

Décodage manuel d'un signal FSK bruité.

E. Montseny
(emmanuel@montseny.pro)

2009-2010

La transmission de signaux dans des canaux de transmission nécessite d'adapter le signal, en terme de spectre, aux propriétés physiques du canal, pour assurer la qualité de la transmission et le débit d'information que l'on pourra transmettre ; c'est la raison pour laquelle on **module** les signaux préalablement à leur transmission. Parmi les plus classiques, on peut citer la modulation d'amplitude (AM), de phase (PM) ou de fréquence (FM).

La modulation par déplacement de fréquence (Frequency Shift Keying, FSK) est une technique de modulation numérique basique très utilisée. Celle-ci consiste à coder les symboles 0 et 1 par des fréquences distinctes f_0 ou f_1 , le signal modulé étant alors une succession de périodes élémentaires sinusoïdales de fréquences f_0 ou f_1 (cf. Figure 3).

Dans cet exercice, on vous propose de remplacer un récepteur-décodeur qui recevrait un signal FSK bruité (bruit dû à son transport par exemple). Vous devrez donc mettre à profit vos (maigres !) connaissances en traitement du signal pour filtrer le signal et décoder le message modulé par déplacement de fréquences.

1 Préliminaires

1. Lancez le programme *generer_signal*. Il vous sera demandé combien d'octets (i.e. de caractères) vous souhaitez utiliser pour le code ; celui-ci est alors généré aléatoirement parmi les caractères 0-9,a-z,A-Z, converti en signal FSK, bruité, et enregistré dans un fichier.
N.B. Dans un premier temps, prenez 1 ou 2 octets ; vous pourrez refaire l'opération avec plus d'octets lorsque votre programme sera au point.
2. Ouvrez le fichier *filtrage_signal*, qui sera votre fichier de travail. Exécutez les instructions d'initialisation pour charger les données, puis exécutez les instructions pour tracer le signal. Sauriez-vous retrouver le code dans ce signal ?
N.B. Pour exécuter certaines instructions pas-à-pas, sélectionnez-les et faites F9. Pour exécuter d'un coup toutes les instructions d'un bloc d'instructions (en surbrillance jaune), faite Ctrl+enter.

2 Filtrage du signal

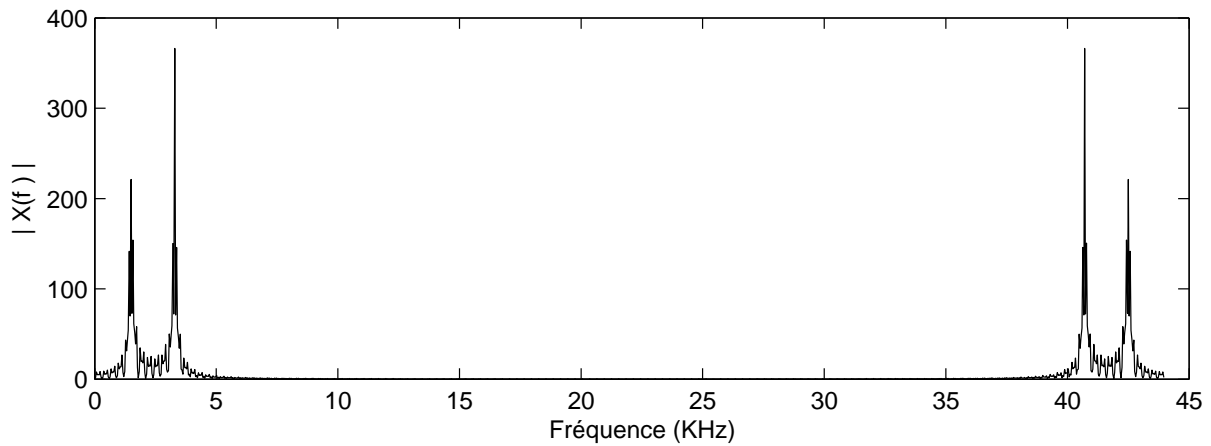
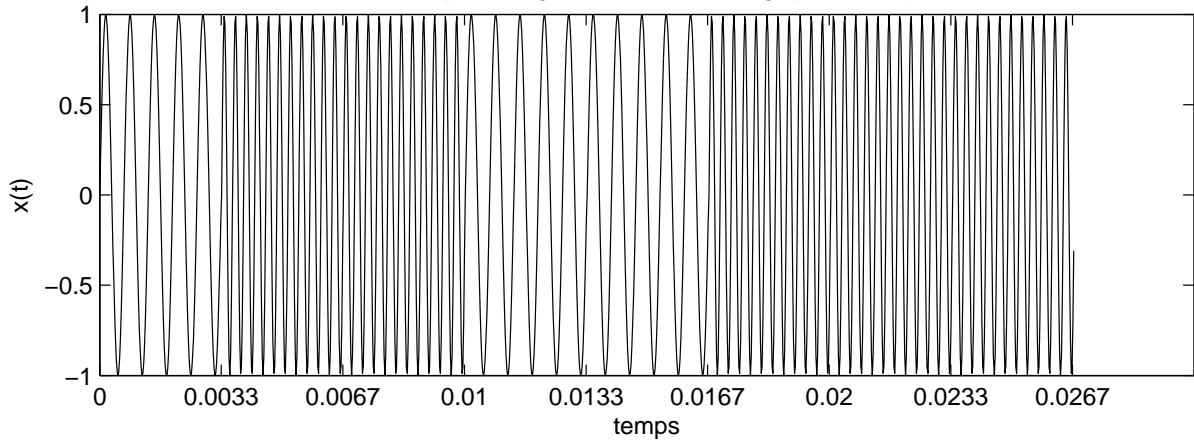
Le but est désormais de "nettoyer" le signal (partie *Traitement (fréquentiel) du signal*).

1. Tracez le periodogramme du signal (ne pas oublier l'axe des fréquences), et déterminez les fréquences f_0 et f_1 utilisées pour coder respectivement le symbole 0 et le symbole 1.
2. Effectuez un "filtrage manuel" du spectre du signal en utilisant la fonction *coupe_bande* qui est fournie (*help coupe_bande*)
3. Cette procédure de filtrage peut-elle être envisagée en temps réel ?

3 Décodage

1. Exécutez le bloc d'instructions *Tracé du signal temporel*, et comparez le signal original avec le signal filtré. Notez le code binaire que vous devriez désormais être en mesure de lire.
2. Convertissez ce code en caractères ASCII en utilisant la table fournie.
3. Testez la validité de votre code avec la fonction *verifier_code*.
N.B. Pour recommencer avec un nouveau code, relancez la procédure generer_code.

Exemple de signal codant la lettre 'g' (01100111)



ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(0010 1000
N	0100 1110	l	0110 1100)	0010 1001
				space	0010 0000