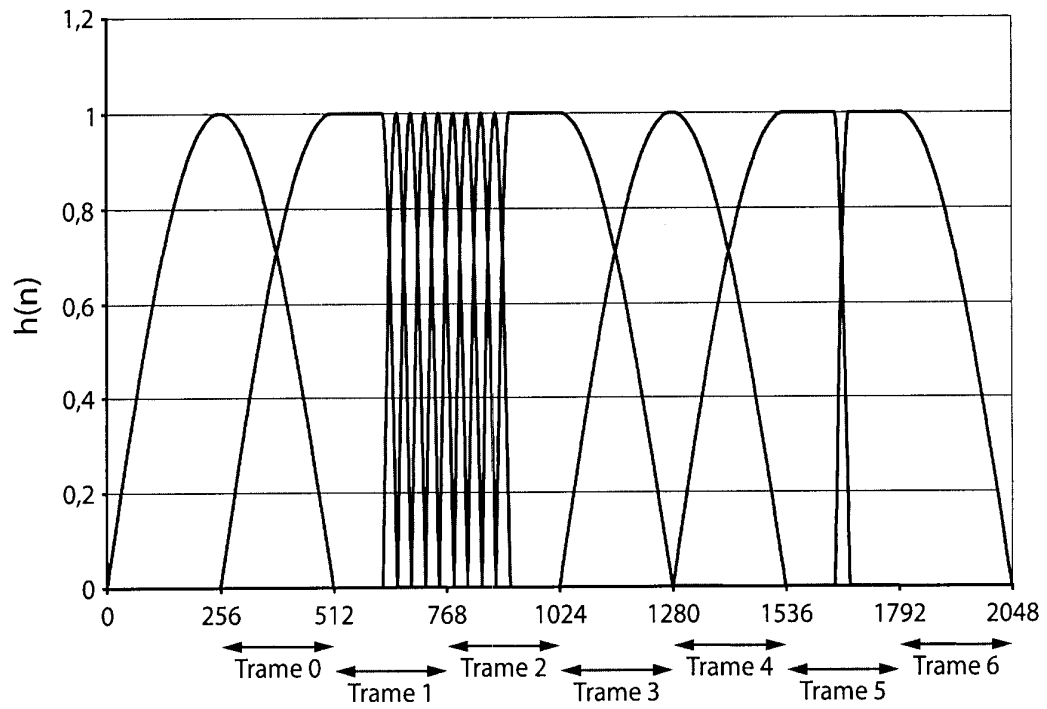
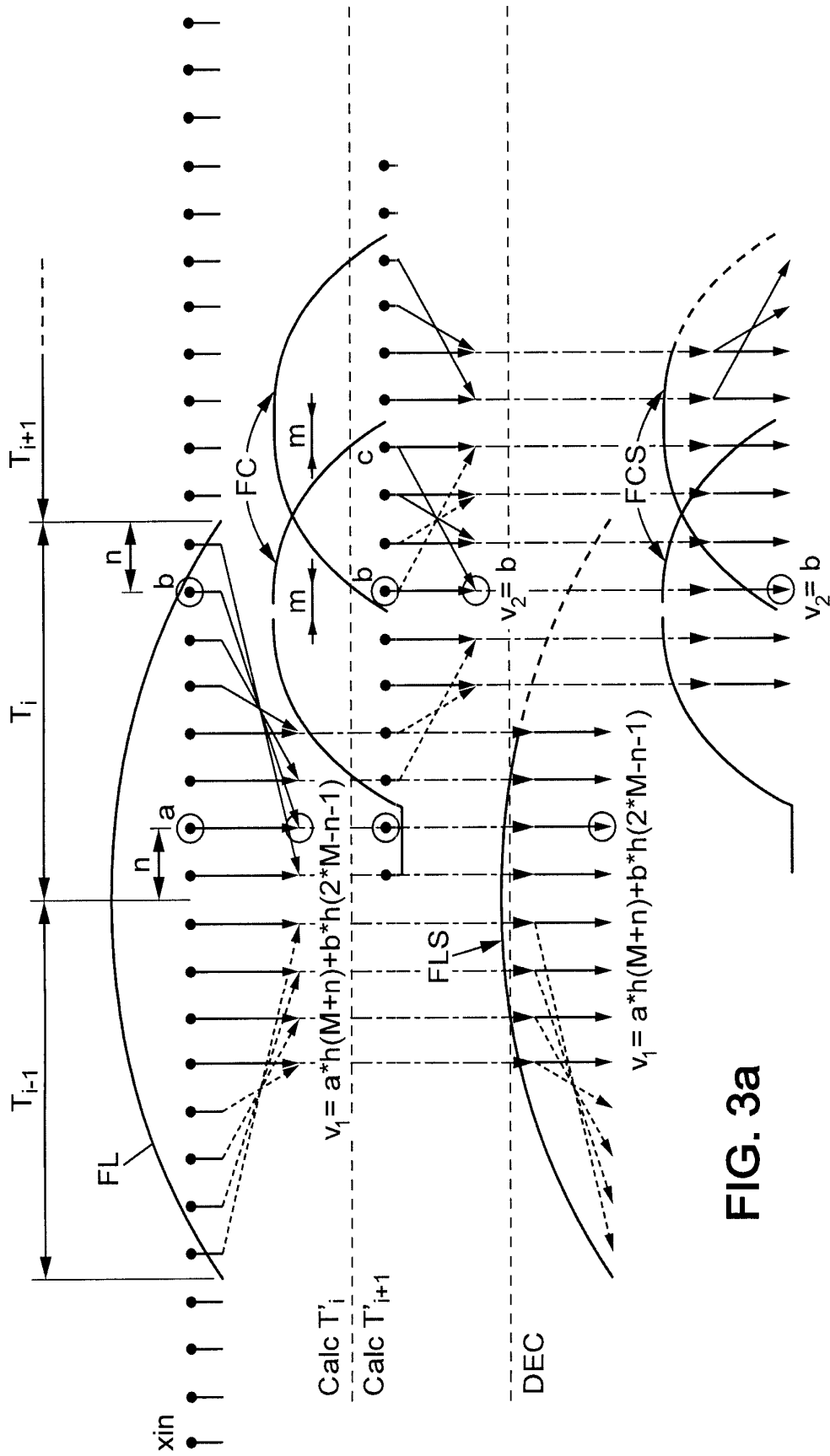


