

МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ КАНАЛА СВЯЗИ В СРЕДЕ МАТЛАБ/SIMULINK

Арстанбеков Кайрат Арстанбекович, специалист СКБ ИЭТ при КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Мира 66, e-mail: kairatluxury@mail.ru

Кармышаков Аскарбек Камалдинович, к.т.н., доцент, КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Мира 66, e-mail: askar1969@mail.ru

Целью данной работы является моделирование и исследование помехоустойчивости канала связи с использованием известного пакета Matlab/Simulink. Для достижения поставленной цели решались задачи:

- исследование основных методов кодирования для систем передачи информации;
- моделирование каналов связи в пакете программ Matlab+Simulink;
- получение осциллограмм сигналов;
- сравнительный анализ и обработка результатов моделирования.

Ключевые слова: кодирование, спектр сигналов, цифровая информация, система передачи информации, синхронизация сигналов, битовые ошибки.

MODELING AND INVESTIGATION OF INTERFERENCE CODING OF THE COMMUNICATION CHANNEL IN THE MEDIUM MATLAB / SIMULINK

Arstanbekov Kairat Arstanbekovich, specialist SDB IET at KSTU named after I.Razzakova, Kyrgyzstan, 720044, Bishkek, Kyrgyzstan, e-mail: kairatluxury@mail.ru

Karmyshakov Askarbek Kamaldinovich, PhD (Engineering), Associate professor, KSTU named after I.Razzakova, Kyrgyzstan, 720044, Bishkek, Kyrgyzstan, e-mail: askar1969@mail.ru

The purpose of this work is to simulate and study the noise immunity of the communication channel using the well-known Matlab / Simulink package. To achieve this goal, the following tasks were solved:

- The study of basic coding methods for information transmission systems;
- Simulation of communication channels in the Matlab/Simulink software package;
- Reception of oscillograms of signals;
- Comparative analysis and processing of simulation results/

Keywords: coding, spectrum of signals, digital information, information transmission system, signal synchronization, bit errors.

Исследование основных методов кодирования передачи информации

При обработке, передаче и приеме цифровой информации важное значение имеет определение (детектирование) ошибок и их коррекция. При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей:

- минимизировать ширину спектра сигнала, полученного в результате кодирования;

- обеспечивать синхронизацию между передатчиком и приемником;
- обеспечивать устойчивость к шумам;
- обнаруживать и по возможности исправлять битовые ошибки;
- минимизировать мощность передатчика.

В работе будут проведены следующие виды моделирования:

1. Моделирование кодера Хэмминга.
2. Моделирование кодера BCH
3. Моделирование кодера Рида-Соломона.
4. Моделирование сверточного кодера.

№	Тип кодера	Данные	K	N	Кратность ошибок T
1.	Хэмминга	Двоичные	5	15	0, 5, 8
2.	БХВ	Двоичные	5	15	0, 5, 8
3.	Рида-Соломона	Целые числа			
4.	Сверточный	Двоичные			

Рассмотрим несколько примеров.

Задание 1. Выполнить моделирование кодера Хэмминга (11,15) без ошибок и с ошибками с кратностью T1=4, T2=5.

