

DIXIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

437



NICE du 20 au 24 MAI 1985

UN ALGORITHME POUR LA SYNTHÈSE DES FILTRES NUMÉRIQUES BIDI-
MENSIONNELS RIF A PHASE LINÉAIRE ET A COEFFICIENTS
DE LONGUEUR FINIE

Acyl Benslimane

CCETT Rue du Clos Courtel 35510 CESSON-SEVIGNE FRANCE

RESUME

Un algorithme de synthèse discrète de filtres numériques bidimensionnels à réponse impulsionnelle finie et à phase linéaire est proposé. Cet algorithme, efficace dans le cas de filtres comportant peu de coefficients, est basé sur un principe de recherche locale. Il s'inspire des méthodes combinatoires de type séparation et évaluation progressive et utilise un algorithme de synthèse continue optimale au sens de la norme de Tchebycheff : la programmation linéaire.

SUMMARY

An algorithm for the discrete synthesis of two-dimensional digital filters with finite impulse response and linear phase is proposed. Particularly efficient for filters with few coefficients, this algorithm is based on a principle of local search. It is derived from combinatorial methods of branch and bound type and uses an optimal algorithm in the Tchebycheff sense for the continuous synthesis : linear programming.



UN ALGORITHME POUR LA SYNTHÈSE DES FILTRES NUMÉRIQUES BIDIMENSIONNELS
RIF A PHASE LINEAIRE ET A COEFFICIENTS DE LONGUEUR FINIE

Acyl Benslimane

1 - INTRODUCTION

La synthèse des filtres numériques bidimensionnels à réponse impulsionnelle finie (RIF-2D) à phase linéaire et optimaux au sens de la norme de Tchebycheff est un sujet pour lequel de nombreuses études ont été entreprises. Parmi les techniques de synthèse fréquemment utilisées, la programmation linéaire bien qu'optimale au sens de Tchebycheff est souvent écartée en raison d'une part, des temps de calcul très importants qu'elle nécessiterait et d'autre part, en raison des problèmes de stockage dus à sa mise en oeuvre. Or, on constate qu'en appliquant un algorithme performant du simplexe à la formulation duale du problème de synthèse des filtres RIF-2D les temps de calcul obtenus sont tout à fait raisonnables, notamment pour des filtres comportant assez peu de coefficients comme l'exigent de nombreuses applications en temps réel. En outre, la plupart des méthodes exposées ne prennent pas en compte le fait que dans un grand nombre d'applications, les coefficients du filtre sont codés sur un nombre limité d'éléments binaires. Or, le simple arrondi de la solution continue peut conduire à des filtres très éloignés du filtre optimal. Ces deux constatations permettent d'envisager, pour améliorer l'arrondi de la solution continue, l'utilisation des techniques d'optimisation en nombres entiers de type séparation et évaluation progressive (SEP ou en anglais Branch and Bound) [1] avec pour méthode d'évaluation continue la programmation linéaire. Les filtres numériques RIF-2D, concernés par l'algorithme proposé, nécessitent, outre la symétrie quart plan, la symétrie par rapport à la première bissectrice, ce qui inclut un type de filtres très utilisé.

2 - FORMULATION DU PROBLEME DISCRET

Un filtre numérique bidimensionnel non récursif, de dimension $(2N+1)(2M+1)$ peut être défini par la transformée de Fourier discrète de sa réponse impulsionnelle $h(n,m)$ $n=-N, \dots, +N$; $m=-M, \dots, +M$.

$$H(f_1, f_2) = \sum_{n=-N}^N \sum_{m=-M}^M h(n,m) \exp(-j \pi (n f_1 + m f_2))$$

En imposant à la réponse impulsionnelle la relation de symétrie suivante : $h(n,m) = h(|n|, |m|)$ $n=-N, \dots, N$; $m=-M, \dots, M$ on peut écrire :

$$H(f_1, f_2) = \sum_{n=0}^N \sum_{m=0}^M a(n,m) \cos 2 \pi n f_1 \cos 2 \pi m f_2$$

avec :

$$\begin{aligned} a(0,0) &= h(0,0) \\ a(n,0) &= 2h(n,0) \quad ; \quad n=1, \dots, N \\ a(0,m) &= 2h(0,m) \quad ; \quad m=1, \dots, M \\ a(n,m) &= 4h(n,m) \quad ; \quad n=1, \dots, N \quad ; \quad m=1, \dots, M \end{aligned}$$