FIG. 2e



**FIG. 3a**

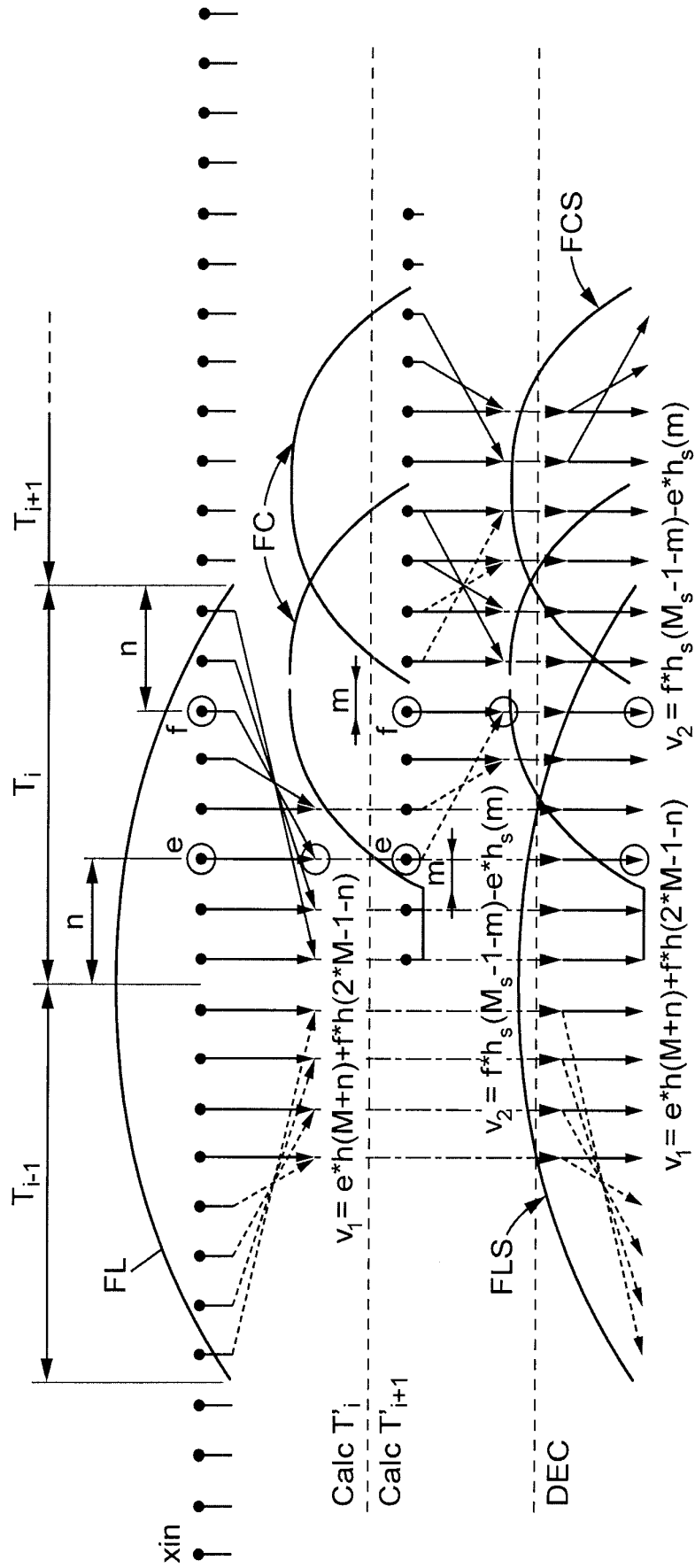


FIG. 3b

10/12

