GÉNÉRATION ET DÉTECTION DE DTMF

SOMMAIRE

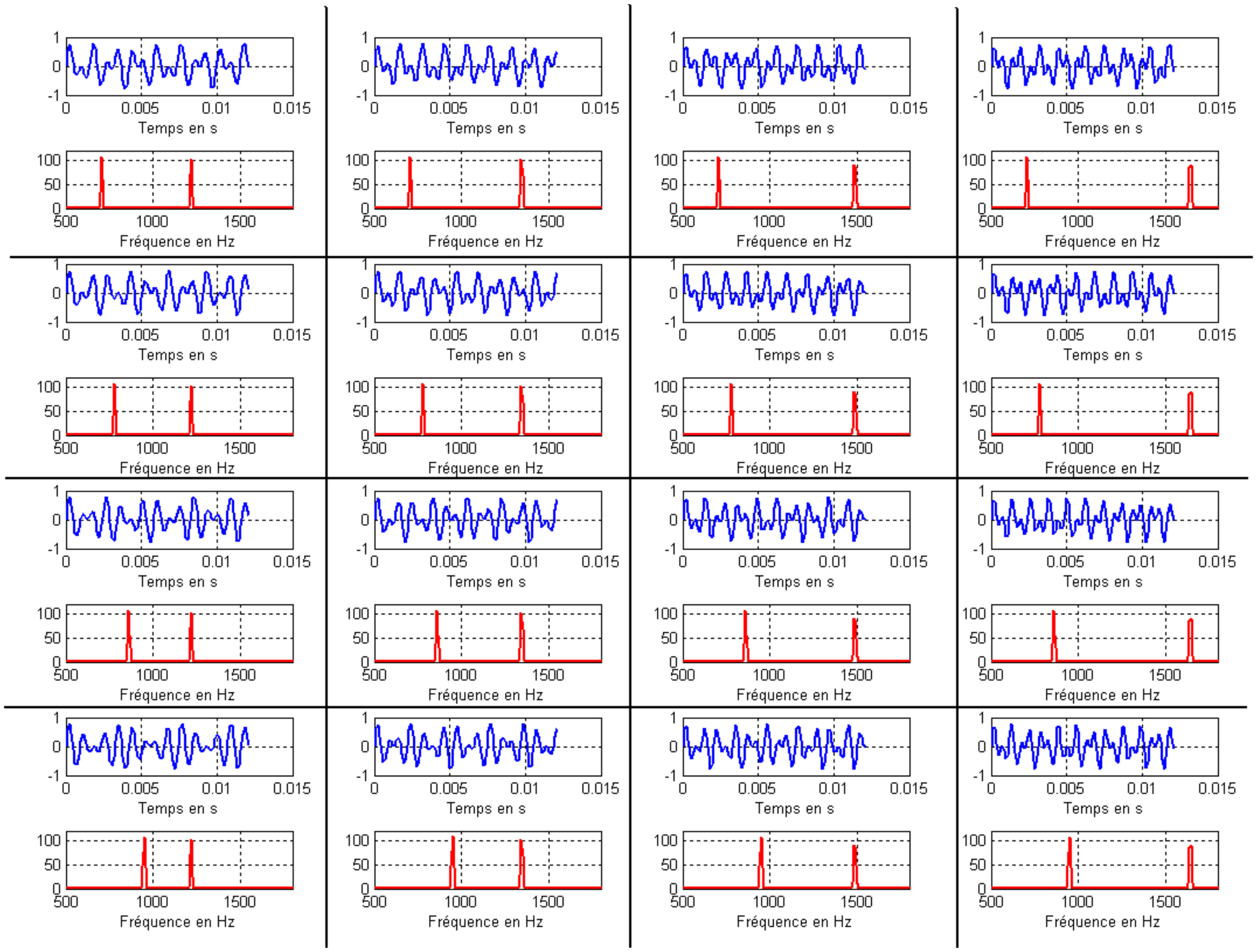
I. LE PROBLÈME

Le DTMF (Dual Tone Multi-Frequency) est utilisé en téléphonie pour le codage du clavier du téléphone.

Lors de l'établissement d'une communication téléphonique, il est nécessaire de transmettre au standard le n° demandé. Sous quelle forme cette indication est-elle transmise ? Dans les téléphones à touche modernes le clavier est vu comme une matrice et l'appui sur une touche envoie un signal qui est le mélange de deux sinusoïdes. Avec deux groupes de fréquences respectivement hautes et basses nous pouvons coder 16 touches selon le schéma suivant :

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Ceci correspond aux signaux temporels et aux spectres associés portés sur la figure suivante.