Si en outre on impose $N=M$ et $h(n,m) = h(m,n)$ pour $n,m=0, \dots, N$ la matrice A formée par les coefficients $a(n,m)$ devient symétrique et $H(f_1, f_2)$ peut se mettre sous la forme :

$$\begin{aligned} H(f_1, f_2) &= \sum_{n=0}^N a(n,n) \cos 2 \pi n f_1 \cos 2 \pi n f_2 \\ &+ \sum_{n=1}^N \sum_{m=0}^{n-1} a(n,m) (\cos 2 \pi n f_1 \cos 2 \pi m f_2 + \cos 2 \pi m f_1 \cos 2 \pi n f_2) \end{aligned}$$

Les coefficients $h(n,m)$ issus de la synthèse continue devant être quantifiés avec un nombre fini de b bits (signe compris), le pas de quantification q vaut $1/2^{b-1}$, et l'ensemble des valeurs possibles pour les coefficients $h(n,m)$ après quantification devient :

$$= \left\{ \begin{aligned} &h_q(n,m) \quad n,m=0, \dots, N \\ &= \{ -2^{b-1}q, \dots, -q, 0, q, \dots, (2^{b-1}-1)q \} \end{aligned} \right.$$

Ceci suppose toutefois que les valeurs des coefficients $h(n,m)$ sont comprises dans l'intervalle $[-1,1]$, ce qui est le plus souvent le cas, sinon il est aisé par un simple changement d'échelle de s'y ramener.

Cette quantification implique également que les coefficients de A ne peuvent prendre qu'un nombre fini de valeurs discrètes, déduites des relations suivantes :

$$\begin{aligned} a_q(0,0) &= h_q(0,0) \\ a_q(0,m) &= 2h_q(0,m) \quad m=1, \dots, M \\ a_q(n,0) &= 2h_q(n,0) \quad n=1, \dots, N \\ a_q(n,m) &= 4h_q(n,m) \quad n=1, \dots, N \quad ; \quad m=1, \dots, M \end{aligned}$$

Si $A_q = \{a_q(n,m) \quad n=0, \dots, N \quad ; \quad m=0, \dots, M\}$ désigne l'ensemble des valeurs discrètes que peuvent prendre les coefficients $a_q(n,m)$ on a :

$$\begin{aligned} \{a_q(0,0)\} &= \{h_q(n,m) \quad n=0, \dots, N \quad ; \quad m=0, \dots, M\} \\ \{a_q(0,m) \quad m=1, \dots, M\} &= \{a_q(n,0) \quad n=1, \dots, N\} \\ &= \{-2(2^{b-1}-1)q, \dots, -2q, 0, 2q, \dots, 2(2^{b-1}-1)q\} \\ \{a_q(n,m) \quad n=1, \dots, N \quad ; \quad m=1, \dots, M\} &= \{-4(2^{b-1}-1)q, \dots, -4q, 0, 4q, \dots, 4(2^{b-1}-1)q\} \end{aligned}$$

La synthèse au sens de Tchebycheff des filtres numériques bidimensionnels non récursifs à phase linéaire et à coefficients de longueur finie, peut se ramener au problème d'approximation suivant :

$$(P) \quad \delta_e = \min \max_{(f_1, f_2) \in F} W(f_1, f_2) |D(f_1, f_2) - P_N(f_1, f_2)|$$

δ_e désigne la valeur minimale de l'erreur d'approximation au sens de Tchebycheff pour des coefficients quantifiés.

$$P_N(f_1, f_2) = \sum_{n=0}^N a_q(n,n) \cos 2 \pi n f_1 \cos 2 \pi n f_2$$

$$+ \sum_{n=1}^N \sum_{m=0}^{n-1} a_q(n,m) (\cos 2 \pi n f_1 \cos 2 \pi m f_2 + \cos 2 \pi m f_1 \cos 2 \pi n f_2)$$

$W(f_1, f_2)$ est une fonction de pondération continue et strictement positive.

$D(f_1, f_2)$ représente la fonction continue désirée que l'on cherche à approcher.

F représente la réunion de toutes les régions du plan fréquentiel où $D(f_1, f_2)$ et $W(f_1, f_2)$ sont définies.

