

## Simulation sous Matlab/Simulink d'une chaîne de transmission numérique en bande de base. Application au codage canal

E. Rius<sup>\* x</sup> et C. Quendo<sup>\*\*</sup> (*Université de Bretagne Occidentale, Brest*)  
Mis en ligne le 3 octobre 2003.

### Résumé

Cet article présente une expérience d'enseignement des techniques de codage de canal [1-8] à l'Université de Bretagne Occidentale à Brest. Cet enseignement, développé sous le logiciel Simulink, s'adresse à des étudiants de second cycle en électronique et télécommunications. Il repose sur la mise en oeuvre de codeurs/décodeurs de canal classiques au sein d'une chaîne de transmission numérique. La transmission se fait en bande de base, sur un canal idéal perturbé par du bruit. Une telle manipulation permet aux étudiants de visualiser aisément l'efficacité des techniques de codage sur la sécurisation de l'information transmise, mais aussi en contrepartie la dégradation du débit.

**Mots-clés** : travaux pratiques, Matlab/Simulink, transmission numérique, transmission en bande de base, codage de canal, codage de parité, codage de Hamming.

© EDP Sciences, 2003.

---

**Niveau de connaissances requis.** Théorie de l'information, codage de source, codage de canal, notions de systèmes de télécommunication, notions de Matlab et Simulink.

**Niveau des étudiants.** Licence ou Maîtrise (filière EEA ou assimilée).

---

\* **Eric Rius**<sup>1</sup> est actuellement maître de conférences à l'Université de Brest en 63<sup>e</sup> section. Il enseigne au département d'Électronique et à l'IUP Télécoms et Réseaux de l'université. Ses enseignements concernent les systèmes de télécommunication (codage de canal, électronique basse fréquence et dispositifs hyperfréquences). Les systèmes de télécommunication et plus spécifiquement le filtrage hyperfréquence constituent les thèmes privilégiés de ses activités de recherche, effectuées au LEST UMR CNRS 6165.

✉ e-mail : Eric.Rius@univ-brest.fr (auteur de correspondance)

\*\* **Cédric Quendo**<sup>1</sup> est actuellement attaché temporaire d'enseignement et de recherche à l'Université de Brest. Ses enseignements concernent les systèmes de télécommunication (codage de canal, électronique basse fréquence et dispositifs hyperfréquences). Les systèmes de télécommunication et plus spécifiquement le filtrage hyperfréquence constituent les thèmes privilégiés de ses activités de recherche, effectuées au LEST UMR CNRS 6165.

---

<sup>1</sup> Université de Bretagne Occidentale, Département Électronique, UFR Sciences, 6 av. le Gorgeu, F-29285 Brest Cedex, France.

IUP Télécom et Réseaux, av. le Gorgeu, F-29285 Brest Cedex, France.

## 1. Introduction

Dans cet article, nous décrivons une série de manipulations écrites sous le logiciel Matlab/Simulink et permettant de se familiariser avec les techniques de codage de canal. L'objectif ici est d'illustrer le plus simplement possible le module de théorie de l'information dont la partie concernant le codage de canal constitue la finalité. La démarche consiste à comparer deux chaînes de transmission numérique en bande de base. La première ne contient pas d'accessoires de codage et de décodage de canal tandis que la seconde en est pourvue. Une comparaison des performances de ces deux chaînes de transmission en présence de bruit, permet de mettre en évidence aisément un certain nombre de concepts élémentaires. D'une part, on visualise relativement bien les effets du bruit en terme de brouillage du signal utile et d'autre part on peut observer l'effet des codeurs et décodeurs sur la détection et la correction des erreurs. La sécurisation de l'information n'est pas le seul point à être observé, car on visualise aussi la dégradation du débit. Celle-ci est associée au nombre de bits de redondance des mots-codes qui vont dépendre de la puissance de correction du code choisi.

Cette manipulation se voulant illustrative, elle doit posséder un certain nombre de paramètres réglables par l'opérateur, tels que l'amplitude du signal utile et la puissance du bruit. De plus, le caractère modulaire de Matlab/Simulink permet de découper la chaîne de transmission en blocs différents et ainsi de bien dissocier les problèmes. C'est-à-dire que la chaîne de transmission doit faire apparaître clairement les modules de base que sont entre autres, l'émetteur pour la mise en forme physique des signaux, le détecteur optimal qui effectue un pré-traitement sur le bruit, le canal lui-même, etc.

## 2. Méthodologie et objectifs

Le travail demandé aux étudiants est la réalisation complète sous Matlab/Simulink des différentes chaînes de transmission. Un volume horaire de **20** heures sous la forme de projet encadré et incluant une séance de présentation générale du module est prévu. L'objectif de cette dernière est triple.

Dans un premier temps, il s'agit de présenter la finalité du travail, c'est-à-dire de faire quelques démonstrations à partir de chaînes de transmission déjà finalisées. Ceci permet de montrer d'une part la faisabilité et d'autre part l'efficacité des techniques de codage en présence de bruit.

Dans un deuxième temps, une initiation à simulink est proposée. À l'issue de celle-ci, l'étudiant doit être capable de construire et de simuler correctement un modèle et doit savoir créer et utiliser des bibliothèques de blocs. Il doit aussi maîtriser les fonctions-S et la programmation associée. Finalement, il doit être sensibilisé au paramétrage des fonctions-S et des blocs de façon à réaliser des modèles suffisamment modulaires pour être facilement interfaçables.

Enfin, à l'aide d'une série de fichiers de démonstrations, on suggère à l'étudiant une démarche de travail pour atteindre l'objectif final. Cette démarche consiste à construire la chaîne pas à pas, la grande difficulté étant bien entendu d'assurer le synchronisme entre les différents blocs. Les points durs sont mis en évidence ici et des techniques ou des astuces de débogage sont proposées.