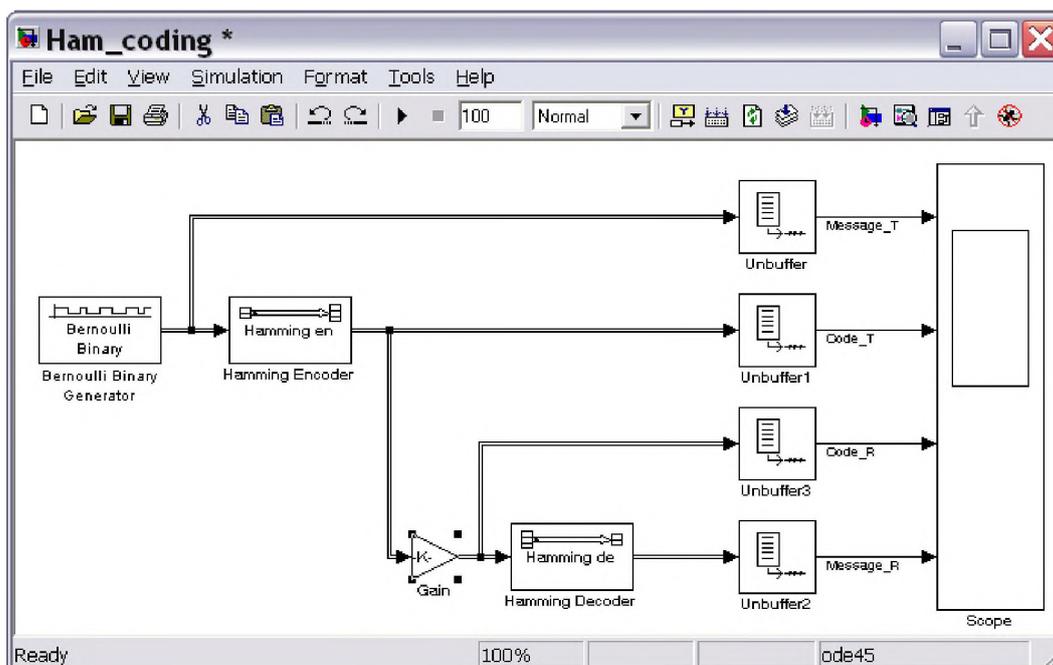


Рис. 1. Имитационная модель кодера Хэмминга

В модель включаем:

Источник данных, генератор случайных двоичных чисел – Bernoulli Binary-Integer, вырабатывающий случайные биты. Выход генератора нужно сделать в виде кадра из K=11 отсчетов.

Кодер Хэмминга – Hamming encoder. Он по сообщению K формирует передаваемый код V. В окне параметров блока задаем N=15.

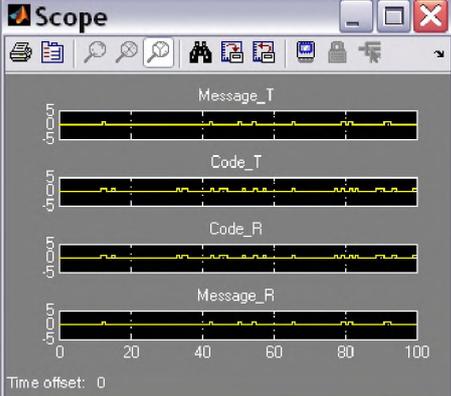
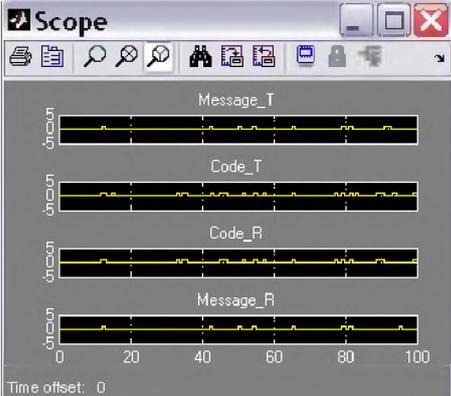
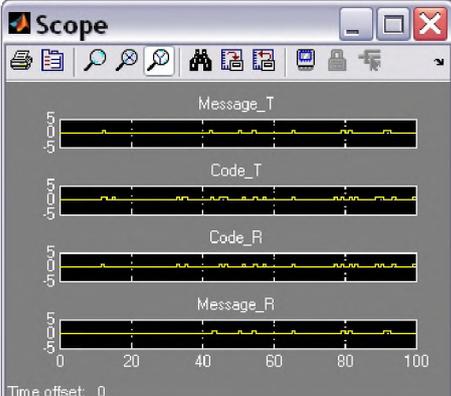
Умножитель Gain. Он умножает выход кодера на вектор ошибки E из 0 и 1, имитирующий ошибки в канале связи (ошибка может получиться при компоненте E, равном 0). Вектор E должен содержать N компонент, которые вычислим функцией $rot90(randerr(N,1,N-ER))$. Получим вектор-строку из N равновероятных нулей и единиц, из которых N-ER равны 1.

Декодер Хэмминга – Hamming decoder. Для него нужно задать параметры, аналогичные кодеру.