Le problème (P) se situe donc dans un espace discret et il n'y a pas de résultats connus permettant d'en déduire un algorithme de résolution. La méthode de résolution proposée s'inspire des techniques habituelles de l'optimisation en nombres entiers (SEP) et utilise une méthode de synthèse continue (programmation linéaire) optimale au sens de Tchebycheff.

Compte tenu du temps important sur ordinateur que nécessite cette technique, le problème (P) ne sera résolu que sur un sous-ensemble de l'espace discret considéré.



UN ALGORITHME POUR LA SYNTHÈSE DES FILTRES NUMÉRIQUES BIDIMENSIONNELS
RIF A PHASE LINÉAIRE ET A COEFFICIENTS DE LONGUEUR FINIE
Acyl Benslimane

3 - RESOLUTION DU PROBLEME CONTINU

La synthèse continue des filtres RIF bidimensionnels non récursifs et à phase linéaire peut se formuler comme le problème d'approximation suivant :

$$(Q) \delta_c = \min_A \max_{(f_1, f_2) \in F} |D(f_1, f_2) - Q_N(f_1, f_2)|$$

δ_c désigne la valeur minimale de l'erreur d'approximation.

$$Q_N(f_1, f_2) = \sum_{n=0}^N a(n, n) \cos 2\pi n f_1 \cos 2\pi n f_2 + \sum_{n=1}^N \sum_{m=0}^{n-1} a(n, m) (\cos 2\pi n f_1 \cos 2\pi m f_2 + \cos 2\pi m f_1 \cos 2\pi n f_2)$$

Dans ce cas, les coefficients de la matrice A ont une précision infinie. Les propriétés de cette approximation de type minimax sont connues. Les théorèmes classiques d'existence et de caractérisation [2] assurent qu'au moins une solution répond au problème (Q).

La méthode présentée est une des techniques habituelles de résolution de (Q). Optimale au sens de Tchebycheff, elle exploite la linéarité de Q_N par rapport aux coefficients $a(n, m)$.

On introduit une nouvelle notation afin de simplifier l'écriture de Q_N . On pose :

$$i = (n(n+1))/2 + m + 1, \quad n=0, \dots, N; \quad m=0, \dots, n$$

$$g_i(f_1, f_2) = \cos 2\pi n f_1 \cos 2\pi m f_2 + \cos 2\pi m f_1 \cos 2\pi n f_2$$

$$R = ((N+1)(N+2))/2$$

Ainsi on peut écrire :

$$Q_N(f_1, f_2) = \sum_{i=1}^R b(i) g_i(f_1, f_2) \quad \text{avec}$$

$$b(i) = 2a(n, m) \quad \text{si } n=m$$

$$b(i) = a(n, m) \quad \text{sinon}$$

Si on note :

$$E = \max_{(f_1, f_2) \in F} W(f_1, f_2) |D(f_1, f_2) - Q_N(f_1, f_2)|$$

pour tout $(f_1, f_2) \in F$ on a :

$$Q_N(f_1, f_2) - E/W(f_1, f_2) \leq D(f_1, f_2)$$

$$-Q_N(f_1, f_2) - E/W(f_1, f_2) \leq -D(f_1, f_2)$$

Le problème (Q) est alors équivalent à minimiser E sous les contraintes :

E positif

$$\sum_{i=1}^R b(i) g_i(f_1, f_2) - E/W(f_1, f_2) \leq D(f_1, f_2)$$

$$-\sum_{i=1}^R b(i) g_i(f_1, f_2) - E/W(f_1, f_2) \leq -D(f_1, f_2)$$

Après discrétisation de F, (Q) devient un problème de minimisation avec un nombre fini de contraintes qui peut donc se résoudre par un algorithme de programmation linéaire. Toutefois, ce problème linéaire contient un nombre très important de contraintes par rapport aux R+1 variables ($b(1), \dots, b(R), E$). Il est donc plus opportun, aussi bien pour des raisons de temps de calcul que d'encombrement mémoire d'en obtenir une solution en traitant le dual de (Q). La solution primale ($b(1), \dots, b(R), E$) s'obtient alors à partir des coûts réduits du dual [1].

Il est à noter l'importance de la technique de discrétisation de F. Si un ensemble discret suffisamment dense dans F fournit une solution pratiquement identique à l'optimum recherché, il semble très efficace d'y inclure des points de la frontière de F (fig.1). En effet on constate a posteriori que l'erreur maximale d'approximation de D est très souvent atteinte en des points frontières de F.

