**Указания при изпълнение на задача № 2.**

Създаденият и проверен за правилна работа, при изпълнение на задача №1 от втората част на курсовия проект, симулационен модел с име Course Project Part 2.mdl. се записва с ново име, например Course Project Part 3.mdl.

В съответсвие с поставените изсквания в задача №1 от втората част на курсовия проект е необходимо да се промени вида на предаваната цифрова информация и вместо използваната в първата част на проекта случайна двоична последователност от данни, да се използва при предаването реален поток от избрани тестови аудио сигнали. За целта е необходимо блока за генериране на случайна двоична последователност от данни да се замени в модела Course Project Part 3.mdl с нов блок, който да създава също така двоична последователност от данни, но тя да съдържа информация за реален аудио сигнал. Превръщането на аудио сигналите в подходяща за предаване чрез симулационния модел двоична последователност от данни е необходимо да се извърши в следната последователност:

1. Въвеждане в работното пространство на Matlab (Workspace) на предварително записан аудио файл (във формат Wave);
2. Преобразуване на дискретните стойности на въведения като едномерен вектор аудио сигнал от десетичен в двоичен код (например десетичното число със стойност 56 в 8 –разрядно двоично число в случая 00111000).
3. Преобразуване на получените в двоичен код едномерни вектори на аудио сигнала от формат на 8-разрядни двоични числа в двоична последователност от едноразрядни двоични числа, които да бъдат използвани в разработвания симулационен модел.

Изпълнението на горепосочените действия може да се извърши чрез предварително (преди стартирането на модела от тази точка на заданието) изпълнение на следната съответна входна програма в работното пространство (Workspace) на Matlab:

За случая на преобразуване на дискретните стойности на въведения като едномерен вектор аудио сигнал от десетичен в двоичен код се използва следната програма (например с име sig\_in.m), която следва да бъде съставена в Matlab:

fd=uigetfile('\*.wav'); % Audio file open

[sigin fs] = wavread(fd); % Audio file read

lsigin=length(sigin); % Audio signal length calculation

sigin=sigin\*127+127; % Audio signal format from double to integer transform

sigin=fix(sigin); % Audio signal format from double to integer transform

plot(sigin) % Audio signal visualization

title('Input Audio Signal')

pause(3) % Wait 3 seconds for audio signal visualization

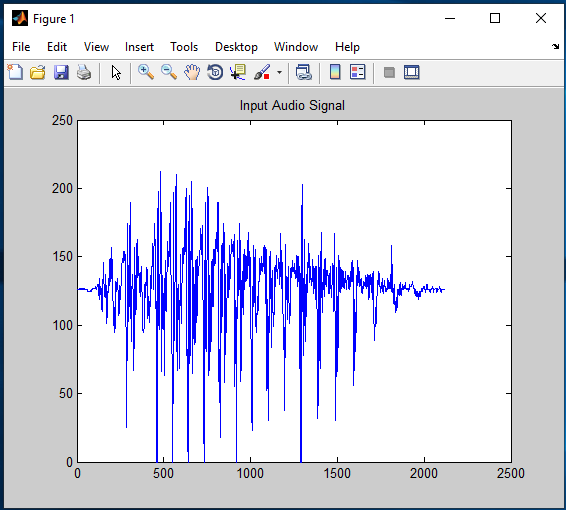
sound(sigin,fs) % Play Input Audio signal

siginby=de2bi(sigin); % Audio signal format from integer to byte transform

siginbi=reshape(siginby,lsigin\*8,1) ;% Audio signal format from byte to bit transform

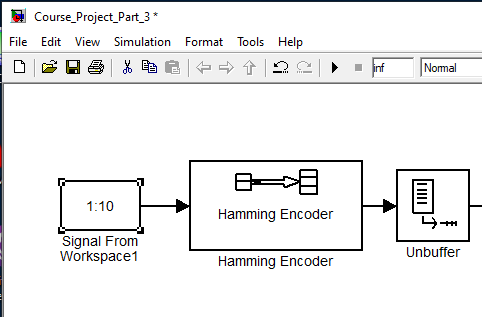
lsiginbi=length(siginbi); % Calculation of length of audio signal in bit format

След стартиране на програмата (sig\_in.m) се избира тестов аудио сигнал. В случая, за пример е избран тестов сигнал (AA.wav). След изпълнение на програмата (sig\_in.m) се визуализира времедиаграмата на избрания тестов аудио сигнал (AA.wav), показана на фиг.1, а в работното пространство (Workspace) на Matlab се създава едномерен вектор siginbi. Също така е възможно този аудишо сигнал да се прослуша. В създадения едномерен вектор информацията за въведеният аудио сигнал (AA.wav) е в двоичен код и представлява двоична последователност от едноразрядни двоични числа, които могат да бъдат използвани в разработвания симулационен модел и предавани по комуникационен канал за връзка.