Nous insistons ici sur le fait que l'objectif pour l'étudiant est de construire un ensemble complet de chaîne de transmission incluant les outils de codage de canal et de vérifier leur efficacité dans des cas où le signal reçu est bruité. À cet effet, une partie du volume horaire doit être impérativement affectée à l'étude de la chaîne de transmission dans sa version finale. En construisant lui-même les codeurs et décodeurs, l'étudiant utilise les notions de matrice génératrice et de matrice de contrôle de parité et fait ainsi le lien avec les enseignements magistraux concernant la théorie de l'information. C'est un passage obligé, avant d'utiliser les blocs déjà prédéfinis dans la boîte à outil « communications toolbox simulink block library ». L'usage de cette dernière permet d'aller plus loin dans l'étude des systèmes de communications numériques, en autorisant par exemple pour un système donné la quantification du taux d'erreur binaire en fonction du rapport signal à bruit. Cependant, cela suppose déjà acquis pour l'étudiant un certain nombre de bases.

Comme nous le verrons par la suite, nous avons choisi d'implanter les codeurs et décodeurs sous forme de portes logiques. Il est évident que l'implantation sous forme de FPGA est aussi tout à fait pertinente et peut être examinée dans un second temps.

### 3. Description de la chaîne de transmission (sans codage de canal)

#### 3.1. Présentation générale

La chaîne de transmission permet le transport, en bande de base, d'une information sous forme de données numériques codées en binaire. Elle est composée d'un émetteur, d'un canal et d'un récepteur. Les perturbations sont modélisées par un bruit blanc (de puissance réglable), gaussien et centré qui est ajouté au signal utile en sortie du canal. Le synoptique général de la manipulation est présenté à la figure 1.

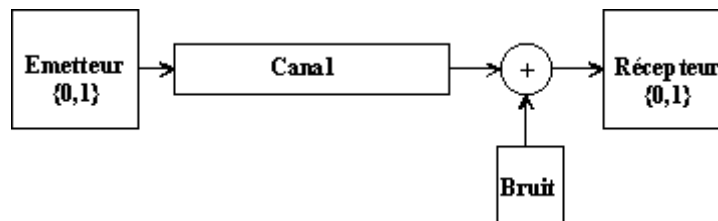


Fig. 1. Schéma synoptique général de la chaîne de transmission sans codage de canal.

#### 3.2. L'émetteur

Les données numériques qui représentent l'information sont codées sur  $lgmot$  bits et sont présentées à l'entrée de la chaîne de transmission les unes après les autres. Chacune d'elles doit rester présente pendant  $lgmot$  unités de temps  $Tech$ , une interface série permettant ensuite d'émettre ces  $lgmot$  bits de façon série à travers le canal. L'unité de temps  $Tech$  choisie est bien sûr arbitraire et elle cadencera de façon synchrone tous les blocs de la chaîne de transmission.  $1/Tech$  est donc le rythme digital. Enfin, un émetteur physique assure la mise en forme des signaux en associant aux symboles binaires des tensions bipolaires d'amplitude réglable. Il s'agit donc d'un signal NRZ.  $lgmot$  est bien sûr un entier.

#### 3.3. Le récepteur

Le récepteur est constitué d'une partie détection optimale et d'une partie de remise en forme des données numériques reçues. Le détecteur optimal est réalisé à l'aide d'un intégrateur et d'un échantillonneur, synchronisés par le rythme digital, et d'un comparateur à seuil. La valeur du seuil dépend bien sûr de l'entropie de la source (elle est supposée maximum ; on considère donc que l'opération de codage de source a été effectuée) et de l'émetteur physique (émetteur bipolaire). Dans notre cas, la valeur du seuil est nulle. La partie remise en forme des données numériques est chargée de paralléliser les symboles binaires de façon à reconstituer l'information reçue.

#### 3.4. Le canal

Dans un souci de simplification, le canal est supposé idéal dans le sens où il n'est constitué que d'un fil de transmission. Cependant, il est tout à fait possible de choisir un modèle plus réel qui prendrait en compte une atténuation, une réponse en fréquence non uniforme et aussi les phénomènes d'échos et de trajets multiples.

De la même manière, rien n'interdit de compléter la chaîne de transmission en utilisant par exemple un filtre passe bas à la réception...

### 3.5. Mise en oeuvre sous Matlab/Simulink

La mise en oeuvre sous le logiciel Matlab/Simulink se fait en utilisant conjointement des blocs déjà prédéfinis sous Simulink et des Fonctions-S que l'on doit programmer. Bien que ces dernières soient d'un emploi moins aisé que les blocs, elles peuvent être utilisées avantageusement pour réaliser les opérations les plus complexes, telles que les interfaces série et parallèle. La figure 2 présente le schéma complet de la chaîne de transmission. Il faut noter que tous les éléments de la chaîne, les fonctions-S ainsi que les blocs, sont paramétrés selon un certain nombre de variables qui sont ici l'unité de temps **Tech** et le nombre de bits **lgmot** pour la représentation binaire des données d'information. Dans l'exemple présent **lgmot = 4**. La figure 3 présente à la fois le signal utile (à l'entrée du canal) et le signal bruité (à la sortie du canal). Ce type de signal en fonction du temps est obtenu avec l'oscilloscope sur la figure 2. Pour transmettre la totalité des 6 données, il est nécessaire de régler le « temps de simulation » à sensiblement **24.Tech**. Nous sommes ici au débit optimal.

Dans cet exemple, la puissance du bruit est de **0,5 W** et l'amplitude du signal est de **1 V**. Comme on peut le constater sur la figure 2, sur les **6** données émises, seules **4** ont été reçues sans erreurs. La seule manière ici d'améliorer la réception des informations est d'augmenter le rapport signal à bruit en sortie du canal.