D'un point de vue pratique deux problèmes se posent :

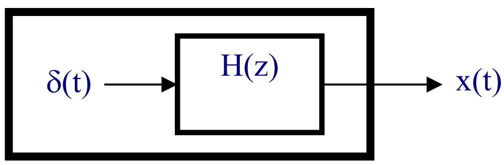
- A l'émission : comment réaliser par programmation les différentes sinusoïdes ?
- A la réception : comment les détecter ?

Pour cela nous utiliserons la technique du numérique, les signaux utilisés sont des signaux échantillonnés et la norme internationale fixe la fréquence d'échantillonnage à T_s = 8000 Hz.

II. GENERATION DE SIGNAUX DISCRETS : FILTRE GENERATEUR (OU FORMEUR)

II.1. Cas général :

Tout signal discret peut être modélisé par la réponse impulsionnelle d'un filtre (filtre formeur ou filtre générateur) selon le schéma :



Modèle du signal x(t)

Pour un signal ayant le filtre générateur de fonction de transfert $H(z) = \dfrac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$ l'équation récurrente entrée-sortie s'écrit :

$$x_k = b_0\, u_k + b_1\, u_{k-1} - a_1\, x_{k-1} - a_2\, x_{k-2}$$

où u(t) est l'entrée du système. Ici, $u(t) = \delta(t)$ c'est à dire que $u_k = 0$ pour $k \neq 0$.

Pour générer le signal x(t) présumé causal, il suffit de faire fonctionner cette équation récurrente avec des conditions initiales nulles et seul $u_0 \neq 0$. D'un point de vue implantation, cela oblige à faire systématiquement un test sur k pour attribuer la valeur u_k et u_{k-1} alors qu'elle est pratiquement toujours nulle. Pour éviter ce test il est possible d'écrire le problème autrement :

$$\text{Pour } k \geq 0 \Rightarrow x_{k+2} = -a_1\, x_{k+1} - a_2\, x_k$$

Pour débuter cet algorithme, il faut les deux conditions initiales qui se calculent avec l'équation récurrente :

- $x_0 = b_0$
- $x_1 = b_1 - b_0\, a_1$

$\Rightarrow$ tout signal peut être généré par une équation récurrente issue de son filtre formeur et un choix judicieux des conditions initiales.

La difficulté est de trouver le filtre formeur associé à un signal donné.

II.2. Cas des signaux sinusoïdaux :

Pour un signal de forme sinusoïdale le cas est très simple. Nous savons que :

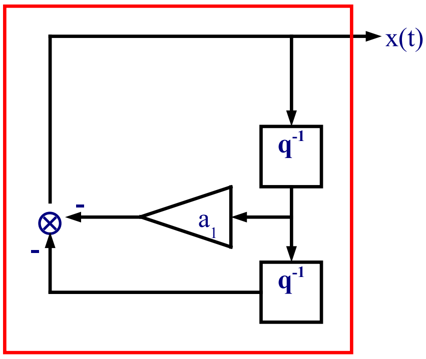
$$TZ[\cos(\omega_0 t)] = \frac{1-\cos(\omega_0 T_s)\, z^{-1}}{1 - 2\cos(\omega_0 T_s)\, z^{-1} + z^{-2}} \Rightarrow \begin{cases} b_0 = 1 \quad ; \quad b_1 = -\cos(\omega_0 T_s) \\ a_1 = -2\cos(\omega_0 T_s) \quad ; \quad a_2 = 1 \end{cases}$$

$$TZ[\sin(\omega_0 t)] = \frac{\sin(\omega_0 T_s)\, z^{-1}}{1 - 2\cos(\omega_0 T_s)\, z^{-1} + z^{-2}} \Rightarrow \begin{cases} b_0 = 0 \quad ; \quad b_1 = \sin(\omega_0 T_s) \\ a_1 = -2\cos(\omega_0 T_s) \quad ; \quad a_2 = 1 \end{cases}$$

Quelque soit la phase, sans importance dans le problème du DTMF, un signal sinusoïdal est ainsi très simple à générer par son filtre formeur.

