Livrable 4

Livrable Niveau B :

1. Phase d’émission :

* Transformation du message : Convertir le message texte en trames.
* Modulation : Moduler les trames en signal sonore
* Emission : Jouer le signal sonore généré

1. Phase de réception :

* Réception du Signal : Capturer le signal sonore
* Démodulation : Démoduler le signal pour obtenir les trames
* Reconstruction du Message : Reconstituer le message d’origine.

1. Système de détection d’Erreur :

* Implémenter un système de détection d’erreur, comme une somme de contrôle

1. Jupyter NoteBook

Livrable Niveau A :

1. Envoi et reconstitution d’un message type son :

* Conversion sonore : Transformer le message en signal sonore et inversement

1. Gestion de la réception avec bruit :

* Simulation de bruit : Ajouter du bruit au signal reçu
* Correction d’erreur (optionnel) : Utiliser un CRC ou un code de correction d’erreur pour corriger les erreurs dues au bruit

1. Liaison Half Duplex (Accusé de réception) :

* Mise en place : Configurer la communication pour permettre la réception d’un accusé de réception

1. Simulation entre 2 PC :

* Génération sur un PC 1 : Diviser le processus d’émission et réception entre deux machines

1. Second type de Modulation / Démodulation :

* Implémentation : Ajouter un second type de modulation / Démodulation en option (Faire une fonction qui choisi la modulation )

Fonction utiliser :

La modulation peut être définie comme le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission

Porteuse : onde sinusoidale de support Phase amplitude fréquence

l'amplitude fait référence à la déviation maximale d'une onde sonore à partir de la position centrale

***Bonjour*** Bienvenue au département Recherche et Développement de l'agence AIL3C, j’ai le plaisir avec mon équipe ici présente composée de Louis William et moi-même de vous proposer une solution permettant la communication vers l’extérieur dans la situation présente

***Moduler\_fsk() :***

* trame : C'est une séquence binaire (une trame) représentant un caractère du message. Par exemple, '01001000' pour le caractère 'H'.
* La boucle **for bit in trame** itère sur chaque bit de la trame.
* Si bit est 0, la fonction utilise la fréquence freq\_bit0 pour moduler le signal. Cela signifie qu'une onde sinusoïdale de freq\_bit0 est générée pendant la durée spécifiée par duree\_bit.
* Si bit est 1, la fonction utilise la fréquence freq\_bit1 pour moduler le signal. Cela signifie qu'une onde sinusoïdale de freq\_bit1 est générée pendant la même durée.
* (2 \* np.pi \* freq\_bit0 \* np.arange(0, duree\_bit, 1/echantillonage))
  + Génère une onde sinusoïdale de fréquence freq\_bit0 pendant la durée spécifiée par duree\_bit.
* (2 \* np.pi \* freq\_bit1 \* np.arange(0, duree\_bit, 1/echantillonage))
  + Génère une onde sinusoïdale de fréquence freq\_bit1 pendant la durée spécifiée par duree\_bit.
* **extend** est utilisé pour ajouter les échantillons générés au signal\_module.

Résumé de la fonction :

La fonction moduler\_fsk est utilisée pour créer le signal modulé en fréquence en concaténant les résultats de l’utilisation de cette modulation à chaque bit de la trame. Elle renvoie un tableau numpy représentant le signal modulé en fréquence pour une trame donnée. En ajustant les valeurs de freq\_bit0 et freq\_bit1.

***demoduler\_fsk(signal\_recu) :***

**signal\_recu**: Le signal sonore reçu.

* Variables locales:
* bits\_recus: Une liste pour stocker les bits démodulés.
* seuil\_detection: Le seuil utilisé pour prendre la décision entre les bits 0 et 1.

Boucle principale:

La boucle parcourt le signal reçu par morceaux de la taille d'un bit.