Блоки Unbuffer преобразуют кадры в линейные последовательности, направляемые к виртуальному 4-входовому наблюдателю Scope.

Наблюдатель Scope позволяет в общем масштабе времени наблюдать сигналы от блоков Buffer. Чтобы в графиках наблюдателя были заголовки, сделаем входам имена. Для этого двойным щелчком по линии входа вызовем окно ввода имени, в которое занесем имя. зададим имена: для передаваемого сообщения – Message_T, для передаваемого кода Code_T, для принятого кода – Code_R, для декодированного сообщения Message_R.

При моделировании с разными кратностями ошибок получим:

	<p>Ошибок нет, ER=0</p>
	<p>Ошибка кратности T1=ER=4. Все ошибки исправлены.</p>
	<p>Ошибка кратности T2=ER=5. Часть ошибок осталась.</p>

Задание 2. Выполнить моделирование кодека BCH (5,15) без ошибок и с ошибками с кратностью T1=4, T2=5.

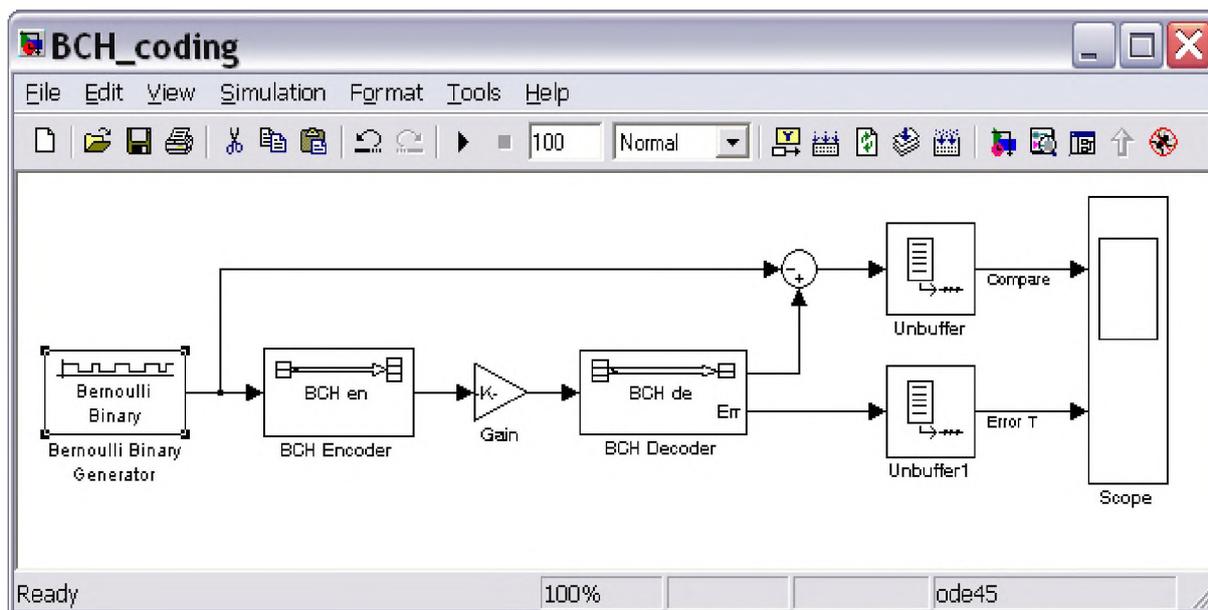


Рис. 2. Имитационная модель кодера BCH

В модель включаем:

Источник данных, генератор случайных двоичных чисел – Bernoulli Binary-Integer, вырабатывающий случайные биты. Выход генератора нужно сделать в виде кадра из $K=5$ отсчетов.

Кодер BCH – DCY encoder. Он по сообщению K формирует передаваемый код V . В окне параметров блока задаем $N=15$, $K=5$.

Умножитель Gain. Он умножает выход кодера на вектор ошибки E из 0 и 1, имитирующий ошибки в канале связи (ошибка может получиться при компоненте E , равном 0). Вектор E должен содержать N компонент, которые вычислим функцией $\text{rot90}(\text{randerr}(N,1,N-ER))$. Получим вектор-строку из N равновероятных нулей и единиц, из которых $N-ER$ равны 1.