Algorithme :

- Calcul de $a_1 = -2\cos(2\pi f_0 T_s)$
- Calcul des conditions initiales $x_0 = b_0$; $x_1 = b_1 - b_0\, a_1$
- En boucle exécuter l'équation récurrente : $x(t) = -a_1\, x(t-1) - x(t-2)$



III. DETECTION - ALGORITHME DE GOERTZEL

III.1. Détection d'une sinusoïde :

Le problème peu sembler trivial : il suffit d'un filtre sélectif à la fréquence donnée. Cependant, pour obtenir une sélectivité suffisante, ce filtre va requérir un nombre appréciable de coefficients *(par exemple, pour séparer correctement les raies 1336Hz et 1477Hz, il faut un filtre de Butterworth d'ordre au moins égal à 25, ceci à faire fonctionner pour chaque raie soit 8 fois)*.

L'autre alternative est de calculer une FFT cependant l'algorithme est assez lourd à programmer et il nous fournit beaucoup trop de renseignements puisqu'il calcule les N raies de la bande de fréquence de Shannon. Pour détecter une sinusoïde, nous n'avons besoin que d'une raie du spectre pour savoir si oui ou non elle est présente, c'est l'idée de Goertzel qui le conduit à la lumière de cette réflexion à une simplification extrême du calcul de la TFD.

III.2. Rappel sur la TFD :

Avec N points, la TFD d'un signal est définie par :

$$X_n = \sum_{k=0}^{N-1} x_k \, e^{-j\frac{2\pi kn}{N}}$$

les $\{x_k\}$ étant les N échantillons du signal ($k \in [0;N-1]$) et les $\{X_n\}$ les N échantillons de la TFD ($n \in [0;N-1]$) qui estiment la TF du signal échantillonné. L'écart fréquentiel entre deux échantillons successifs de cette TFD (précision fréquentielle) est égal à :

$$\Delta f_{TFD} = \frac{F_s}{N} = \frac{1}{N\,T_s}$$

III.3. Algorithme de Goertzel :

Nous recherchons une sinusoïde, nous avons besoin de calculer uniquement la composante X_n telle que :

$$n = \frac{f_0}{\Delta f_{TFD}} = \frac{f_0}{F_s} N$$

Formulation à partir de la TFD :

En reprenant les notations de la TFD nous pouvons réécrire la formule de la TFD :

$$\left.\begin{array}{l} X_n = \sum_{k=0}^{N-1} x_k \, e^{-j\frac{2\pi kn}{N}} \\ W_N = e^{-j\frac{2\pi}{N}} \end{array}\right\} \Rightarrow \quad X_n = \sum_{k=0}^{N-1} x_k \, W_N{}^{kn}$$

$$\left.\begin{array}{l} X_n = \sum_{k=0}^{N-1} x_k \, W_N{}^{kn} \\ W_N{}^{-nN} = 1 \end{array}\right\} \Rightarrow \quad X_n = \sum_{k=0}^{N-1} x_k \, W_N{}^{-n(N-k)}$$

Solution récurrente :

Posons : $Q_m = \sum_{k=0}^{m} x_k \, W_N^{-n(m+1-k)}$ la solution recherchée est : $X_n = Q_{N-1}$

Nous pouvons écrire : $$Q_m = \underbrace{\left[\sum_{k=0}^{m-1} x_k \, W_N^{-n(m-k)}\right]}_{Q_{m-1}} W_N^{-n} + x_m W_N^{-n}$$

Soit la relation de récurrence :

$$(1 - q^{-1} W_N^{-n})\, Q_m = x_m W_N^{-n}$$

Les Q_m se déduisent ainsi facilement des x_m par un filtrage grâce à une cellule du premier ordre. L'inconvénient est que le pôle associé est W_N^{-n} c'est à dire qu'il est complexe. Cela nécessite dans les calculs des coefficients complexes et peu de processeurs permettent aisément cette manipulation. Une petite manipulation va remédier à ce souci.