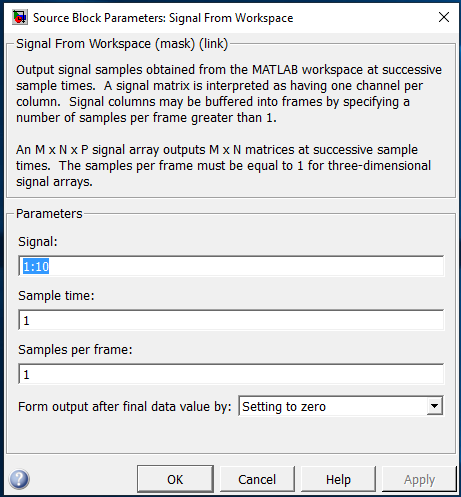
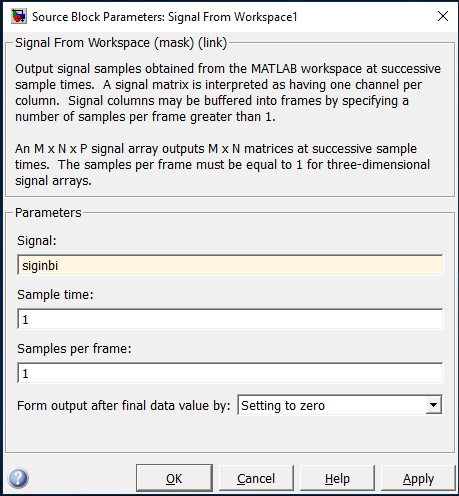
Фиг. 1. Времедиаграма на избран тестов аудио сигнал (AA.wav) след изпълнението на програмата (sig\_in.m)

Прехвърлянето на данните от създадения едномерен вектор siginbi от работното пространство (Workspace) на Matlab в симулационния модел Course Project Part 3.mdl следва да се извърши като в симулационния модел се замени блока Random Integer Generator с блок Signal From Workspace и се свърже към следващия съществуващ блок в симулационния модел, а именно блока за кодиране на канала с цел откриване на грешки. На Фиг. 2 е представен пример за извършване на тези действия, при които след прибавения нов блок Signal From Workspace следва използвания в примера блок Hamming Encoder.



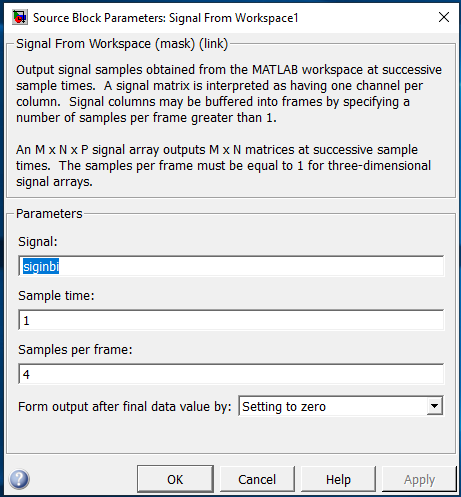
Фиг. 2. Пример за прибавяне на нов блок Signal From Workspace, след който следва използвания в примера блок Hamming Encoder.

След прибавянето на блока Signal From Workspace в симулационния модел Course Project Part 3.mdl, за неговата правилна работа, следва да се промени първоначалното наименование 1:10 (виж фиг.2) на блока Signal From Workspace. За целта чрез двукратно кликване върху блока Signal From Workspace се отваря прозорец с неговите параметри, както е показано на фиг.3 вляво, а след това се задава името siginbi на едномерния вектор, както е показано на фиг.3 вдясно. По този начин аудио информацията за въведеният аудио сигнал (AA.wav) под формата на двоична последователност от едноразрядни двоични числа в двоичен код се прехвърля от работното пространство (Workspace) на Matlab с наименование siginbi в блока Signal From Workspace симулационния модел Course Project Part 3.mdl.

Фиг. 3. Чрез двукратно кликване върху блока Signal From Workspace се отваря прозорец с неговите параметри, както е показано на фиг.3 вляво, а след това се задава името siginbi на едномерния вектор, както е показано на фиг.3 вдясно.