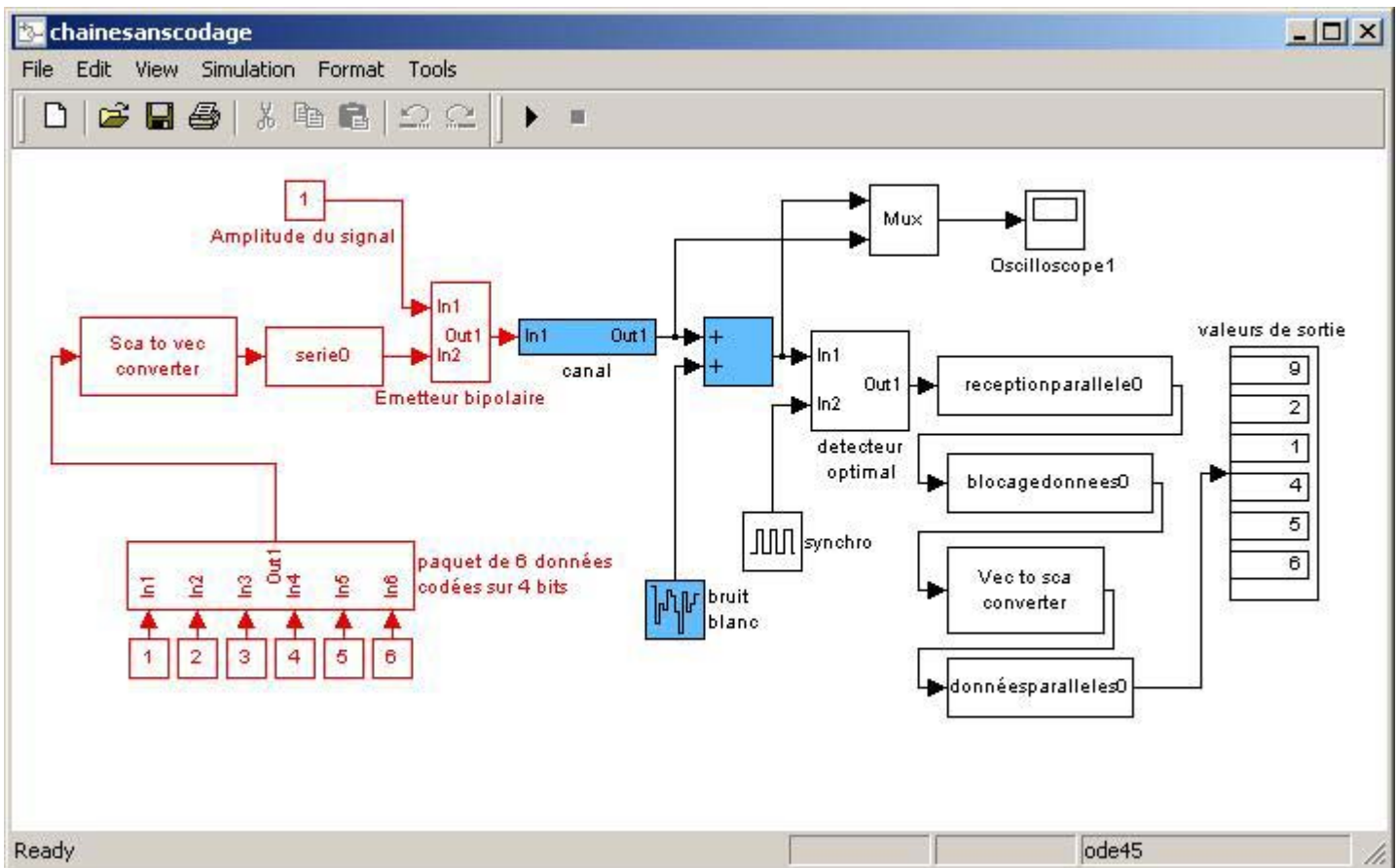


Fig. 2. Modèle de la chaîne de transmission sous Matlab/Simulink.

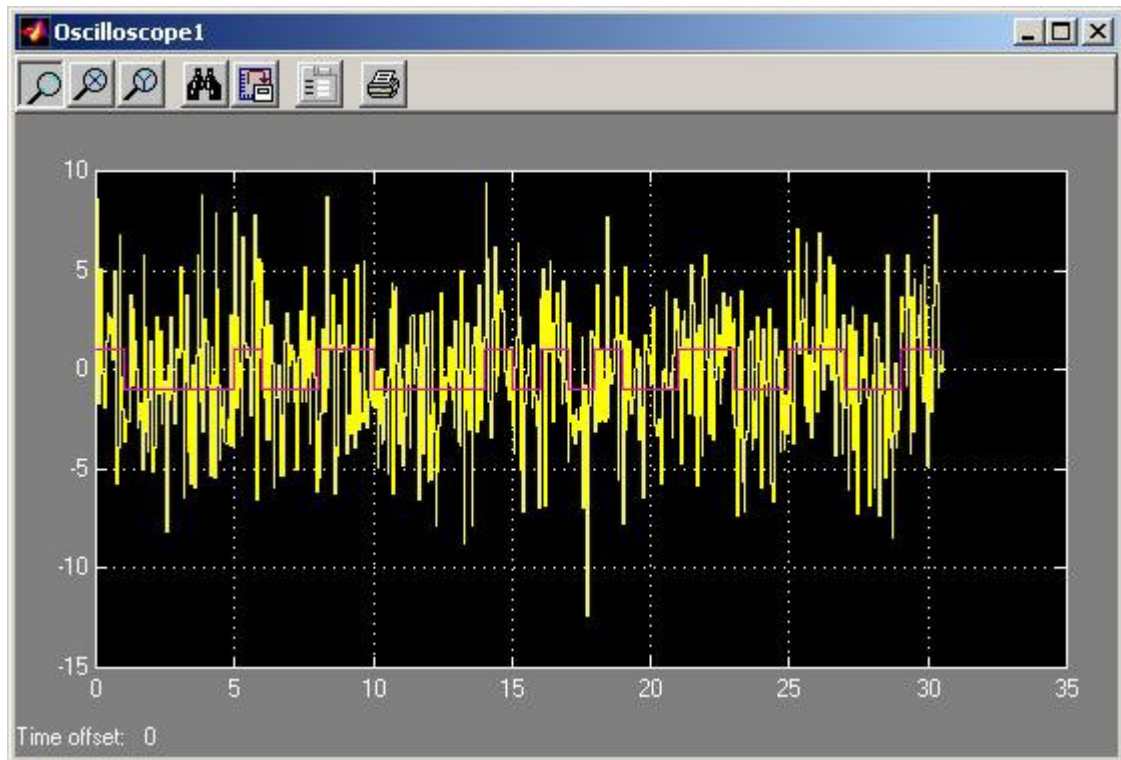


Fig. 3. Signal utile et signal bruité transitant sur le canal.

