The instruction for work with a demo program of superfast algorithm of multithreshold decoding

1. General information.

The program for work simulation of superfast multithreshold decoding algorithm (MTD) of convolutional codes is intended for analysis of efficiency and work speed of decoders of this type in гауссовских channels, and also if there is batch components of a noise in the received data flow. The description of principles of MTD operation is possible to find on our web-site www.mtdbest.iki.rssi.ru and in the reference book of V.V.Zolotarev and G.V.Ovechkin "Noiseproof coding. Methods and algorithms ", publishing house " Hot Line-Telecom", Moscow, Russia, 2004, 126 p.

The program creates an information flow, data coding with code rate R=3/4, transmission them through a channel with a large white Gaussian noise, and subsequent correction the distorted received sequence by the decoder. During depositing errors into transmitting message error packet forming of different length is possible. It allows to estimate a very high performance of the decoder work at composite nature of errors arising in communication channel.

Results of the program work are the speed data of coding system with MTD after all steps of data processing: at their formation, coding, depositing of a noise of a channel and then at decoding. Besides the program gives the information on number of errors in the data flow before the beginning of decoding and after each iteration of data correction.

The program works under the control of one of versions OS "WINDOWS" and at start at first it reads out the data from an input file **d34dv2d.c.**

2. Selection of parameters of simulation.

The selection of parameters of decoding is made in the file: **d34dv2d.c.** Its view in any editor is shown below.

```
code=3
 kst=155
  infind=2
   ndkdd=500000
    iterat=8
     AN=4.0
      npacketll=10
       lim88=88
```

The user can assign on the selection following channel parameters:

- initial state of the random-number generator for generating different sequences of errors. For this purpose parameter **kst** can be selected from range from 1 up to 999;

- length of encoded sequence in bits for each of 6 information branches, from 20000 up to 500000, parameter **ndkdd**;

- number of iterations of decoding from 3 up to 15, parameter **iterat** ;

- noise level, dB, from 2.0 up to 5.0, parameter **AN**;

- length of an error burst of a channel, from 10 up to 500 bits, for parameter **npacketll** . For cutoff of error burst generator this parameter might be selected negative.

Besides the selection of one of four versions of transmitted information sequence is possible . At **indinf=0** the zero information flow is encoded. At **indinf=1** - is encoded and transmits further to information flow consisting only of units. At **indinf=2** the information sequence with the 29 period. At last, if **indinf=3**, it transmits a random information flow. It is determined by defined above parameter **kst** for the random-number generator.

After that the data in this file should be saved, and then double click of "mouse" on a dv34v2.exe module simulation of system operation of coding(code) may be started.

3. Estimation of simulation results.

The completion of simulation process of system coding work ends usually after less than 30 seconds. For this purpose after appearance in a console message box "Press any key to continue" they must click once the "Input" key and to confirm then with the same key replacement of an older output file **dv34v2t.c** new, if you work in medium Visual C ++.

After that it is possible to analyze the texts of an output file **dv34v2t.c**, keeping all outcomes of the program work, with usage of any editor.

The initial part of an output file is submitted below. It contains all those items of information on simulation parameters, which ones were assigned by the user in an input file **d34dv2d.c** before start of the program.

```
========= start demo quick MTD ============
 convolution MTD soft   last corr 28.08.04 var.3!
  -------- code=3 -----------
 ================================
  kst=155
  infind =2
   ndkdd=500000
  iterat=8
```

```
 AN=4.000000
 npacketll=10
===  lim88=88  ==== -free control parameter
 ==================================
 start time is       0 s
==========================
```

The line

- free control parameter -

is a monitoring one and demonstrates, that if it has such a view, then a quantity of variables corresponds to their suspected number and the input data have been read correctly.

The line

- start time is 0 s -

demonstrates the moment, when systems of coding, noise generating and subsequent decoding starts work.

Further after the line "Program finish print" submission of those simulation results starts which ones are determined by adjustments of the user and consequently are most interesting.

The version of this main completing part of output data listing is also submitted below:

```
 ===========================
 Program finish print
  inf start   110 ms
  coding start   380 ms
  inf weight=1551726
   check weight=586095
  inftot=3000000
 coding end & noise  gen start = 710 ms
******** coder speed= 9090000  bits/s
   noise gen end   2690 ms
 inftot=3000000
  inferr gen=37990     isaerr gen=12319
  time    decoding start=2690 ms
  time   decoding end  = 5430 ms
 time decoding work= 2740  ms
 ******** decoder speed= 1094000  bits/s
packet length=10
 3000000 37992 12320 9863 3135 720 197 108 1 0 0
 ----=========================================--
```

Let's remind, that a full volume of simulation **inftot** because of features of demoprogram technology is always in 6 times more large, than parameter, assigned by the user, **ndkdd**. In this example **ndkdd=500000**, and

the full volume of simulations expressed as a number of information bits is equal to **inftot=3000000**.

As it follows from a view of shown output data on results of simulation, the formation of an information flow for transmission began through 440 milliseconds after start of the program, the process of coding began in 710 ms. Further there are some data about number of "ones" (binary vector weight) of informational input flow, and then the same weight of check array. Next we see a volume of input data **inftot** = 3 000 000 bits, time of coding end after 710 sec and coding was fulfilled with a speed 9,09 Mbit/s.

The following line demonstrates, that the generating of a noise was terminated through 2690 ms after the program start and then the endorsement from other module repeats, that the volume of the decoded data is 3 Mbits. Further there is an information about number of the generated errors in an input informational stream of the decoder - **inferr gen=37990**, and also about number of errors in check bits in a code flow - **isaerr gen=12319**.

Then there is an information on the most important stage - beginning of a decoding procedure - **time decoding start=2690 ms** and its end - **time decoding end = 5430 ms**, and also about its full time work - **time decoding work = 2740 ms.**

And at last, most important result - about decoding speed:

********** decoder speed = 1094000 bits/s.**

These numerical data are shown for the computer with clock rate less than 1 GHz.

Further a length of an error burst is indicated, which one was entered into an input stream - **packet length=10**.

In the last string 11 numbers - detailed totals of MTD simulation work are indicated. A full volume of transmitted information characters - 3 Mbit, is repeated again, then - number of errors in informational bits transmitted through a channel - 37993, number of errors in check characters - 12321 and further - number of t he errors which have stayed in the decoder after 8 iterations of error correction at usage of MTD algorithm at first is again injected: from 9864 after first up to 0 (!) - after seventh and 8-th iterations. The total number of errors transmitted through a channel is given with the count of errors generated in a packet.

At assigning other values of those parameters, which one can change by the user during simulation of MTD work, the analysis of outcomes of total listing is carried out under the same scheme.

*

* *

* * * *

*

It is possible to test also, that at replacement in an array of input data with length of an error burst from 10 to 90 bits repeated simulations of process of error correction also results in absence even of tracks of this packet on an MTD exit, т. е. after its last iteration there are no any errors at all. It also determines approximately very relevant capabilities of the given decoding circuit on correcting long error bursts also. Actual length of such a packet on the MTD input can more, than in 10 times exceed length of a maximum packet which is permissible at usage of Viterbi algorithm, for which one it may be only 7-10 bits.

4. Conclusion.

The designed demoprogram **demo Quick MTD** allows to analyze the characteristics of throughput and correcting capability for MTD algorithms both in Gaussian channels, and in channels with the considerable component of a packet noise.

----------==========----------

-