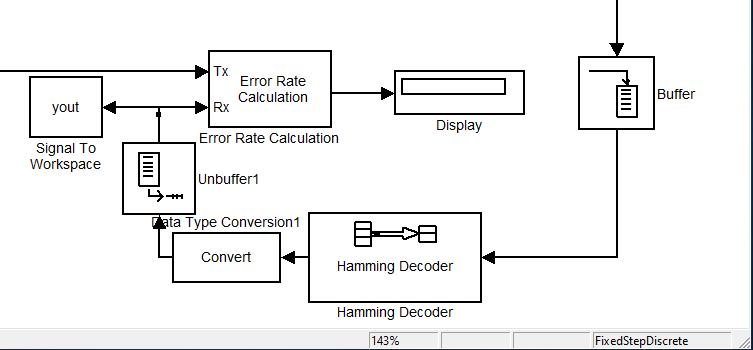
Необходимо е също така режимът на блока Signal From Workspace да се промени, като се зададе броят на дискретните стойности чрез параметъра Samples per frame. Това е показано на Фиг. 4 за пример на задание на кодер от типа Hamming Encoder във втората част на курсовия проект. За целта отново чрез двукратно кликване с мишката върху блока кодер Signal From Workspace се отваря прозореца с неговите параметри и в полето Samples per frame се въвежда (за разглеждания пример с кодер Hamming Encoder) стойността Samples per frame = 4, с цел съгласуване на дължината на изходните данни от блока Signal From Workspace с дължината на входните данни Message Length K=4 на кодера Hamming Encode, в съответствие с връзката между тях показана на фиг.2.



Фиг. 4. Промяна на режима на блока Signal From Workspace, като се задава броят на дискретните стойности чрез параметъра Samples per frame = 4, както това е показано за пример на задание на кодер от типа Hamming Encoder във втората част на курсовия проект, така че да се съгласува дължината на изходните данни от блока Signal From Workspace с дължината на входните данни Message Length K=4 на кодера Hamming Encoder

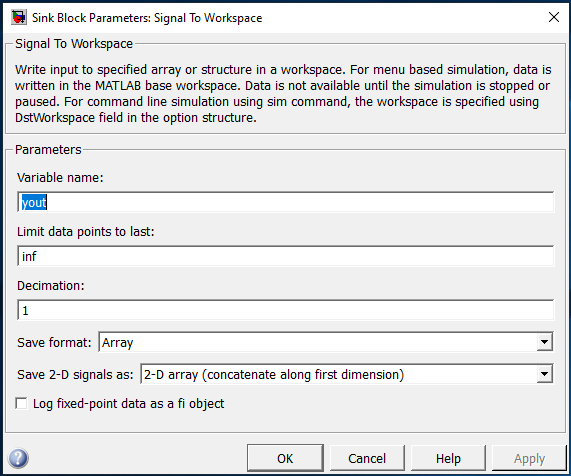
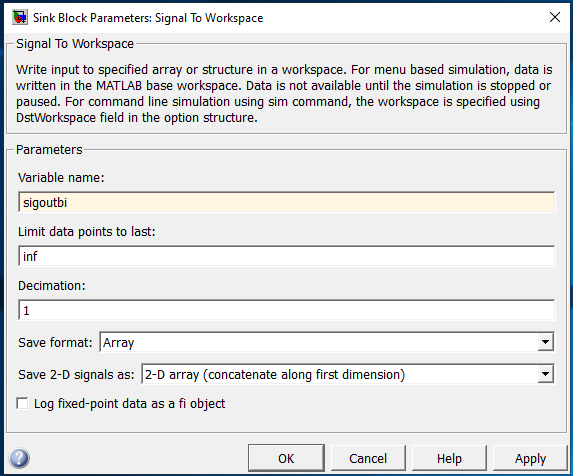
Описаните по-горе действия за промяна на вида на предаваната входна цифрова информация от случайна двоична последователност данни в предаване на реален поток от избрани тестови аудио сигнали следва да се приложат и в изходната част на симулационния модел по подобен начин, но в обратен ред, т.е. превръщането на приетата двоична последователност от данни за аудио сигналите, обратно в едномерен вектор на приетия аудио сигнал.

За целта първо следва в модела Course Project Part 3.mdl да се прибави блок Signal To Workspace с първоначално наименование yout, както е показано за пример на фиг. 5. В този блок се записва приетата и декодирана двоична последователност от данни за аудио сигналите, за да се прехвърли от този блок на модела в работното пространство (Workspace) на Matlab.



