**TP 3 : CODAGE SOURCE ET CANAL**

**Audio, Image et Vidéo**

**Master 1 RT et ST**

1. **INTRODUCTION**

Le codage canal et le codage source sont deux types de codage indispensables dans pratiquement la majorité des systèmes de transmission numériques notamment sans fil. Ils sont également utilisés lors du stockage des données dans un support à taille limitée tels que les DVD, disques durs et autres clés USB. Si le codage canal permet souvent de détecter et de corriger les erreurs dues au canal de transmission, le codage source spécialement avec perte permet de réduire drastiquement le débit des données originales pour les adapter aux capacités du canal de transmission ou à la taille du support de stockage. Comme il existe différents types de données telles que la vidéo, l’image et l’audio, alors différentes techniques et normes aussi bien pour le codage canal que pour le codage source, ont été créées et adoptées. Ainsi, pour le codage source on trouve, à titre d’exemples, le JPEG pour la compression d’images, le MP3 pour la compression du son et les MPEGx pour la compression vidéo. Alors que pour le codage canal avec les différents FEC on trouve également plusieurs techniques pour chaque type de données et surtout pour chaque type de canal de transmission.

Dans ce TP, nous allons utiliser un outil fort intéressant, connu sous le nom ‘**’IHU Virtual Lab C’’** créé sous Matlab par ‘**’IHU SCHOOL OF SCIENCE & TECHNOLOGY’’**

[www.tech.ihu.edu.gr/index.php/en/research/item/30-virtual-labs.html](http://www.tech.ihu.edu.gr/index.php/en/research/item/30-virtual-labs.html)

Cet outil, exécuté sous Matlab, permet d’effectuer différentes opérations sur les images, les sons et la vidéo. Nous allons nous intéresser uniquement à :

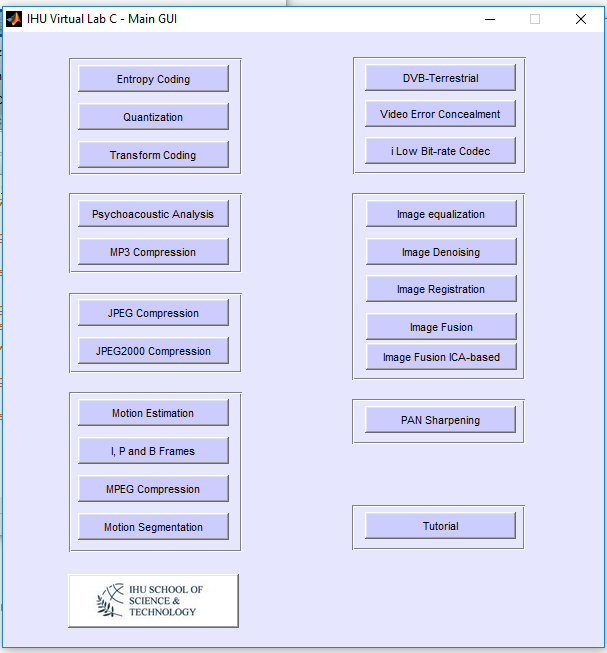
* Codage Entropique
* Codage par Transformation
* Compression JPEG
* Codage canal iLow Bit-Rate

Même si les trois premières ont déjà été vues auparavant avec d’autres outils, il est intéressant de comparer les résultats obtenus avec ce nouvel outil.

1. **EXECUTION DE VLabC**

Commencez par copier et décompresser cet outil, dans un dossier que vous devez d’abord créer. Ce fichier compressé contient bien évidemment le programme source de cet outil sous Matlab mais également quelques images tests et deux fichiers sons tests. On vous propose en plus un autre fichier compressé contenant d’autres sons tests que nous vous invitons à décompresser dans le même dossier.

Pour exécuter ce programme il suffit de lancer sous Matlab le fichier ‘’MainVlabGUI’’, alors une fenêtre va s’ouvrir pour vous offrir toutes les possibilités de cet outil (figure 1).



**Figure 1 : GUI (Graphic User Interface) du VLabC**

1. **PREMIER TRAVAIL A FAIRE**

Le premier travail demandé concerne le codage entropique, le codage par transformation et la compression JPEG

**3.1 Codage entropique (Entropy Coding)**

Cliquez sur le bouton ‘’Entropy Coding’’ de la fenêtre GUI obtenues après l’exécution sous Matlab du programme VLabC. Une nouvelle fenêtre va apparaitre. Dans le menu ‘’File’’ nous pouvons alors ouvrir différents types de fichiers (Binaire, Image ou Son). Vous devez faire faire plusieurs tests sur des sons différents et images différentes. Pour chaque test, image ou son, vous aurez plusieurs résultats affichés sur cette fenêtre. Essayez d’expliquer de quoi il s’agit et surtout de discuter ces résultats.