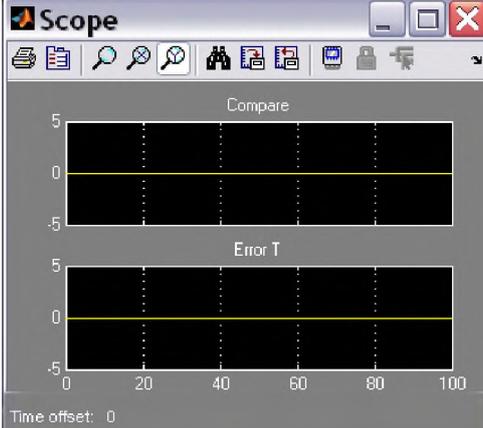
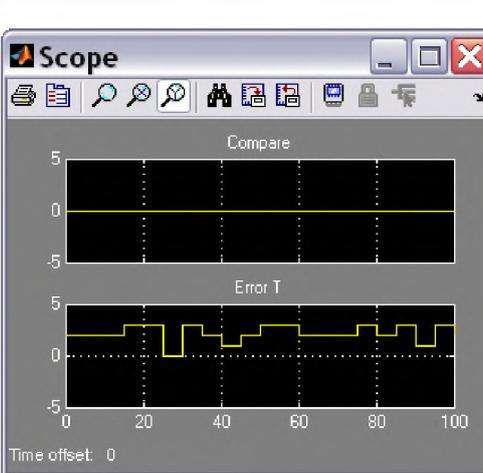
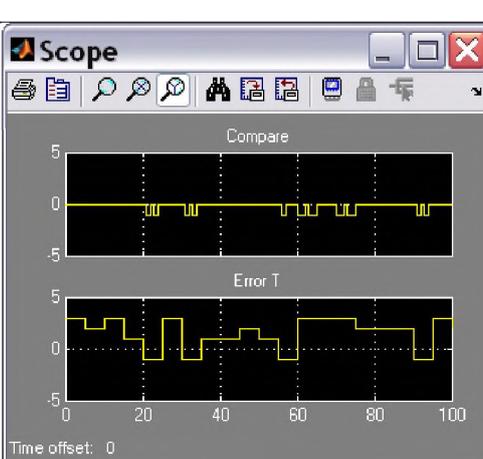
Декодер BCH – BCH decoder. Для него нужно задать параметры, аналогичные кодеру. Декодер имеет два выхода: верхний декодированное сообщение, нижний – кратность исправленных ошибок.

Блок Sum в котором путем вычитания сравниваются исходное и декодированное сообщение. В окне параметров блока нужно задать знаки операндов минус и плюс.

Блоки Unbuffer преобразуют кадры в линейные последовательности, направляемые к виртуальному 2-входовому наблюдателю Scope.

Наблюдатель Scope позволяет в общем масштабе времени наблюдать сигналы от блоков Buffer. Чтобы в графиках наблюдателя были заголовки, сделаем входам имена. Для этого двойным щелчком по линии входа вызовем окно ввода имени, в которое занесем имя. зададим имена: для сигнала сравнения – Compare, для передаваемого кода Code_T, для кратности исправленных ошибок – Error_T.

При моделировании с разными кратностями ошибок получим:

	<p>Ошибок нет, ER=0</p>
	<p>Ошибка кратности T1=ER=4. Все ошибки исправлены.</p>
	<p>Ошибка кратности T2=ER=5. Часть ошибок осталась.</p>

Задание 3. Выполнить моделирование кодека Рида-Соломона RS(5,15) без ошибок и с ошибками с кратностью T1, T2.

Создаем модель в среде Simulink. В модель (Рис.3) включаем:

Источник данных, генератор случайных целых чисел – Random-Integer, вырабатывающий числа в интервале от 0 до N-1. Выход генератора нужно сделать в виде кадра из K чисел.

Кодер Рида-Соломона – RS encoder. Он по сообщению K формирует передаваемый код из N символов.