#### 4. Insertion des techniques de codage de canal

##### 4.1. Présentation générale

La chaîne de transmission numérique décrite dans la section précédente peut être réutilisée avantageusement car les modifications apportées par la présence des codeurs et décodeurs de canal seront mineures. Tout le problème se résume à prendre en compte le fait que le codage de canal introduit des bits supplémentaires au niveau de la représentation des données d'information. Par exemple, un codage de parité rajoute 1 bit de contrôle aux 4 bits d'information. Il faut donc en tenir compte tout au long de la chaîne de transmission et notamment au niveau des interfaces série et parallèle.

Ce problème de synchronisation étant réglé, on peut alors ajouter les blocs de codage et de décodage. Le schéma synoptique devient alors le suivant (Fig. 4).

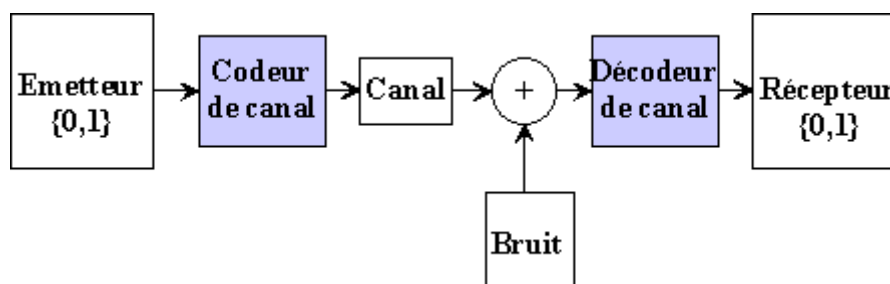


Fig. 4. Schéma synoptique général de la chaîne de transmission avec codage de canal.

#### 4.2. Code détecteur d'erreur (codeur de parité)

Il apparaît évident que le codage le plus connu est sans doute le codage de parité. La mise en oeuvre d'un tel codeur est très simple car il s'agit d'utiliser une simple porte *ou-exclusif* à 4 entrées, le décodage se faisant quant à lui en utilisant une seconde porte *ou-exclusif* à 5 entrées. La figure 5 présente le schéma de ces deux codeur et décodeur. La figure 6 présente une chaîne de transmission complète incluant cette technique de codage. Ce schéma ne se différencie de celui de la figure 2 que par la présence du codeur, du décodeur et d'un nouvel afficheur qui indique les erreurs éventuelles survenues au cours de la transmission et aussi par le fait que tous les éléments de la chaîne sont synchronisés sur une longueur de mot code égale à 5 (4 bits d'information et 1 bit de redondance). À ce niveau, l'étudiant voit tout de suite la dégradation du débit dans le sens où il doit augmenter le « temps de simulation » à sensiblement **30.Tech** s'il veut transmettre la même quantité d'information, que dans l'exemple précédent. Nous reprenons ici les mêmes caractéristiques de réglage, à savoir une amplitude des signaux de **1 V** et une puissance de bruit de **0,5 W**. Comme nous pouvons le constater, le résultat de la simulation indique deux erreurs, la première sur un bit d'information pour la donnée **2** et la seconde sur un bit de contrôle pour la donnée **4**. Dans ce dernier cas, bien qu'il y ait une erreur, la donnée utile reste correcte. Ceci illustre très bien la limitation d'un tel type de codage, car s'il autorise la détection des erreurs il ne permet pas leur correction.

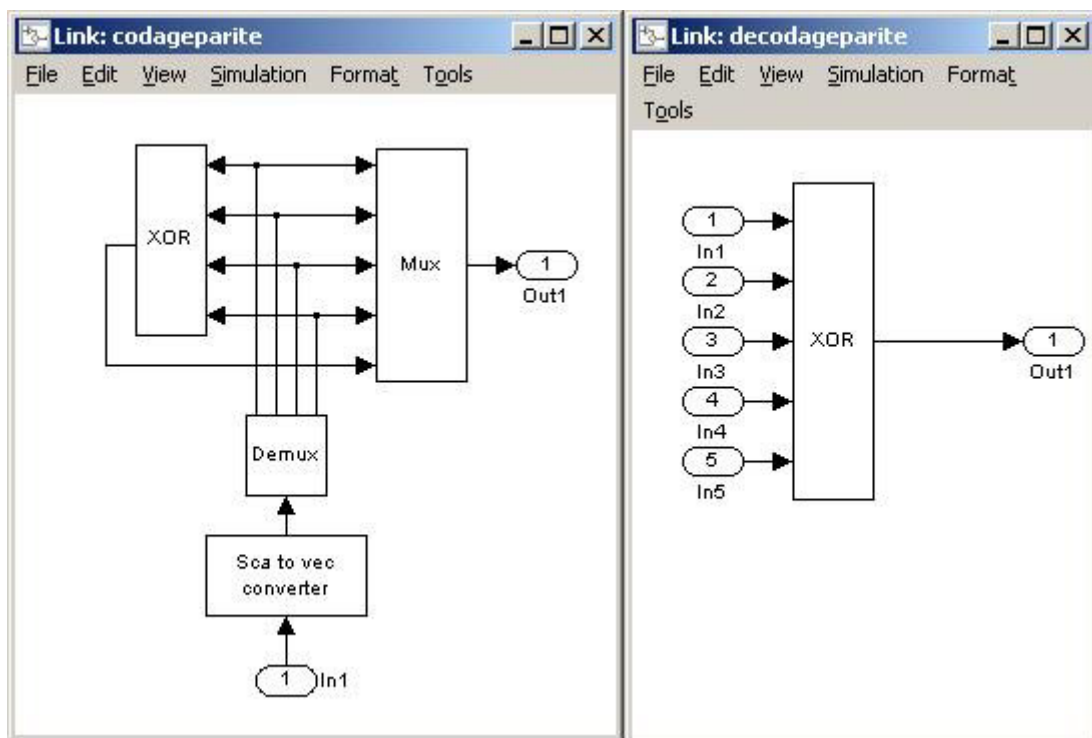


Fig. 5. Codeur et décodeur de parité.