**3.2 Codage par transformation (Transform Coding)**

Cliquez sur le bouton ‘’Transform Coding’’ de la fenêtre GUI obtenues après l’exécution sous Matlab du programme VLabC. Une nouvelle fenêtre va apparaitre. Cette fois ci elle ne concerne que les images. Dans le menu ‘’File’’ on ouvre un exemple d’image, alors instantanément plusieurs images vont apparaitre en plus de la valeur de l’entropie. A droite de la fenêtre et aussi en bas de la première image affichée (image originale) nous avons quatre curseurs qu’on peut ajuster en ligne. Essayez de comprendre le rôle de ces curseurs. Nous pouvons aussi visualiser l’histogramme de l’image quantifiée. C’est quoi un histogramme d’une image ? Répéter ces opérations avec plusieurs images tests.

**3.3 Compression JPEG (JPEG Compression)**

Cliquez sur le bouton ‘’JPEG Compression’’ de la fenêtre GUI obtenues après l’exécution sous Matlab du programme VLabC. Une nouvelle fenêtre va apparaitre. Cette fois ci aussi, elle ne concerne que les images. Dans le menu ‘’File’’ on ouvre un exemple d’image, alors instantanément plusieurs images vont apparaitre en plus de la valeur du facteur de qualité (Quality Factor), Bitrate, du taux de compression (Compression ratio) et du PSNR. Un curseur ‘’Quality Factor’’ va nous permettre de choisir la qualité de l’image reconstruite en fonction du Bitrate ….etc.

Essayez de comprendre et d’expliquer chaque information (image ou donnée) affichée. Pour chaque image testée prenez au moins cinq valeurs du ‘’Quality Factor’’ par exemple 1, 25, 50, 75 et 100 et notez le PSNR et Bitrate correspondants. Ensuite tracez par Matlab la courbe PSNR en fonction du Bitrate. Qu’en déduisez vous.

Refaire cette opération avec plusieurs images différentes.

1. **DEUXIEME TRAVAIL A FAIRE**

Dans ce cas nous allons nous intéresser au codage canal et plus particulièrement à codage particulier utilisé pour les codecs vocaux adaptés aux applications de voix sur IP.

**4.1 Codec ILBC**

Le codec Internet à faible débit binaire (iLBC) est un codec vocal à bande étroite, adapté aux applications de voix sur IP (VoIP). Il a été développé par Global IP Solutions. C'est l'un des codecs utilisés par Google Talk, Yahoo! Messenger et beaucoup d'autres. En bref, l'algorithme de codage iLBC utilise un codage prédictif linéaire indépendant des blocs. Il fonctionne sur des trames de parole d'une longueur de 20 ou 30 millisecondes. Le choix du codage indépendant des trames de parole peut être justifié pour les raisons suivantes (entre autres): Exploitation de codecs ordinaires à faible débit les dépendances entre les cadres de parole. Cela entraîne toutefois une propagation d'erreur lorsque des paquets sont perdus ou retardés. En revanche, les trames de parole codées par iLBC sont indépendantes, ce qui évite le problème ci-dessus. iLBC est défini dans la RFC 3951. Ses caractéristiques les plus importantes peuvent être résumées comme suit:  
• Fréquence d'échantillonnage et quantification: 8 kHz et 16 bits / échantillon. Tous les signaux d'entrée vers le codeur iLBC doivent être des signaux de parole à modulation d’impulsion codée ou MIC (**PCM : Pulse Code Modulation**) échantillonnés à exactement 8000 Hz avec 16 bits par échantillon. Par conséquent, le débit binaire d'entrée est de [8 kHz] \* [16 bits] = 128 kbps.

• Longueur de trame vocale: 20 ms (160 échantillons =20×10-3×8×103) ou 30 ms (240 échantillons).

• Deux modes de codage disponibles avec des débits binaires:  
15,20 kbps (trames de 20 ms) ou 13,33 kbps (trames de 30 ms). L’avantage du codage toutes les 30 ms est que le débit de données codé est inférieur: 13,33 kbps par opposition à  
15,20 kbps pour des trames de 20 ms. Cependant, le codage toutes les 30 ms entraîne un délai supplémentaire de 50% dans la parole reçue. Etant donné que le débit binaire d'entrée est de 128 kbps, iLBC compresse la voix à 10,4% ou 11,9% du débit de données d'origine pour les modes de 13,33 kbps et de 15,20 kbps, respectivement.