Calcul de la corrélation:

Le calcul de la corrélation est utilisé dans ce contexte pour mesurer la similarité entre le signal reçu et les signaux modulés correspondant aux bits 0 et 1. La corrélation est une mesure statistique qui indique à quel point deux signaux sont semblables. Dans le cas de la démodulation FSK (Frequency Shift Keying), cela permet de déterminer quelle fréquence (et donc quel bit) a été transmise.

La fonction np.correlate est utilisée pour calculer la corrélation entre l'échantillon actuel et les signaux modulés correspondant aux bits 0 (corrélation\_bit0) et 1 (corrélation\_bit1).

Calcul de la corrélation avec corrélation\_bit0 et corrélation\_bit1:

corrélation\_bit0: La corrélation entre l'échantillon reçu et le signal modulé correspondant au bit 0 (fréquence freq\_bit0).

corrélation\_bit1: La corrélation entre l'échantillon reçu et le signal modulé correspondant au bit 1 (fréquence freq\_bit1).

Décision basée sur la corrélation maximale:

La décision sur le bit reçu (bit\_recu) est prise en comparant les corrélations maximales. Si la corrélation maximale pour le bit 0 est supérieure à un seuil (multiplié par la corrélation maximale pour le bit 1), alors le bit reçu est décidé comme étant '0', sinon c'est '1'.

En d'autres termes, la corrélation est utilisée pour évaluer à quel point le signal reçu correspond aux signaux modulés attendus pour les bits 0 et 1. La décision finale est prise en comparant les valeurs de corrélation maximales.

Le seuil est introduit pour introduire une marge de tolérance. Si la corrélation maximale pour le bit 0 est suffisamment plus élevée que celle pour le bit 1 (en fonction du seuil), le bit 0 est décidé comme étant le bit reçu, et vice versa.

En résumé, le calcul de la corrélation est un outil qui permet d'évaluer la similarité entre le signal reçu et les signaux modulés, facilitant ainsi la détection du bit transmis

Retour:

La fonction retourne la liste des bits démodulés.

Cette fonction démodule le signal FSK en comparant la corrélation entre l'échantillon reçu et les signaux modulés correspondant aux bits 0 et 1. Elle utilise un seuil de détection pour prendre la décision finale sur le bit reçu. Les bits démodulés sont stockés dans la liste bits\_recus.

***Reconstruire\_message(bits\_recus) :***

La fonction reconstruire\_message prend une liste de bits démodulés en entrée, organise ces bits en trames de 8 bits, puis convertit chaque trame en un caractère ASCII. En résumé :

Entrée :

bits\_recus: Une liste de bits démodulés.

Fonctionnement :

La fonction divise la liste de bits en trames de 8 bits.

Chaque trame est convertie en un caractère ASCII à l'aide de la fonction chr.

Les caractères ASCII correspondant à chaque trame sont concaténés pour former le message complet.

Sortie :

La fonction renvoie le message reçu, qui est une séquence de caractères ASCII reconstruite à partir des bits démodulés.

En d'autres termes, la fonction reconstruire\_message traduit la séquence binaire de bits démodulés en une chaîne de caractères ASCII, ce qui permet de récupérer le message d'origine à partir du signal démodulé.

***Système de contrôle :***

Le système de contrôle utilise une somme de contrôle (bit de parité) pour détecter d'éventuelles erreurs dans la transmission des trames.

Voici comment cela fonctionne en détail :

Ajout de la somme de contrôle lors de l'émission :

Avant de moduler les trames pour l'émission, je calcul une somme de contrôle pour chaque trame. La fonction calculer\_somme\_controle somme tous les bits de la trame (représentant un caractère ASCII) et prend le reste de la division par 2. Ce résultat est ajouté à la fin de chaque trame.

Transmission du signal modulé avec la somme de contrôle :

Le signal modulé, comprenant les trames originales avec leurs bits de parité, est ensuite émis.

Démodulation lors de la réception :

Lors de la réception, le signal est démodulé pour obtenir les bits qui ont été transmis.