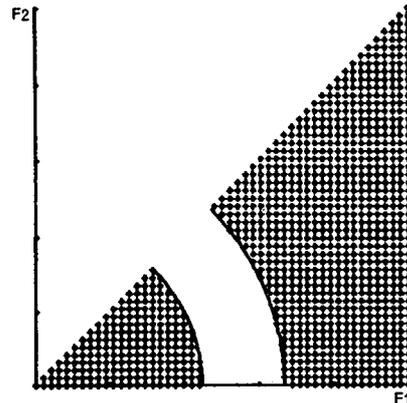


Fig1 - Exemple de discrétisation utilisée pour la synthèse d'un filtre passe-bas circulaire.

Les temps de calcul obtenus (voir table 1) sont très inférieurs à ceux attribués à la programmation linéaire dans la littérature [3], [4]. Une explication possible, outre l'application de l'algorithme dual, peut être le fait que l'on utilise la méthode révisée du simplexe.

4 - PRESENTATION DE L'ALGORITHME DE SYNTHÈSE DISCRÈTE

L'algorithme de synthèse discrète proposé est un algorithme d'énumération où un sous-ensemble des solutions possibles est décrit sous forme d'un arbre dont on examine chaque noeud. Il applique la règle suivante :

La matrice A des coefficients du filtre solution du problème continu est quantifiée ligne par ligne, l'optimisation des coefficients restant à chaque étape s'effectue en continu par la méthode de programmation linéaire. Seule la partie triangulaire inférieure de A est considérée car on a supposé que $a(n, m) = a(m, n)$ avec $N=M$. Dans ce cas la quantification d'une ligne consiste à définir un voisinage autour de l'arrondi de ses coefficients de la manière suivante :

Soit $a_q(n, m)$ $m=0, \dots, n$ les valeurs arrondies sur b bits des coefficients de la ligne n de A. Si on note par $a_*(n, m)$ les valeurs entières associées aux valeurs discrètes $a_q(n, m)$ il vient que :

$$a_*(n, m) = a_q(n, m)/q \quad \text{pour } m=0, \dots, n$$

n étant fixé on définit le voisinage V_n d'ordre l des $a_*(n, m)$ associé à la ligne n, comme l'ensemble des suites de n+1 valeurs entières dont chaque terme $a_e(m)$ s'écarte au plus de une unité par rapport à $a_*(n, m)$.

$$V_n = \{ (a_e(0), \dots, a_e(n)) \text{ tel que } a_e(m) = a_*(n, m) + k, \quad k = -1, 0, 1; \quad m = 0, \dots, n \}$$

on a $\text{Card} V_n = 3^{n+1}$, $n = 0, \dots, N$



Un voisinage V_n va constituer un niveau de l'arbre.

Afin de reposer à chaque noeud un nouveau problème d'approximation, on sépare le polynôme $Q_N(f_1, f_2)$ dans l'ordre décroissant des indices de lignes. Ainsi à un niveau l de l'arbre, $l=1, \dots, N+1$, on fixe successivement les $n+1$ valeurs de $a_q(n, m)$ à chacune des 3^{n+1} suites de V_n . A chaque noeud du niveau l , l différent de $N+1$, le problème à résoudre est de déterminer δ_l tel que :

$$\delta_l = \min_{A} \max_{(f_1, f_2) \in F} |D_l(f_1, f_2) - Q_{N-l}(f_1, f_2)|$$

avec

$$D_l(f_1, f_2) = D(f_1, f_2) - \sum_{n=N-l+1}^N q_a(n) \cos 2\pi n f_1 \cos 2\pi n f_2$$

$$- \sum_{n=N-l+1}^N \sum_{m=0}^{n-1} q_a(m) \frac{\cos 2\pi n f_1 \cos 2\pi m f_2 + \cos 2\pi m f_1 \cos 2\pi n f_2}{\cos 2\pi m f_1 \cos 2\pi n f_2}$$

et

$$Q_{N-l}(f_1, f_2) = \sum_{n=0}^{N-l} a(n, n) \cos 2\pi n f_1 \cos 2\pi n f_2$$

+

$$\sum_{n=1}^{N-l} \sum_{m=0}^{n-1} a(n, m) \frac{\cos 2\pi n f_1 \cos 2\pi m f_2 + \cos 2\pi m f_1 \cos 2\pi n f_2}{\cos 2\pi m f_1 \cos 2\pi n f_2}$$