• L'algorithme de codage est basé sur l'analyse du signal de parole par le codage prédictif linéaire (**LPC : Linear Predictive Coding**) et le codage du signal résiduel LPC. Le codage est indépendant d'une image à l'autre.

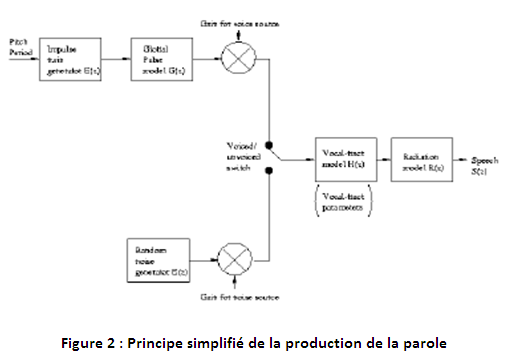
• En raison de l’utilisation du codage par images indépendantes:

o Il n'introduit pas la propagation d'erreur d'une image à l'autre, comme expliqué.

o La mise en œuvre et l'utilisation des techniques de masquage de perte de paquets (**PLC : Packet Loss Concealment**) sont faciles. En bref, le travail de la technique ‘’**PLC’’** consiste à interpoler la parole pour les paquets manquants, en se basant sur les paquets avant et immédiatement après le paquet manquant.

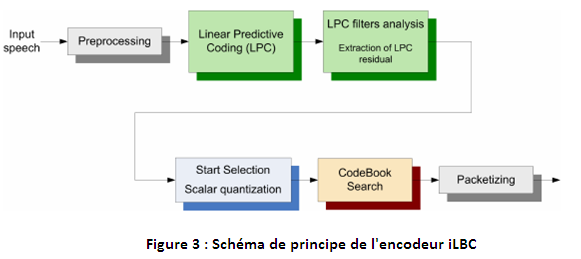
* 1. **Aperçu sur le Codage prédictif linéaire (LPC)**

La parole est produite par l'interaction du tractus vocal avec les cordes vocales. La procédure de production de la parole peut être décrite grossièrement comme suit: La glotte (accords vocaux) produit un son «buzz». Ce le son est caractérisé par son intensité (loudness) et sa fréquence (pitch). L’appareil vocal, c’est-à-dire la gorge et la bouche, peut être modélisé comme un tube. Ce tube est caractérisé par ses fréquences de résonance. La propagation du son entrant dans le conduit vocal donne naissance à des formants ou à des bandes de fréquences améliorées dans le son produit. La procédure décrite produit les sons voisés. Cependant, des sibilants ("sifflements") et des sons plosifs ("popping") sont également produits. Celles-ci sont générées par l'action de la langue, des dents, des lèvres et de la gorge. Bien que brut, ce modèle est en réalité une approximation proche de la procédure de production de parole réelle. Par conséquent, le codage prédictif linéaire (LPC) est basé sur ce modèle. Le LPC analyse un signal de parole de la manière suivante: Initialement, les formants sont estimés. Utilisant une approche basée sur un filtrage inverse, l’effet des formants est supprimé du signal de parole. L'intensité et la fréquence du signal "buzz" restant sont ensuite estimées. Le signal restant après la soustraction du signal modélisé filtré est appelé résidu LPC. L'intensité et la fréquence du «buzz», les paramètres décrivant les formants et le signal de résidu peuvent être stockés et / ou transmis. La synthèse du signal de parole basée sur le LPC est basée sur le processus inverse: les paramètres «buzz» et le résidu sont utilisés pour créer un signal source. Les paramètres de formants sont utilisés pour créer un filtre. Le signal source passe à travers le filtre pour produire de la parole.



* 1. **Encodeur iLBC**

Le schéma fonctionnel de la figure 3 donne un aperçu de tous les composants de la procédure de codage iLBC. Ici, une brève description du codeur iLBC est donnée. Pour des raisons de simplicité, la description est limitée au mode 15,20 kbps.



• Prétraitement: dans certaines applications, le signal de parole enregistré n'a pas un niveau de tension continue nul (n’est pas centré) et / ou contient du bruit 50/60 Hz. Ces composants doivent être supprimés à l'aide d'un filtre passe-haut.