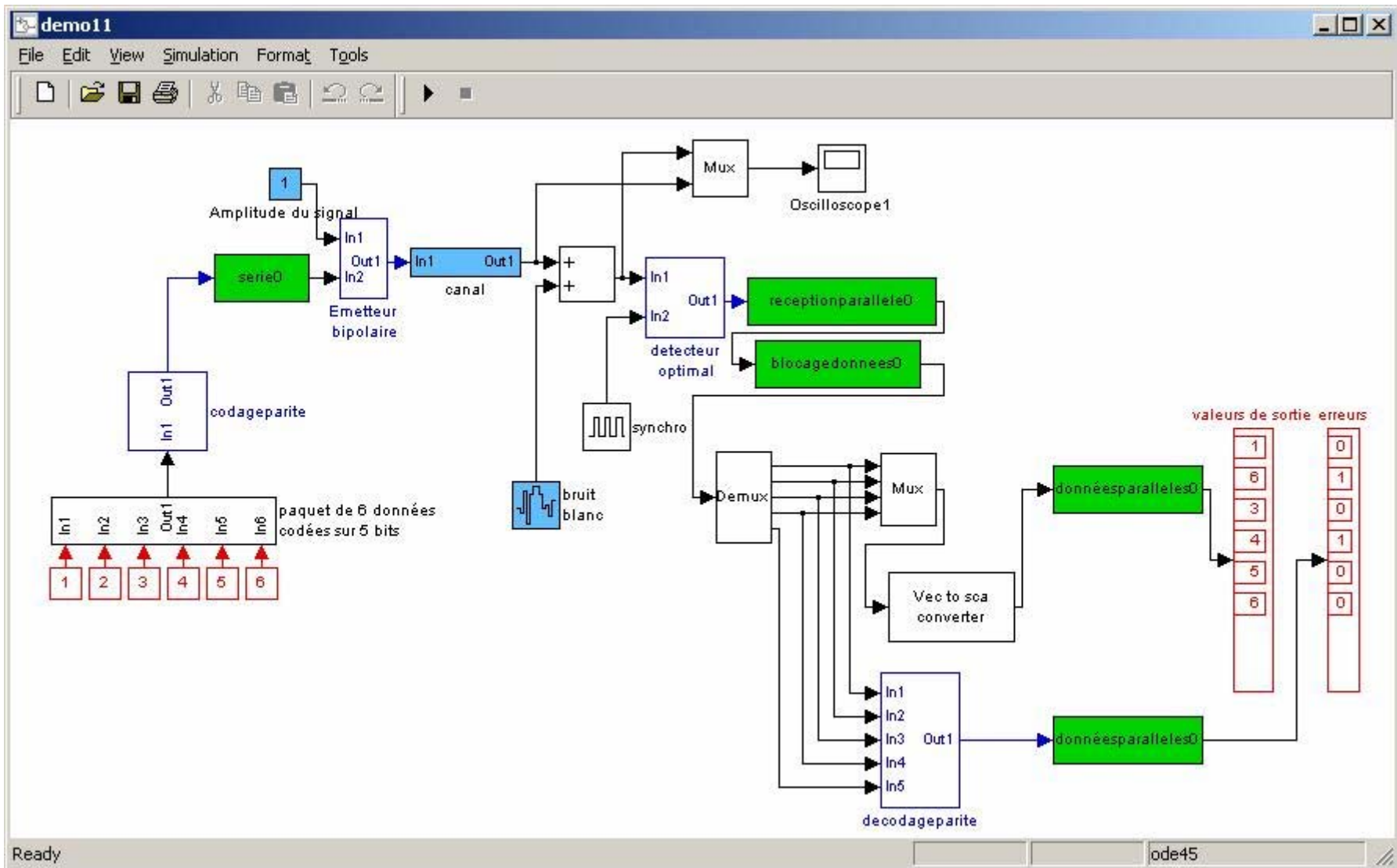


Fig. 6. Chaîne de transmission complète avec codage de parité.

### 4.3. Code correcteur d'erreur (codeur de Hamming)

L'étudiant peut donc continuer à progresser en mettant en oeuvre cette fois des codes correcteur d'erreur, l'exemple le plus connu étant sans doute le code de Hamming [9]. Il s'agit d'un code en bloc linéaire, basé sur le respect d'une distance minimale de 3 entre tous les mots du code et qui permet ainsi la localisation de toutes les erreurs simples. L'avantage d'un tel code est qu'il est facilement implémentable à l'aide de portes logiques et qu'il peut être décliné sous différentes formes suivant la matrice génératrice retenue (systématique ou non).

Par exemple, pour un code de Hamming **H(3,7)** non systématique dont une matrice génératrice est donnée ci-après, le schéma d'un codeur possible est présenté sur la figure 7.

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

éq 1 : matrice génératrice du code de Hamming **H(3,7)**.

Le schéma du codeur comprend simplement **3** portes logiques *ou-exclusif*. À l'entrée de ce codeur, sont connectés les **4** bits d'information et à la sortie de celui-ci nous obtenons le mot-code constitué des **4** mêmes bits d'information et de **3** bits de contrôle. La chaîne de transmission sera donc synchronisée cette fois à partir d'une longueur de mot code égale à 7.