Un signal intermédiaire - le résonateur :

En remarquant que : $\overline{W_N^{-n}} = W_N^{n}$ nous écrivons :

$$(1 - q^{-1} W_N^{n})(1 - q^{-1} W_N^{-n})\, Q_m = (1 - q^{-1} W_N^{n})\, x_m W_N^{-n}$$
$$(1 - (W_N^{n} + W_N^{-n}) q^{-1} + q^{-2})\, Q_m = (1 - 2\cos(\frac{2\pi n}{N}) q^{-1} + q^{-2})\, Q_m = (1 - q^{-1} W_N^{n})\, x_m W_N^{-n}$$

En faisant intervenir un signal intermédiaire V(t) :

$$V_m = \frac{1}{(1 - 2\cos(\frac{2\pi n}{N}) q^{-1} + q^{-2})} x_m$$
$$Q_m = (W_N^{-n} - q^{-1})\, V_m$$

Le signal filtré étant réel, les coefficients du filtre étant eux aussi réels $\Rightarrow$ le signal V_m est un signal réel. Ce signal V(t) est un signal échantillonné obtenu par filtrage de x(t) par le filtre de fonction de transfert H(z):

$$V(z) = H(z)\, X(z) \qquad H(z) = \frac{z^2}{z^2 - 2\cos(\frac{2\pi n}{N}) z^1 + 1}$$

Ce filtre possède deux pôles complexes conjugués $z = e^{\pm j \frac{2\pi n}{N}}$, deux pôles sur le cercle unité d'où la résonance pour la fréquence $f = \frac{n}{N\, T_s} = \frac{n}{N} F_s = f_0$. C'est la fréquence de la sinusoïde recherchée.

Le résultat final :

Le résultat qui nous intéresse est Q_{N-1}. Le résonateur fonctionne bien à la fréquence F_s = 8000Hz (T_s = 125µs) alors que le résultat qui nous intéresse ne se calcule que tous les $N.T_s$:

$$Q_{N-1} = (W_N^{-n} - q^{-1})\, V_{N-1} = W_N^{-n}\, V_{N-1} - V_{N-2}$$

Nous nous intéressons au module ou module au carré :

$$|Q_{N-1}|^2 = (W_N^{-n} V_{N-1} - V_{N-2})(W_N^{n} V_{N-1} - V_{N-2}) = V_{N-1}^2 + V_{N-2}^2 - 2\cos(\frac{2\pi n}{N}) V_{N-1}V_{N-2}$$

$$\boxed{|X_n|^2 = V_{N-1}^2 + V_{N-2}^2 + a_1 V_{N-1}V_{N-2}}$$

III.4. Implémentation :

En permanence tourne l'algorithme qui reconstruit le signal V(t). C'est un filtrage du second ordre des données reçues avec une équation de récurrence telle que :

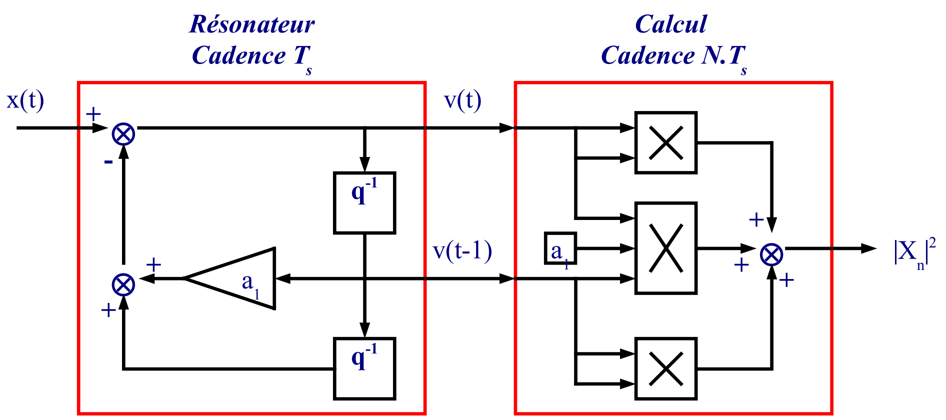
$$V_m = x_m - a_1 V_{m-1} - V_{m-2}$$
$$a_1 = -2\cos(\frac{2\pi n}{N})$$

Au N^ième^ échantillonnage on peut déterminer l'amplitude de la raie recherchée soit :

$$|X_n|^2 = V_{N-1}^2 + V_{N-2}^2 + a_1 V_{N-1}V_{N-2}$$

si la raie n'est pas présente, cette valeur sera nulle (en pratique faible).

Pour les échantillons suivants, on réinitialise l'algorithme pour reprendre le calcul tant qu'il y a un signal : l'algorithme est réinitialisé tous les N T_s.



En pratique, on peut aussi faire fonctionner l'ensemble à la période d'échantillonnage T_s, le résultat du calcul n'étant pris en compte que tous les instants $N.T_s$.