• Calculer les paramètres LPC, quantifier et interpoler: le signal d’entrée pré-traité est partitionné en blocs de 160 échantillons (taille d'image 20ms). Chaque bloc est divisé en 4 sous-blocs consécutifs de 40 échantillons chacun. Maintenant, le codeur applique une fenêtre lisse, centrée sur le troisième sous-bloc, puis effectue une analyse LPC: un ensemble de fréquences spectrales linéaires (LSF : line-spectral frequencies) est obtenu, quantifié et interpolé pour obtenir des coefficients LSF pour chaque sous-bloc (Les LSF ont plusieurs propriétés, par exemple une sensibilité plus faible au bruit de quantification, ce qui les rend supérieures à une quantification directe des LPC).

• Utiliser des filtres d'analyse sur la parole pour calculer le résidu: le bloc d'échantillons de parole est filtré par les filtres d'analyse LPC quantifiés et interpolés pour produire le signal résiduel. En particulier, le filtre d'analyse LPC correspondant pour chaque sous-bloc de 40 échantillons est utilisé pour filtrer les échantillons de parole pour le même sous-bloc. Le signal en sortie de chaque filtre d'analyse LP constitue le signal résiduel du sous-bloc correspondant.

• Sélectionnez la position de l'état de démarrage de 57 échantillons: le codeur identifie les deux sous-blocs consécutifs du résidu ayant l'énergie pondérée maximale. Dans ces deux sous-blocs, l'état de début (segment) est sélectionné parmi deux choix: les 57 premiers échantillons ou les 57 derniers échantillons des deux sous-blocs consécutifs. Le segment sélectionné est celui de plus haute énergie. Ensuite, l'état de départ est quantifié.

• Rechercher dans le ‘’codebook’’ pour chaque sous-trame: une procédure de codage dynamique est utilisée pour coder:

1. les 23 échantillons restants dans les deux sous-blocs contenant l'état de début,
2. les sous-blocs après l'état de début dans le temps,
3. 3) les sous-blocs avant l'état de départ dans le temps. À savoir, la cible de codage peut être soit les 23 échantillons restants des deux sous-blocs contenant l'état de départ, soit un sous-bloc de 40 échantillons. Les trois gains de vecteur de code résultants sont codés avec une quantification scalaire à 5, 4 et 3 bits, respectivement.

Le codage par codebook est basé sur codebook adaptatif construit à partir d'une mémoire de codebbok contenant des échantillons d'excitation LPC décodés de la partie déjà codée du bloc.

* 1. **Décodeur iLBC**

Chaque bloc de signal codé reçu est décodé au décodeur. Si un bloc de signal est perdu ou retardé pendant la transmission, le décodeur exécute une opération de masquage de perte de paquet (**Packet Loss Concealment : PLC**). Le décodage de chaque bloc commence par décoder et interpoler les coefficients LPC. Ensuite, l'état de départ est décodé. Pour les segments codés par le livre de codes, chaque segment est décodé en construisant les trois vecteurs de code donnés par les index du livre de codes reçus, de la même manière que les vecteurs de code ont été construits dans le codeur. Les trois facteurs de gain sont également décodés et le signal décodé résultant est donné par la somme des trois vecteurs de livre de codes, mis à l'échelle avec le gain respectif. Une opération de dissimulation de perte de paquets (PLC) est facilement intégrée au décodeur. Le fonctionnement de l'automate peut être basé sur la répétition de filtres LPC et l'obtention du signal résiduel LPC à l'aide d'une estimation prédictive à long terme à partir de blocs résiduels précédents.

* 1. **Guide de l’utilisateur ‘’I-LBC’’**

L’interface utilisateur graphique «Internet Low Bit Rate (iLBC)» est illustrée à la Fig. 1. Avec l’utilisation d’un modèle Simulink approprié, cette interface graphique simule le fonctionnement du codec iLBC, un codec conçu pour le codage et la décodage de la parole pour transmission via VoIP (Voice Over Internet Protocol). Le modèle iLBC Simulink a été implémenté dans le bloc de traitement de signaux de Mathwork. Le modèle déjà existant a été légèrement modifié et étendu, en fonction des besoins du laboratoire virtuel. La partie principale du modèle étendu est illustrée à la Fig. 2. L’utilisation de l’interface graphique «iLBC» permet à l’utilisateur de modifier facilement les paramètres du modèle lors de la simulation et d’étudier les résultats du modèle. En utilisant cette démo, l'utilisateur peut:

• Découvrez quelques idées utilisées dans le codage de la parole pour la transmission VoIP.  
• Étudiez les techniques de dissimulation de perte de paquets utilisées par iLBC.  
• Observez la qualité améliorée de la parole décodée lorsque la dissimulation est utilisée.