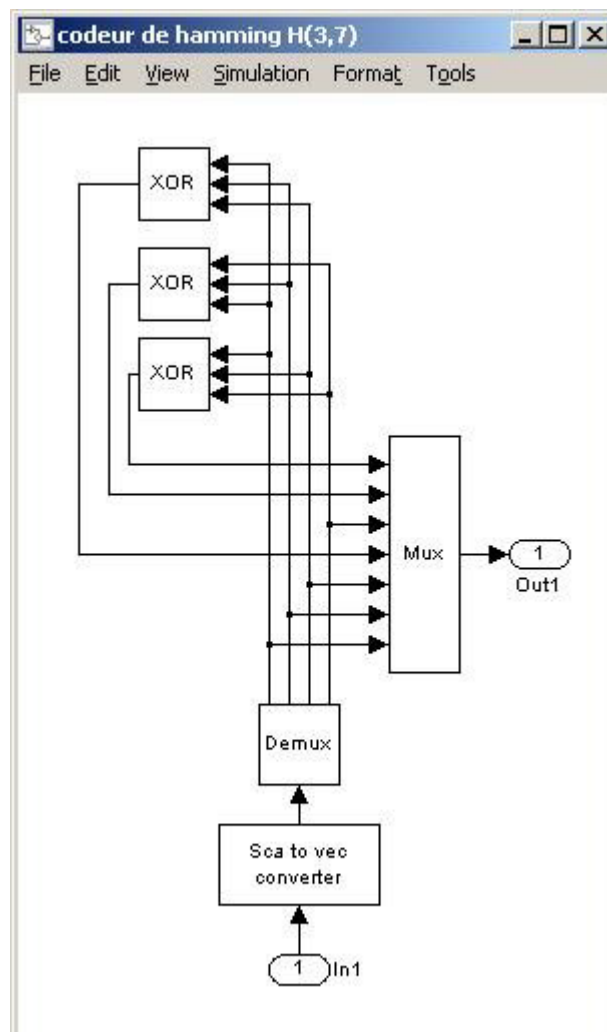


Fig. 7. Codeur de Hamming non systématique **H(3,7)**.



#### 4.4. Code correcteur d'erreur (décodeur de Hamming)

La correction des erreurs dans un tel code passe obligatoirement par la construction d'un syndrome d'erreur. Celui-ci sera construit très simplement à l'aide de 3 portes *ou-exclusif* comme dans le cas du codeur. Le détecteur est présenté figure 8. Le syndrome d'erreur est codé sur 3 bits et permet de localiser les bits erronés dans le mot code reçu. Dans le cas d'un syndrome d'erreur nul, on considère que la réception de l'information est correcte.

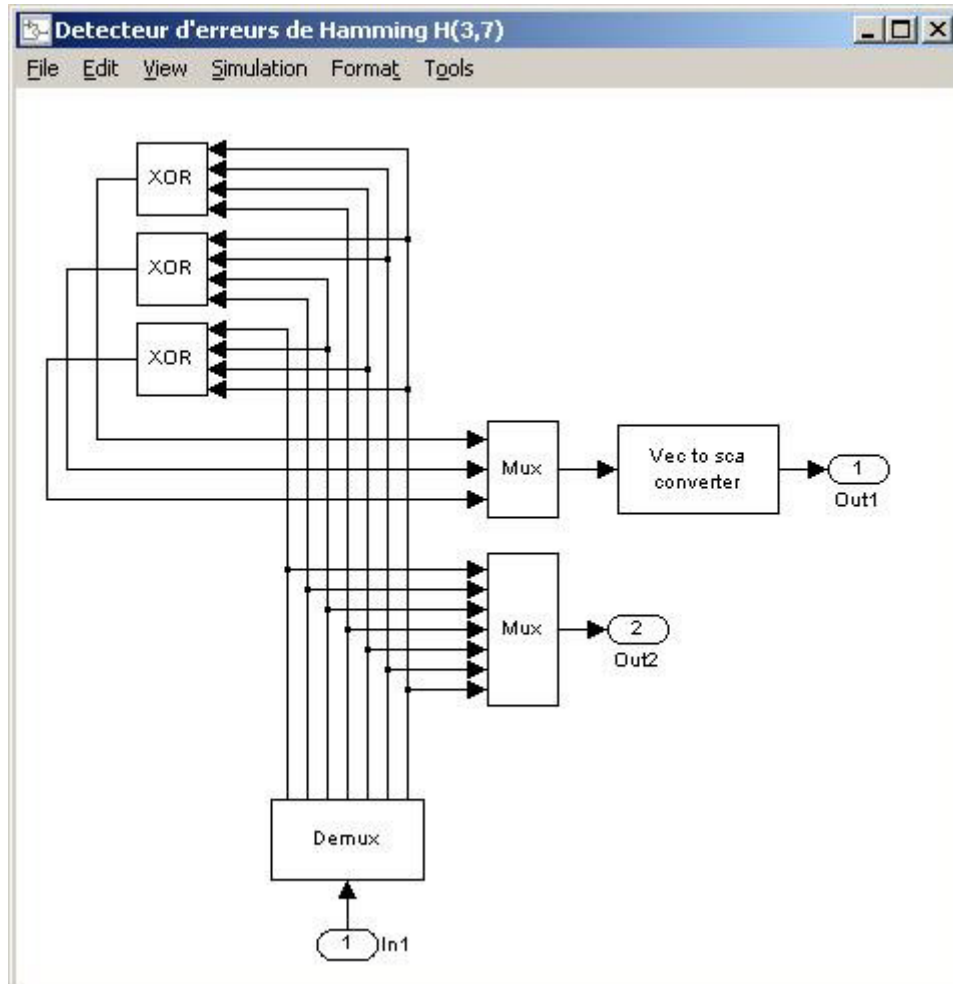


Fig. 8. Décodeur de Hamming non systématique  $H(3,7)$ .