Умножитель Gain. Он умножает выход кодера на вектор ошибки E из 0 и 1, имитирующий ошибки в канале связи (ошибка может получиться при компоненте E, равном 0). Вектор E должен содержать N компонент, которые вычислим функцией $\text{rot90}(\text{randerr}(N,1,N-ER))$. Получим вектор-строку из N равновероятных нулей и единиц, из которых N-ER равны 1.

Декодер Рида-Соломона – RS decoder, имеющий два выхода. На верхнем выходе из принятого кода R формируется декодированное сообщение, на нижний выход выводится кратность исправленных ошибок.

Сумматор Sum. В нем из декодированного сообщения вычитается исходное сообщение.

Блоки Unbuffer преобразуют кадры в линейные последовательности, направляемые к виртуальному двухвходовому наблюдателю Scope.

Наблюдатель Scope позволяет в общем масштабе времени наблюдать сигналы от блоков Buffer.

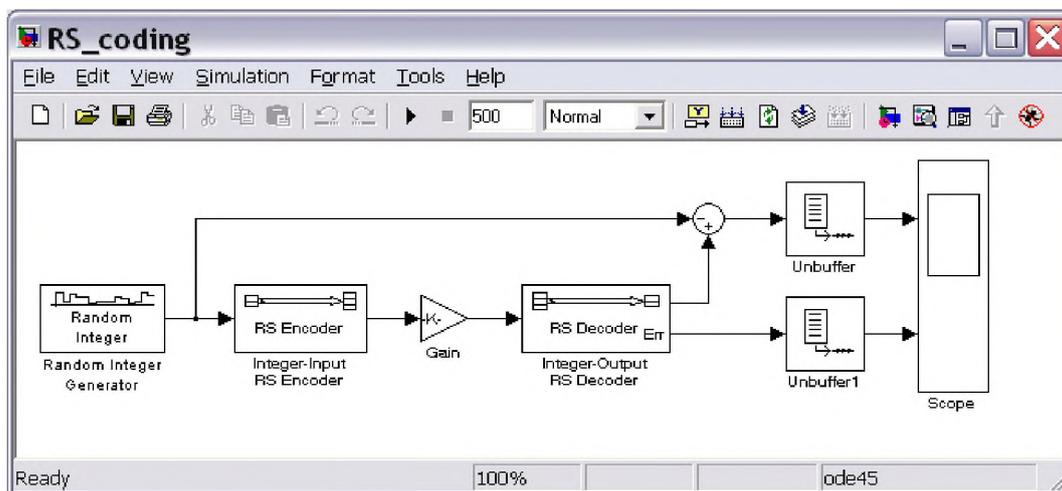
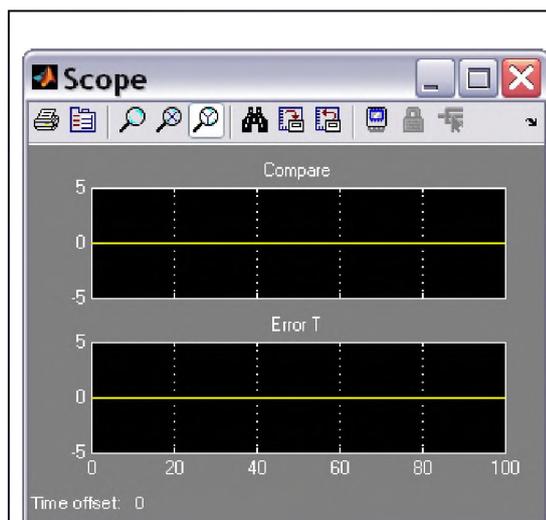


Рис. 3. Имитационная модель RS кодека

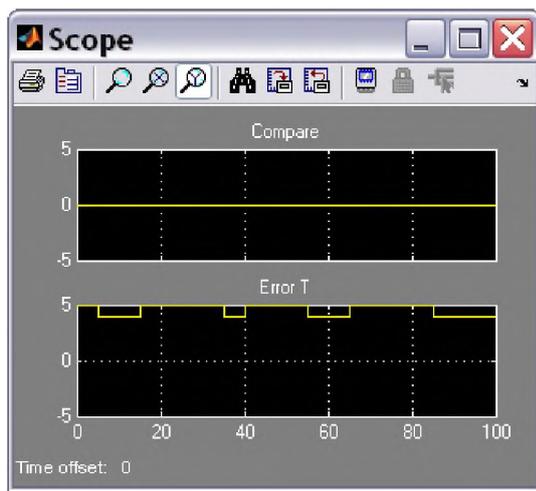
После размещения блоков в модели их нужно соединить. Для этого просто рисуем соединительную линию от выхода одного блока к входу другого.