Norme internationale :

Dans la pratique la norme utilisée est celle du Bell Laboratory soit : F_s = 8000 Hz ; N = 205 (

$\Delta f_{TFD} = \frac{F_s}{N} \approx 39$ Hz ce qui est suffisant pour séparer les raies du codage DTMF. Le tableau des valeurs de n à utiliser est le suivant (la valeur pratique est l'entier immédiatement le plus proche) :

f_0 (en Hz)	697	770	852	941	1209	1336	1477	1633
n	18	20	22	24	31	34	38	42

Le processeur réalisant l'algorithme dispose d'au plus 125 µs pour cela, pour décider de la présence ou non d'une raie il faudra au moins 26ms. Les signaux DTMF durant environ 100 ms, le calcul est réalisé 4 à 5 fois lors de la réception d'un signal.

IV. MANIPULATIONS

IV.1. Génération de signaux sinusoïdaux :

Programmation de l'équation de récurrence :

En choisissant une fréquence d'échantillonnage de 8000 Hz pour les fréquences f_0 = 440Hz, 523Hz et 1046Hz calculer et tester des signaux de durée 0,25s. Pour cela on effectuera les opérations suivantes :

- Calcul des coefficients b_0, b_1 et a_1 pour générer un cosinus d'amplitude 0,4
- Calcul des échantillons du cosinus
- Calcul des coefficients b_0, b_1 et a_1 pour générer un sinus d'amplitude 0,4
- Calcul des échantillons du sinus
- Représenter les deux signaux sur le même graphe, les comparer
- Avec la fonction **sound** les écouter grâce à la carte son

Programmation graphique :

En utilisant Simulink implanter l'algorithme de génération d'un sinus. Les blocs "retard" nécessaires sont *Signal Processing Blockset > Signal Operations > **Delay***. Les paramètres des différents blocs seront calculés par le fichier script précédent. Visualiser le signal (*Signal Processing Blockset > Signal Processing Sinks > **Time Scope***) et l'écouter (*Signal Processing Blockset > Signal Processing Sinks > **To Wave Device***).

En réalité ce travail est déjà programmé sous forme complète dans le bloc Simulink : *Signal Processing Blockset > Filtering > Filter Designs > **Digital Filter***. Ajouter ce bloc et comparer son fonctionnement avec le bloc précédent.

Signaux du DTMF :

Calcul des paramètres :

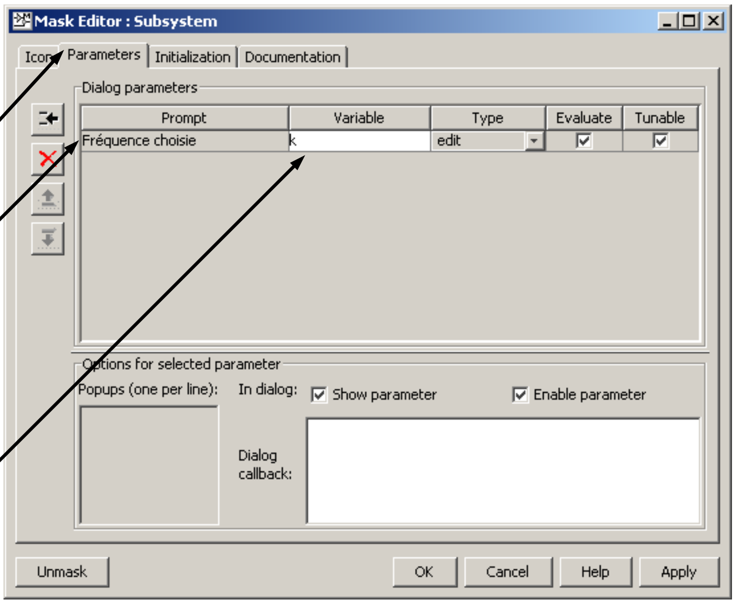
Avec un fichier script, calculer pour les 8 fréquences du DTMF les conditions initiales x_0; x_1 et le coefficient a_1. On utilisera pour cela des vecteurs de dimension 8, les fréquences du DTMF étant traditionnellement classées de la plus basse (697Hz $\rightarrow$ index 1) vers la plus haute (1633Hz $\rightarrow$ index 8). L'amplitude des oscillations obtenues sera fixée à 0,2.