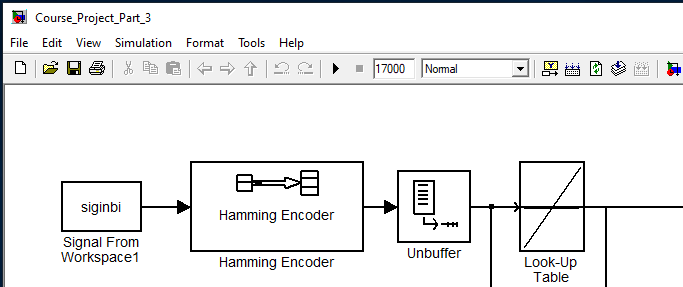
Фиг. 5. Прибавяне на нов блок Signal To Workspace с първоначално наименование yout в модела Course Project Part 3.mdl, за да се прехвърли приетата и декодирана двоична последователност от данни за аудио сигналите от този блок на модела в работното пространство (Workspace) на Matlab.

След прибавянето в модела Course Project Part 3.mdl на новия блок Signal To Workspace следва неговото първоначално наименование да се промени от yout на sigoutbi, както е показано на фиг.6. За целта чрез двукратно кликване върху блока Signal To Workspace се отваря прозорец с неговите параметри, както е показано на фиг.6 вляво, а след това се задава името sigoutbi на едномерния вектор, както е показано на фиг.6 вдясно. По този начин, приетата и декодирана аудио информация под формата на двоична последователност от данни за приетите аудио сигнали, се прехвърля от блока Signal To Workspace на симулационния модел Course Project Part 3.mdl в работното пространство (Workspace) на Matlab с наименование sigoutbi.

Фиг.6. Промяна на първоначалното наименование yout на блока Signal To Workspace като sigoutbi чрез двукратно кликване върху блока Signal To Workspace, отваряне на прозорец с неговите параметри (вляво), а след това задаване на новото име sigoutbi на (вдясно)

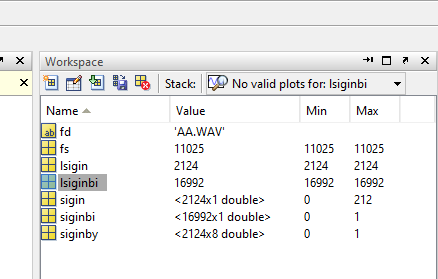
За да се предадe напълно от входния блок Signal From Workspace на симулационния модел Course Project Part 3.mdl през комуникационния канал AWGN цялата аудио информация съдържаща се в едномерния входен вектор siginbi и тя да се приеме изцяло като едномерния изходен вектор sigoutbi в блока Signal To Workspace, е необходимо в полето Simulation stop time (виж фиг. 7), в което първоначално стойността е безкрайност (inf), да се промени стойността в съответствие с дължината lsiginbi на едномерния входен вектор siginbi, като се зададе малко по-голяма от нея, за да се гарантира, че е предадена изцяло аудио информацията, съдържаща се в едномерния входен вектор siginbi.



Simulation stop time

Фиг.7. В полето Simulation stop time следва да се промени стойността в съответствие с дължината lsiginbi на едномерния входен вектор siginbi. За примера от фиг. 8 тази стойност е 17000

Дължината lsiginbi на едномерния входен вектор siginbi може да се види в полето в дясно горе на прозореца MatlabWorkspace (виж фиг. 8). В случая на фиг. 8 за разглеждания пример дължината lsiginbi на едномерния входен вектор siginbi е lsiginbi = 16992 и в полето Simulation stop time в съответствие с тази стойност е въведена като стойност 17000, която е малко по-голяма от действителната дължина lsiginbi на едномерния входен вектор siginbi (lsiginbi = 16992 виж на фиг. 8).



Фиг.8. Дължината lsiginbi на едномерния входен вектор siginbi може да се види в полето в дясно горе на прозореца MatlabWorkspace и в случая на фиг. 8 за разглеждания пример дължината lsiginbi на едномерния входен вектор siginbi е lsiginbi = 16992

След извършване на всички, описани по-горе необходими предварителни действия преди стартиране, моделът Course Project Part 3.mdl изглежда, както е показано на фиг.9 за пример на Hamming Encoder/Decoder и може да се стартира, като се изчака окончателното завършване на неговата работа. След приключване на работата на модела приетите стойности на едномерния изходен вектор sigoutbi от блока Signal To Workspace се прехвърлят в рабатното пространство MatlabWorkspace. Тези стойности са в двоичен код и следва да се преобразуват обратно в десетичен код. За целта следва да бъде съставена следната програма на Matlab (например с име sig\_out.m), която да се използва за преобразуване на получената изходна за модела аудио информация от двоичен код обратно