Далее нужно задать свойства блоков. Для каждого блока при двойном щелчке по нему вызывается окно параметров, в котором нужно разместить нужные значения. Некоторые параметры установлены по умолчанию.

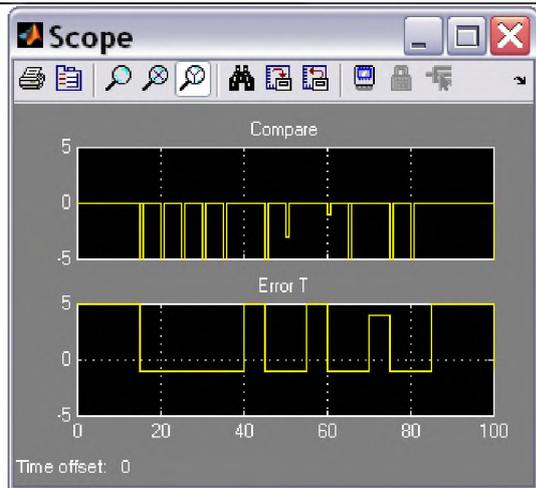
После моделирования можно двойным щелчком по блоку Scope посмотреть результаты:



Ошибок в канале нет. При этом $Gain=rot90(randerr(15,1,15))$. В результате нет ни ошибок, ни исправлений.



Ошибки в канале есть, их кратность T1 не превышает исправляющей способности кода. При этом $Gain=rot90(randerr(15,1,10))$ В результате ошибок нет, а кратность исправленных ошибок меняется до 5.



Ошибки в канале есть, их кратность T2 может превысить исправляющую способность кода. При этом $Gain=rot90(randerr(15,1,9))$ В результате ошибки есть, разница между исходным и декодированным сообщением достигает 5. Часть ошибок исправлена, их кратность меняется до 5.

Задание 4. Выполнить моделирование сверточного кода 2/3 без ошибок и с ошибками с кратностью T1, T2.

Создаем модель в среде Simulink (Рис. 4).

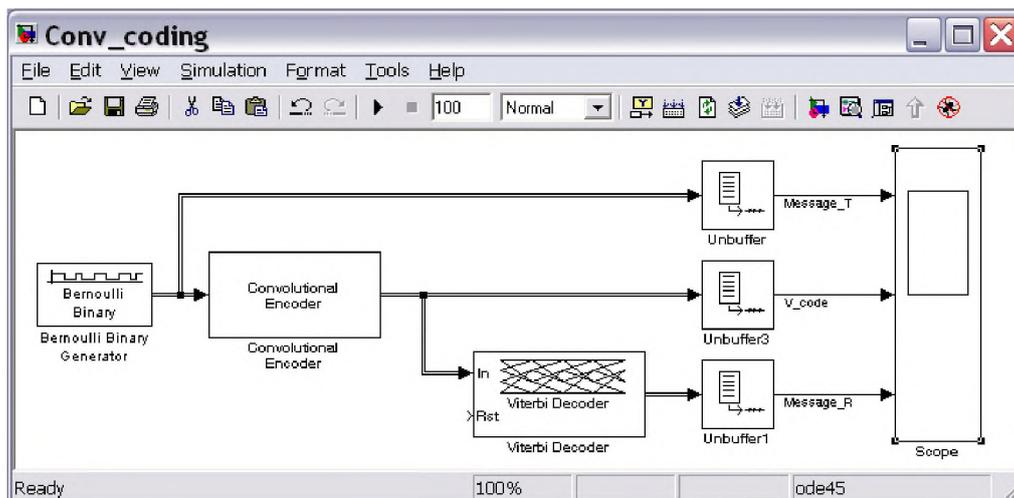


Рис. 4. Имитационная модель сверточного кода

В модель включаем:

Источник данных, генератор случайных двоичных чисел – Bernoulli Binary-Integer, вырабатывающий случайные биты. Выход генератора нужно сделать в виде кадра из $K=2$ отсчетов.