Réalisation d'un bloc élémentaire :

Cet exercice sera prétexte à utiliser une fonctionnalité de Simulink : créer des sous-systèmes et leur attribuer un masque de dialogue :

- Par copier/coller récupérer l'un des schémas de génération d'une sinusoïde de l'étape précédente.
- Lui attribuer le coefficient $a_1(k)$ et les conditions initiales x0(k) et x1(k).
- Avec la souris, sélectionner l'ensemble des éléments du système puis créer un sous système : menu *Edit > Create Subsystem*.

- Sur le système nouvellement créé avec la commande *Edit > Mask Subsystem* nous pouvons créer un interface graphique avec le sous-programme comme l'indique la fenêtre suivante :

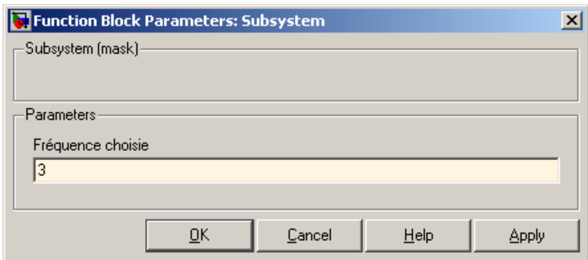


Définition des paramètres

Nom de la variable : texte destiné à rappeler quelle doit être l'entrée.

Variable : porter ici le nom utilisé dans le sous-système. Pour nous ce sera k, l'index de la fréquence.

Cela suffit pour créer un masque simple associé à un sous-système. Une fois ceci sauvegardé, un clic sur le bloc ouvrira une fenêtre de dialogue invitant à préciser le n° de la fréquence utilisée. Ensuite, le bloc ira rechercher dans l'espace de travail les variables $a_1(k)$, x0(k) et x1(k) désirées ce qui implique d'exécuter au préalable le fichier script qui les calcule.



Un signal DTMF :

Dupliquer le bloc réalisé, puis sommer la sortie des deux blocs. Initialiser chaque bloc pour produire une fréquence différente et observer les signaux obtenus d'un point de vue temporel (*Signal Processing Blockset > Signal Processing Sinks > **Time Scope***) et fréquentiel (*Signal Processing Blockset > Signal Processing Sinks > **Spectrum Scope***).

IV.2. Détection des fréquences DTMF :

Programmation de l'équation de récurrence :

Avec un nouveau fichier script on implémentera l'algorithme de Goertzel selon les grandes lignes suivantes :

- Fs = 8000 Hz, N = 205
- Calculer les valeurs de n du tableau et en déduire le vecteur a1(k) des valeurs à utiliser dans le résonateur (k → n° de la fréquence DTMF).
- Créer un signal superposition de deux sinusoïdes du DTMF
- Implémenter l'algorithme de calcul de Goertzel (*Remarque : pour une programmation simplifiée, on pourra utiliser un tableau Vm(3,8) → 1 ligne par résonateur, la colonne correspondant aux 3 valeurs nécessaires à la récursivité. Il peut être intéressant d'utiliser les produits pointés*).

- Afficher graphiquement le résultat de la simulation. Changer les fréquences du signal pour vérifier le fonctionnement de l'algorithme.

Réalisation avec Simulink :

On utilisera les a1(k) calculés par le fichier précédent.

Un résonateur

Avec Simulink, réaliser l'algorithme de calcul. Pour se simplifier l'existence, on fera travailler à la même fréquence le résonateur et le calcul de $|X(n)|^2$. Il est important de réinitialiser l'algorithme tous les N coups et pour cela, il faut utiliser les retards du résonateur avec l'option Reset, cette entrée sera réinitialisée par un générateur d'impulsions de cadence $N.T_s$ (*Simulink > Sources > Pulse Generator)*.

Pour tester son fonctionnement, on utilisera un générateur BF appliqué sur l'entrée ligne de l'ordinateur. Faire varier la fréquence et observer la sortie.

Détecteur complet :

- Dupliquer 7 fois le bloc, attribuer à chacun une fréquence du DTMF (*il peut être astucieux de les mettre dans l'ordre*).
- Utiliser comme signal d'entrée celui fournit par le fichier NumeroBE.wav
- Visualiser sur un même scope le signal d'entrée et les 8 sorties des résonateurs
- En déduire le signal le n° de téléphone enregistré dans le fichier .wav

2ème Partie : Implémentation sur un micro-contrôler ou un DSP de votre choix

Après cette première partie de simulation, il est possible de générer à partir de Simulink un code implémentable sur une cible embarqué. L'objectif de cette seconde partie est de choisir un système emabrqué et de réaliser la génération et la détection de touches DTMF.