sigoutbi=sigoutbi(1:lsiginbi); % Make the length of output audio signal

%equal to length of input signal

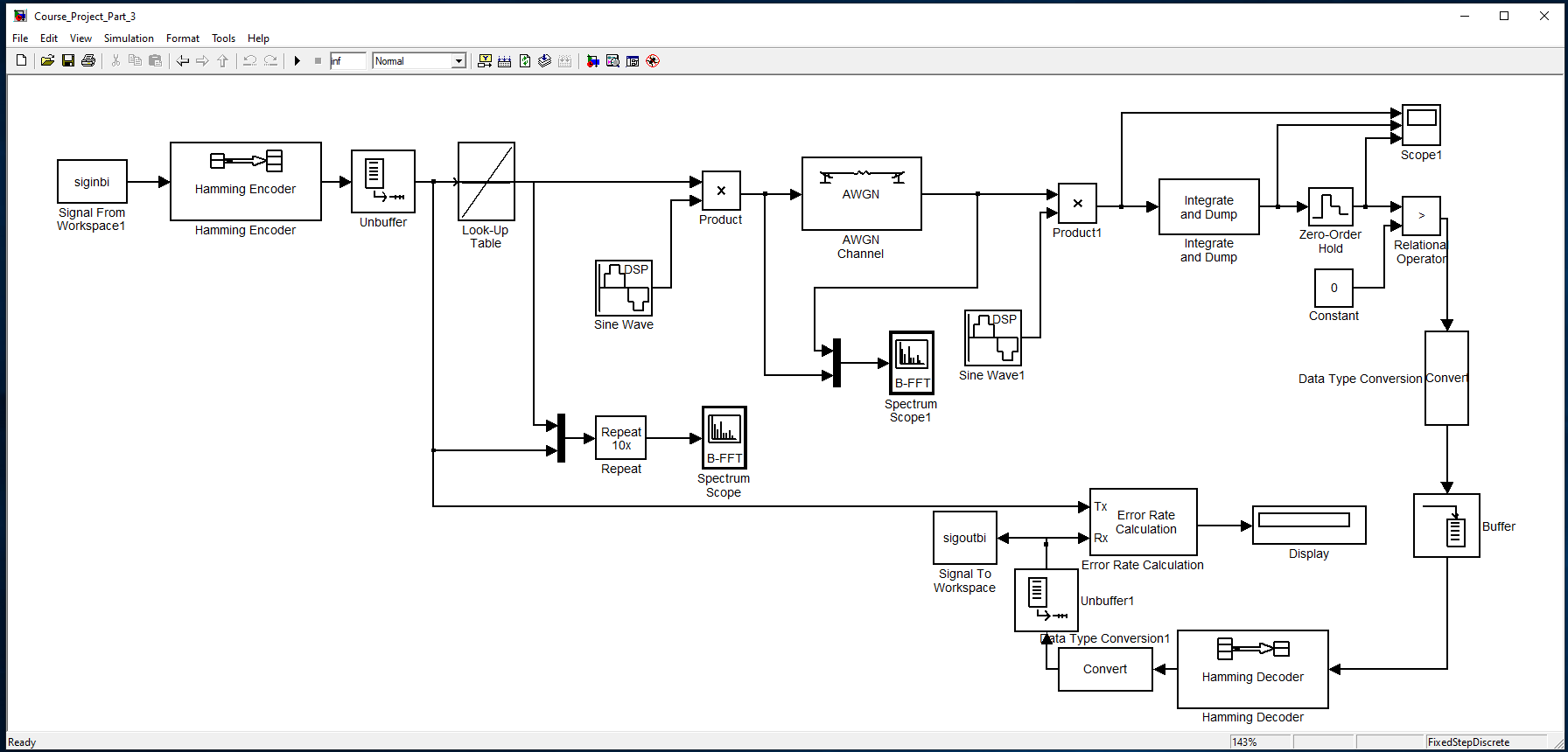
sigoutby=reshape(sigoutbi,lsiginbi/8,8); % Output signal in byte format

sigout=bi2de(sigoutby); % Output signal in decimal format

plot(sigout) % Show Output Audio signal

title('Output Audio Signal')

sound(sigout,fs)% Play Output Audio signal



Фиг.9. След извършване на всички, описани по-горе необходими предварителни действия преди стартиране, моделът Course Project Part 3.mdl изглежда, както е показано на фиг.9 за пример на Hamming Encoder/Decoder

След изпълнението на програмата (например с име sig\_out.m) се визуализира изходния аудио сигнал предаден по комуникационния канал за връзка AWGN и той може да се прослуша за сравнение с входния аудио сигнал.