**NOTATION**La notation utilisée sur la figure 1 est la suivante:

• Ti: bouton de la barre d’outils

• S: curseur

• C: ComboBox (menu contextuel)

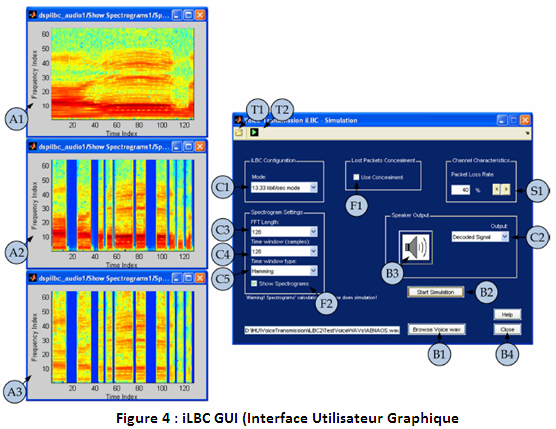
• B: bouton

• F: indicateur (CheckBox)

• A: Axes

* 1. **Utilisation de l’interface utilisateur graphique (GUI)**

L'utilisateur peut sélectionner le fichier vocal d'entrée à l'aide du bouton T1 ou du bouton B1 de la barre d'outils. La simulation du modèle commence en appuyant sur l’un des boutons B2 ou T2. Pendant la simulation, l'utilisateur peut écouter: a) le signal d'origine, b) le signal décodé ou c) le signal résiduel LPC (codage à prédiction linéaire). Le signal envoyé à l'enceinte est spécifié par la sélection dans le ComboBox C2. L'utilisateur peut désactiver / activer les haut-parleurs en appuyant sur le bouton B3.



**4.7 Travail à faire**

**Première expérimentation**

**Partie 1**

• Ouvrez le fichier wav «speech\_dft\_8kHz.wav» (bouton B1).

• Définissez le mode iLBC (ComboBox C1) sur 13,33 kbps.

• Définissez la dissimulation des pertes de paquets (CheckBox F1) sur OFF.

• Faites varier le taux de perte de paquets (curseur S1) de 85% à 5%, avec une diminution de 10%. Pour chaque valeur, écrivez:

o 1, pour un discours clairement compréhensible

o 0, pour un discours marginalement compréhensible

o -1, pour un discours incompréhensible.  
• Tracer le score contre le taux de perte de paquets.

• Répéter l'opération avec la dissimulation de perte de paquets (CheckBox F1) sur ON.  
(Remarque: avant d'activer la simulation, arrêtez la simulation).

**Partie 2**

• Répétez l’expérience de la partie 1 avec le mode iLBC (ComboBox C1) réglé sur 15,20 kbps et avec  
masquage réglé sur ON et OFF.

**Partie 3**

• Comparez les quatre graphiques obtenus à partir des parties 1 et 2 et commentez le résultat. La procédure ci-dessus est appelée «évaluation subjective» et est souvent utilisée dans l’évaluation expérimentale multimédia.

**Deuxième expérimentation**

• Ouvrez le fichier wav “AENAOS.wav” (bouton B1). La parole dans ce fichier contient beaucoup de voyelles (son voisé).

• Répétez la partie 1 de l'exercice 1 en masquant la perte de paquets (CheckBox F1) à ON et à OFF.  
• Activez les spectrogrammes (CheckBox F2). Couper le son du système (bouton B3).  
• Répétez et observez le spectrogramme du signal décodé. Essayez de comprendre comment fonctionne la dissimulation.

**Troisième expérimentation :** (Comparaison des sons sinusoïdaux et à ‘’**shaped-noise’’**)  
• Répétez la partie 1 de l'exercice 1 en utilisant le fichier wav «tonal.wav», en faisant varier le taux de perte de paquets (curseur S1) de 65% à 5%, avec une décrémentation de 10% (avec et sans dissimulation). Pour chaque valeur, écrivez:

o 1, pour un son clair

o 0, pour un son marginalement acceptable

o -1, pour le son corrompu.  
• Recommencez l’utilisation du fichier wav “clapping.wav”.  
• Commentez les résultats.