Décodage des trames reçues :

Les bits reçus sont regroupés en trames, et la somme de contrôle ajoutée précédemment est retirée de chaque trame. Ainsi, chaque trame a une longueur de 8 bits correspondant à un caractère ASCII.

Vérification de la somme de contrôle pour chaque trame :

La fonction calculer\_somme\_controle est à nouveau utilisée pour calculer la somme de contrôle des bits de chaque trame reçue. La somme de contrôle calculée est comparée à la somme de contrôle ajoutée lors de l'émission.

Détection d'erreurs :

Si la somme de contrôle calculée ne correspond pas à la somme de contrôle ajoutée, une erreur est détectée pour cette trame. Cela signifie que des erreurs ou des altérations ont probablement eu lieu pendant la transmission.

Affichage des résultats :

Les bits reçus, les messages reçus et un indicateur d'erreur (True si erreur, False sinon) sont imprimés à des fins de diagnostic.

En résumé, ce mécanisme de contrôle utilise une somme de contrôle pour détecter les erreurs potentielles. Si une trame reçue a une somme de contrôle incorrecte, cela indique qu'il y a probablement eu des erreurs lors de la transmission de cette trame. Ce système n'est pas en mesure de corriger les erreurs, mais seulement de les détecter.

--------------------------------------------------------MODULATION FSK------------------------------------------------------

Documentation du Code "Modulation FSK avec Contrôle"

Introduction

Le code présent réalise une modulation de fréquence par déplacement de fréquence (FSK) avec ajout de contrôle de parité pour la transmission d'un message. La phase d'émission implique la conversion d'un message texte en trames binaires, ajout d'un bit de parité (somme de contrôle), puis modulation FSK. La phase de réception décode le signal modulé, détecte les erreurs grâce à la somme de contrôle, puis reconstruit le message original.

Phase d'Émission

Entrée du Message :

L'utilisateur est invité à saisir un message à moduler.

Conversion en Trames Binaires :

Le message est converti en trames binaires de 8 bits chacune.

Calcul de la Somme de Contrôle :

Une fonction calculer\_somme\_controle est définie pour calculer le bit de parité (somme de contrôle) de chaque trame.

Ajout de la Somme de Contrôle :

La somme de contrôle est ajoutée à chaque trame binaire.

Modulation FSK avec Contrôle :

Le signal est modulé en FSK pour chaque bit de la trame avec les fréquences correspondant à '0' et '1'.

La modulation est réalisée à l'aide de sinusoïdes de fréquences spécifiques.

Enregistrement du Signal Modulé :

Le signal modulé est enregistré dans un fichier audio.

Phase de Réception

Récupération du Signal :

Le signal sonore modulé est récupéré à partir du fichier audio.

Démodulation FSK avec Contrôle :

Une fonction demoduler\_fsk\_avec\_controle est définie pour démoduler le signal FSK.

La corrélation entre l'échantillon reçu et les fréquences de modulation est calculée pour chaque bit.

Vérification de la Somme de Contrôle :

La somme de contrôle est vérifiée pour chaque trame reçue.

Reconstruction du Message :

Une fonction reconstruire\_message\_avec\_controle reconstruit le message original à partir des bits reçus en vérifiant la somme de contrôle.

Affichage des Résultats :

Les bits reçus, les erreurs détectées et le message reconstruit sont affichés.

Affichage du Signal Reçu :

Un graphique du signal reçu est affiché à l'aide de la bibliothèque Matplotlib.

Notes Additionnelles

Le seuil de détection est défini pour décider du bit reçu.

La gestion des erreurs détectées est effectuée pendant la reconstruction du message.

Utilisation

Exécutez la fonction modulation1.

Saisissez le message à moduler (entre 5 et 10 caractères).

Le programme effectue la modulation, enregistre le signal, puis effectue la démodulation, vérifie la somme de contrôle, et reconstruit le message.

Les résultats et le graphique du signal sont affichés.

--------------------------------------------------------MODULATION FSK------------------------------------------------------