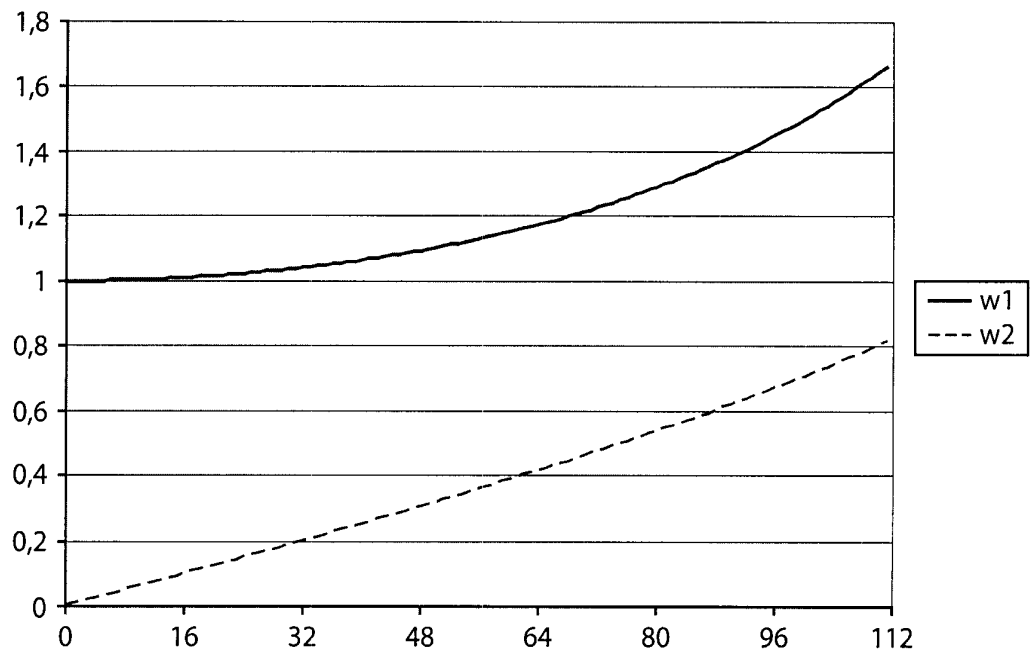


FIG. 4a

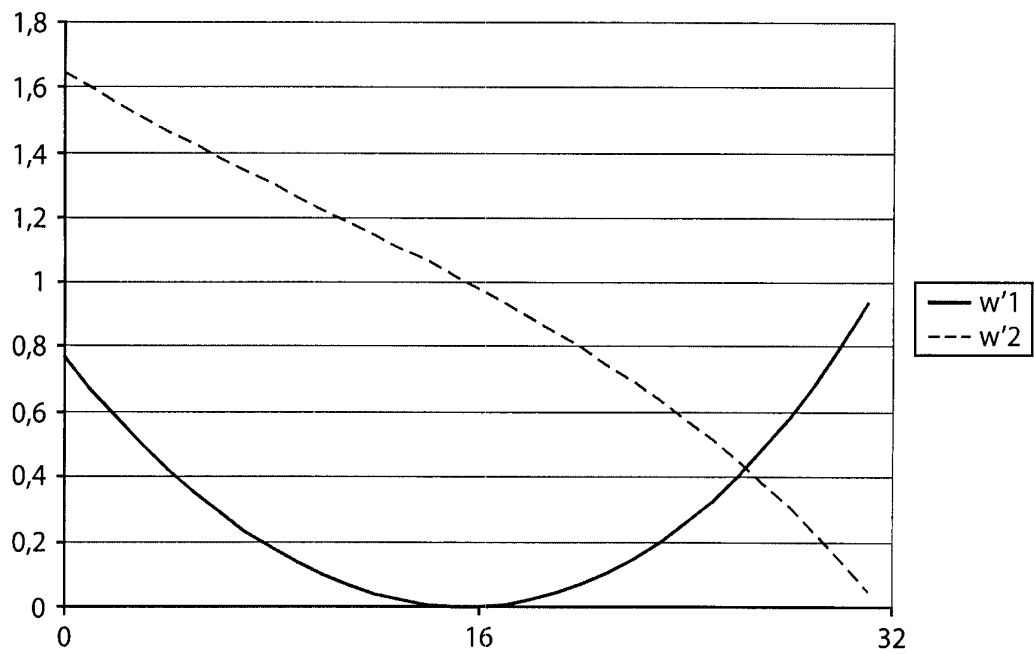


FIG. 4b

11/12