Il est à remarquer qu'au dernier niveau, l'erreur δ_{N+1} se calcule simplement sans faire appel à la programmation linéaire. Les relations précédentes montrent également que l'erreur minimale pour chaque niveau croît lorsque l varie de 1 à $N+1$. L'erreur d'arrondi δ_a étant le point de départ de l'algorithme si, à un niveau l , δ_l est supérieur à δ_a il devient inutile d'examiner les branches qui en découlent. Une première solution réalisable est obtenue lorsqu'une terminaison de l'arbre vérifie δ_{N+1} inférieur à δ_a . Cette

nouvelle valeur appelée erreur optimale locale courante et notée $\delta_{o,c}$ se substitue à δ_a dans la recherche de l'optimum local. Lorsque tous les noeuds ont été explorés $\delta_{o,c}$ représente alors l'erreur optimale dans un voisinage de l'arrondi de la solution continue.

La stratégie de cheminement choisie est dite "profondeur d'abord", elle consiste à sélectionner à chaque niveau un chemin unique pour passer au niveau suivant, le critère de choix étant celui de l'erreur minimale. Cette stratégie présente l'avantage d'améliorer rapidement le point de départ. Dès qu'une solution est obtenue, il faut remonter au niveau le plus proche qui comporte des chemins laissés en suspens, en prenant soin au préalable d'éliminer le chemin qui vient d'être parcouru. L'algorithme se termine quand au premier niveau tous les chemins ont été explorés explicitement ou implicitement.

Un obstacle majeur à la mise en oeuvre de cet algorithme est le temps de calcul énorme que nécessiterait l'usage à chaque noeud de la programmation linéaire, si celle-ci s'appliquait avec une densité élevée de points dans F . Or une grande précision indispensable pour un problème pris séparément

n'est pas nécessaire pour les étapes intermédiaires de l'algorithme SEP. On diminue donc le nombre de points de discrétisation de F (fig2) d'autant plus que l'on ne risque pas de laisser échapper de solution, δ_l étant ainsi sous-estimée. Cette méthode ne s'applique pas au dernier niveau où l'erreur est calculée sur une grille non réduite.

Une autre manière de diminuer le temps de calcul consiste à exploiter la formulation duale de la synthèse continue. Dans ce cas, l'erreur interne de l'algorithme du simplexe croît à chacune de ses itérations. Par conséquent, dès que pour une itération du simplexe, l'erreur devient supérieure à $\delta_{o,c}$, le calcul se termine pour le noeud considéré auquel on attribue une valeur arbitrairement grande.

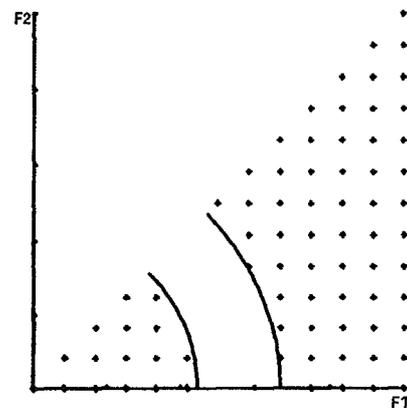


Fig2 - Exemple de discrétisation avec une grille réduite.

5 - EXEMPLES

Remarque préliminaire :

Les indications de temps de calcul qui sont données correspondent à l'utilisation d'un ordinateur DPS7, qui se situe dans la moyenne pour la rapidité des traitements.

Pour illustrer l'algorithme discret, trois exemples sont proposés, choisis parmi ceux dont les résultats en continu sont déjà parus dans la littérature [3].

Il s'agit de filtres passe-bas circulaires définis de la manière suivante :

$$D(f_1, f_2) = 1 ; W(f_1, f_2) = 1 \text{ si } (f_1^2 + f_2^2)^{1/2} \leq R_1$$

$$D(f_1, f_2) = 0 ; W(f_1, f_2) = \lambda \text{ si } (f_1^2 + f_2^2)^{1/2} \geq R_2$$

Les coefficients $h(n, m)$ de chacun de ces filtres sont codés sur 8 bits dont le bit de signe.

La table 1 résume les résultats obtenus pour ces trois exemples par la synthèse continue et la synthèse discrète.