#### 4.5. Code correcteur d'erreur (correcteur d'erreur de Hamming)

Le dernier maillon de l'opération de décodage est le correcteur d'erreur (Fig. 9). Il utilise le syndrome d'erreur calculé précédemment pour corriger le mot-code reçu. Il s'agit ici d'additionner ce dernier, avec un mot de même longueur dont le seul bit à 1 indique la position de l'erreur. Ce vecteur d'erreur est construit à partir de la valeur du syndrome d'erreur. La réalisation du correcteur nécessite quelques portes logiques, des portes *and* et *non* pour la construction du vecteur d'erreur et des portes *ou-exclusif* pour l'addition.

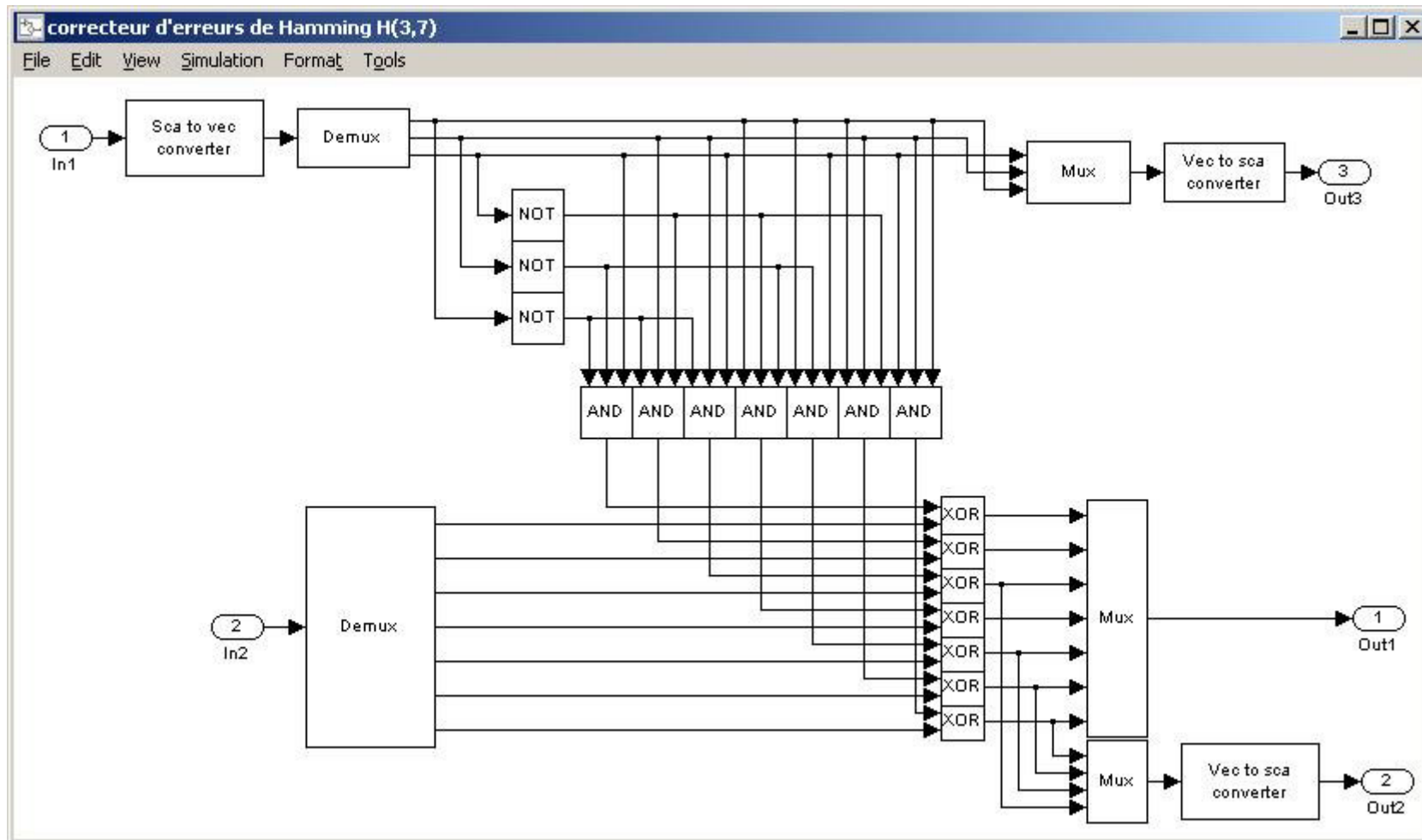


Fig. 9. Correcteur d'erreur de Hamming non systématique  $H(3,7)$ .

#### 4.6. Mise en oeuvre de la chaîne complète (codage de Hamming)

La chaîne complète intégrant le codeur, le décodeur et le correcteur est présentée à la figure 10. Ici on constate tout de suite l'intérêt de ce type de codage car bien que le détecteur ait détecté des erreurs (erreurs simples), celles-ci ont toutes été corrigées. Cette efficacité se paye bien sûr par une dégradation du débit car le temps de simulation est sensiblement de **42.Tech**.