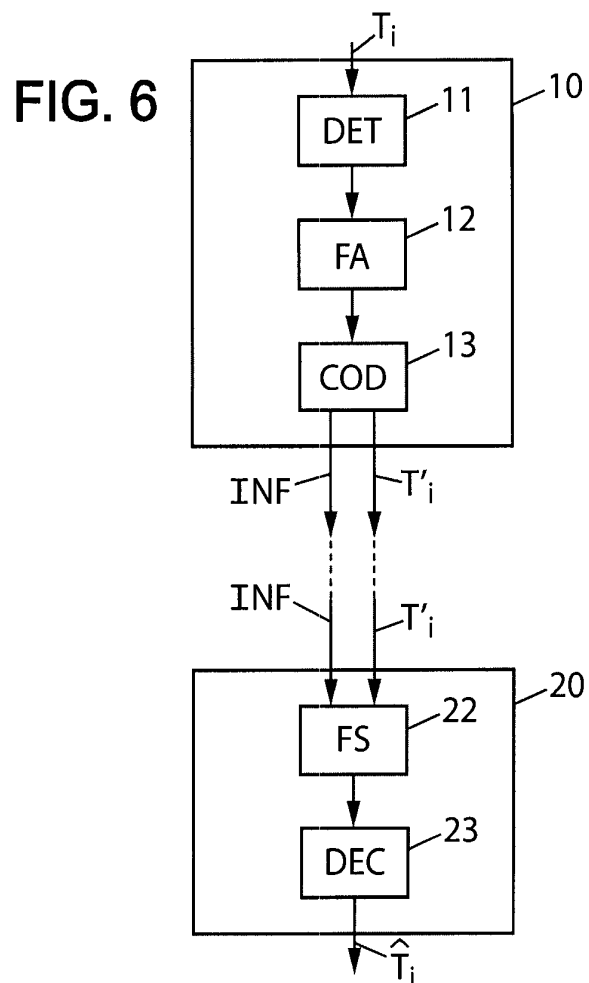
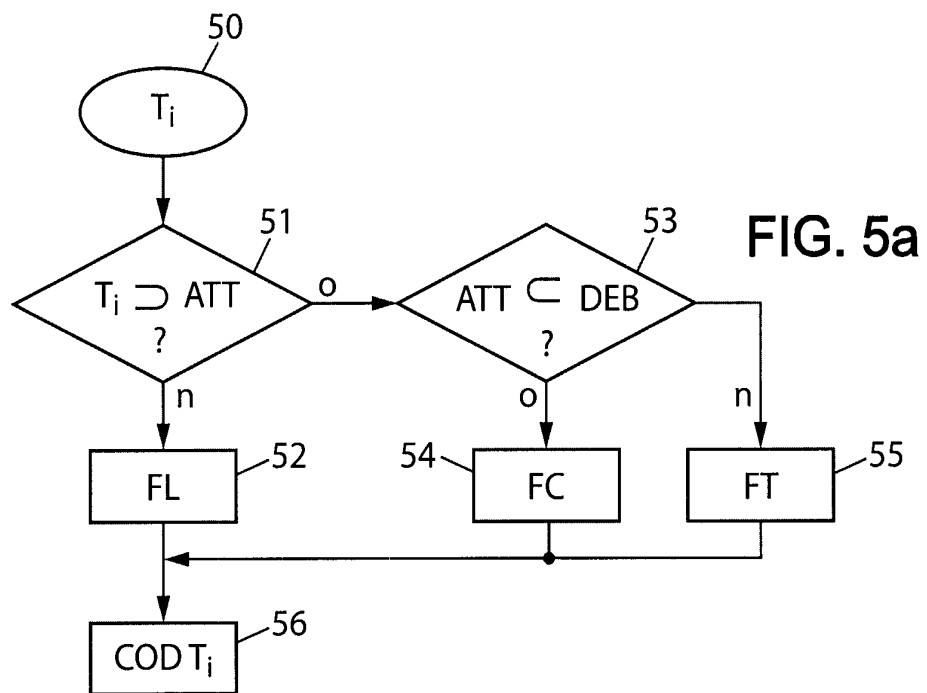


FIG. 5b