La table 2 fournit, pour chaque exemple, les coefficients $h_q(n, m)$ $n=0, \dots, N$; $m=0, \dots, n$ multipliés par 2 et associés aux deux types de solutions : arrondie (typel) et optimisée (type2).

UN ALGORITHME POUR LA SYNTHÈSE DES FILTRES NUMÉRIQUES BIDIMENSIONNELS
RIF A PHASE LINÉAIRE ET A COEFFICIENTS DE LONGUEUR FINIE
Acyl Benslimane

On note par :

- δ_o l'erreur de la solution continue dans la bande passante
- δ_s l'erreur de la solution continue dans la bande atténuée
- δ_a l'erreur pondérée de la solution arrondie
- δ_o l'erreur pondérée de la solution optimisée
- t_c le temps de calcul de la synthèse continue
- t_d le temps de calcul de la synthèse discrète

La figure 3 représente la solution optimisée de l'exemple 1.

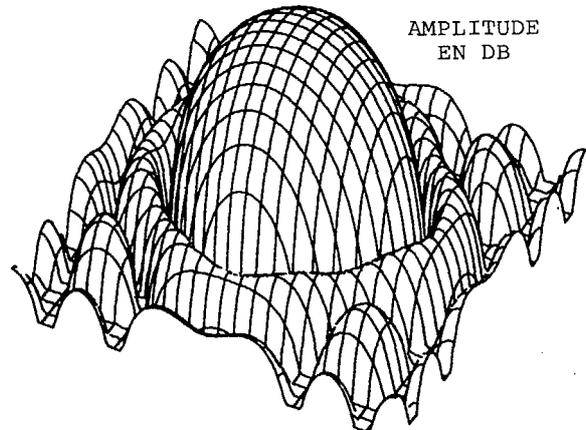


Fig3 - Réponse en fréquence d'un filtre passe-bas circulaire à coefficients quantifiés optimisés.

- TABLE 1 -

	EXEMPLE 1	EXEMPLE 2	EXEMPLE 3
2N+1	7	7	9
R1	0.2	0.2222...	0.2222...
R2	0.3	0.3333...	0.3333...
λ	10	1	10
δ_p	0.045	0.123	0.235
δ_s	0.0045	0.123	0.0235
δ_a	0.899	0.1866	0.8
δ_o	0.52	0.137	0.39
t_c	16 sec	30 sec	54 sec
t_d	137 sec	78 sec	24 mn

6 - CONCLUSION

L'algorithme de synthèse discrète proposé montre son efficacité dans le cas des filtres numériques RIF-2D comportant peu de coefficients. Ses performances en temps de calcul remettent d'actualité la programmation linéaire pour la synthèse des filtres numériques RIF-2D, qui d'une part, se montre efficace pour résoudre un problème de synthèse continue et qui, d'autre part, s'intègre dans un algorithme combinatoire.

Du point de vue de l'amélioration de l'arrondi de la solution continue, cet algorithme de recherche locale donne de bons résultats. Une explication possible est que les coefficients d'un filtre RIF correspondent à une réponse impulsionnelle dont une solution discrète ne peut complètement s'écarter.

REFERENCES BIBLIOGRAPHIQUES

- [1] M.MINOUX "Programmation mathématique théorie et algorithmes" Collection technique et scientifique des télécommunications tome 2 -Dunod- 1983
- [2] JR.RICE "The approximation of functions" Vol.2 Addison-Wesley, Reading, Mass., 1979
- [3] JV.HU and LR.RABINER "Design techniques for two-dimensional digital Filters"IEEE Trans.Audio Electroacoust.Vol.Au 20Oct.1972
- [4] JG.FIASCONARO "Two-dimensional nonrecursive filters" in Picture Processing and Digital Filtering. Editor : TS.HUANG Springer-Verlag Berlin Heidelberg - New York

- TABLE 2 -

	EXEMPLE 1		EXEMPLE 2		EXEMPLE 3	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
	21	19	29	27	28	28
	15	15	21	23	20	20
	12	12	15	13	13	14
	7	7	3	2	4	6
	4	5	-1	-1	2	3
	0	1	-3	-3	-1	-2
	0	1	-2	-3	-2	-2
	0	0	-2	-3	-2	-2
	0	-1	0	-1	-2	-2
	0	0	0	0	0	-1
					-1	-1
					0	-1
					0	0
					0	0
ATTENUATION EN DB	20.9	25.7	14.6	17.3	21.9	28.2