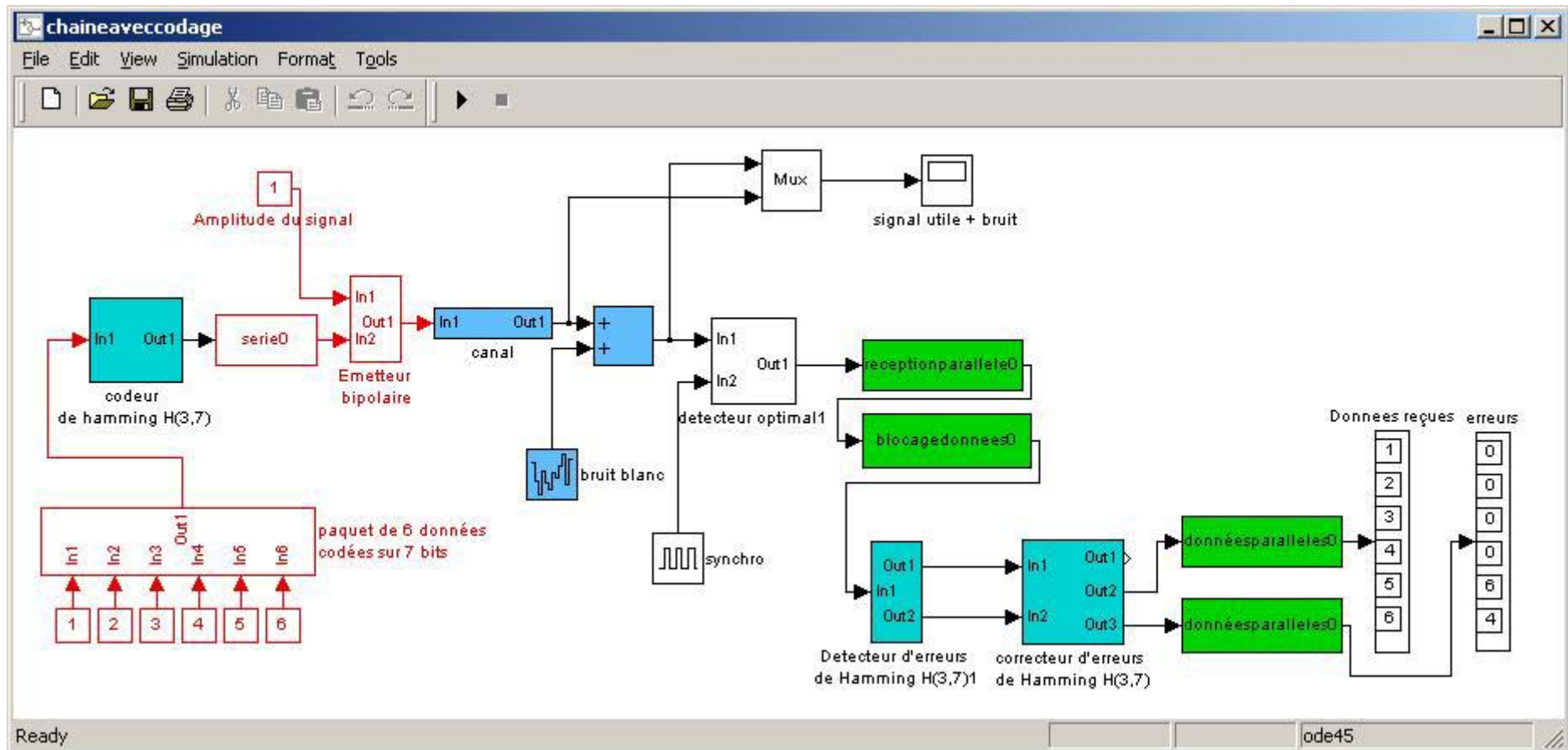


Fig. 10. Chaîne de transmission complète avec codage de Hamming.

## 5. Conclusion

Une telle manipulation est intéressante dans le sens où elle permet aux étudiants de bien visualiser les principes de base du codage de canal. Les possibilités offertes par une telle manipulation sont multiples dans le sens où il est possible de mettre en oeuvre et de vérifier l'efficacité de nombreux types de codage : codages de Hamming + parité, codage cyclique... Il est aussi possible de tester pour un même code des rendements différents. Finalement, nous avons présenté ici une ébauche de chaîne de transmission que l'on peut faire évoluer à souhait, en intégrant des composants plus proches de la réalité : canal, filtres, amplis, convertisseurs. On peut aussi envisager une mise sur porteuse. À ce titre, nous développons actuellement une manipulation similaire sur la base d'une modulation FSK radiofréquence. Cette manipulation est développée sur ADS-Agilent qui permet de simuler conjointement la partie numérique en bande de base et la partie hyperfréquence.

## Références bibliographiques

- [1] A. Spataru, *Fondement de la théorie de la transmission de l'information* (Presses polytechniques et universitaires romandes, 1987).
- [2] *Radiocommunications numériques*, sous la direction de G. Baudoïn (Dunod, 2002).
- [3] J.B. Anderson, *Digital transmission engineering* (Prentice Hall, 1998).
- [4] R.E. Blahut, *Digital Transmission of information*(Addison-Wesley, 1990).
- [5] S. Lin et D.J. Costello Jr., *Error control coding* (Prentice Hall, 1983).
- [6] J. B. Anderson et S. Mohan, *Source and channel coding* (Kluwer, 1993).
- [7] T. Oberg, *Modulation, detection and coding* (Wiley, 2001).
- [8] M. Joindot et A. Glavieux, *Communications numériques* (Masson, 1996).
- [9] R. W. Hamming, *Coding and information theory*, 2<sup>e</sup> édition (Prentice Hall PTR, 1985).

## Annexe : Licence d'utilisation du (ou des) logiciels joints aux articles de J3eA

L'auteur (ou les auteurs) concède au lecteur de l'article un droit d'usage privé, non collectif et non exclusif, pour le logiciel, ainsi qu'un droit d'usage pour son enseignement en mentionnant le nom des auteurs du ou des logiciels. La reproduction du logiciel pour un usage privé ou à des fins pédagogiques sans but lucratif est autorisée à condition de mentionner les auteurs et de reproduire cette licence d'utilisation. La modification du logiciel est interdite sans l'accord préalable et expresse de l'auteur (ou des auteurs) du logiciel.

Toute diffusion, toute exploitation commerciale du logiciel, devra être demandée à l'auteur (ou aux auteurs) du logiciel.

L'auteur (ou les auteurs), ainsi que l'éditeur de la revue J3eA, décline toute responsabilité quant au téléchargement et à l'utilisation du logiciel fourni.

L'éditeur de la revue J3eA n'assure aucun support technique pour les logiciels joints aux articles de J3eA.

---

## Télécharger

- l'archive demov2ri.tar contenant les fichiers Matlab/Simulink **chaineavecscodage** et **chainesanscodage**, les autres fichiers sont des blocs et des fonctions-S nécessaires au bon fonctionnement de la démonstration ;
- le guide d'utilisation de la démonstration.

## Installation

1. décompresser l'archive demov2ri.tar ;
2. lancer matlab ;
3. suivre les indications du guide d'utilisation de la démonstration.