Сверточный кодер – Convolution encoder. Он по сообщению из $K=2$ символов формирует передаваемый код V из $N=3$ символов. В окне параметров кодера нужно задать функцию генерации решетки `poly2trellis([4 3],[4 5 17; 7 4 2])`.

Сверточный декодер – Convolution decoder. Он декодирует сообщение, используя алгоритм Витерби. В окне параметров блока нужно повторить решетку, выбрать декодирование с жестким решением (Hard Decision) и глубину отслеживания назад примерно $3 \cdot (K+N)$.

Блоки Unbuffer преобразуют кадры в линейные последовательности, направляемые к виртуальному 3-входовому наблюдателю Scope.

Наблюдатель Scope позволяет в общем масштабе времени наблюдать сигналы от блоков Buffer. Чтобы в графиках наблюдателя были заголовки, сделаем входам имена. Для этого двойным щелчком по линии входа вызовем окно ввода имени, в которое занесем имя. зададим имена: для передаваемого сообщения – `Message_T`, для передаваемого кода `V_Code`, для декодированного сообщения `Message_R`.

После моделирования можно двойным щелчком по блоку Scope посмотреть результаты (Рис.5):

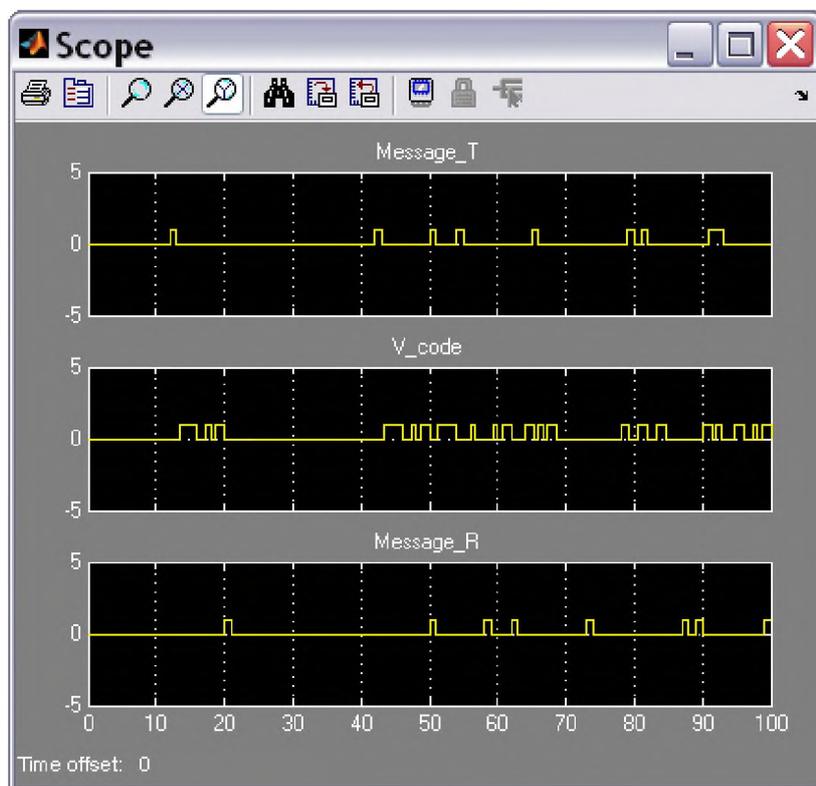


Рис. 5. Результаты моделирования сверточного кодера 2/3

Декодированное сообщение совпадает с исходным. Есть сдвиг во времени.

Список литературы

1. Васин. В.А. Радиосистемы передачи информации. М.: Горячая линия - Телеком, 2005. 472 с.
2. Дьяконов В. MATLAB 6.5 SP1/7.0+Simulink 5/6. Основы применения. М.: СОЛОН-Пресс, 2005.
3. Дьяконов В. MATLAB 6.5 SP1/7.0+Simulink 5/6. Обработка сигналов и проектирование фильтров. М.: СОЛОН-Пресс, 2005.