Modelling of Convolutional Structures Focusing on Error Probability Calculation Via Matlab for Safety-Related Applications

Maria FRANEKOVA, Peter ZDANSKY, Emilia BUBENIKOVA, Peter HOLECKO

Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Zilina, Univerzitna 8215/1, 01026 Zilina, Slovak Republic

maria.franekova@fel.uniza.sk, peter.zdansky@fel.uniza.sk, emilia.bubenikova@fel.uniza.sk, peter.holecko@fel.uniza.sk

DOI: 10.15598/aeee.v17i2.2983

Abstract. *Safety-related communication system SR-ComS uses a safety code for achieving the integrity of messages transferred between safety-related devices. In most cases, it is a block systematic code that uses syndromic decoding technique. The authors discuss the possibilities of using the convolutional structures for safety-related applications requiring creation of safety case based on quantitative assessment in relation to the transmission integrity level. This assessment is based on code analysis focused mainly on the error probability calculation. The results published in the practical part are obtained using a model created in Matlab environment, in which the chosen convolutional structures were tested. These tests were focused on the transmission error probability analysis provided by a binary symmetric channel, and also on optimization of other parameters affecting the residual error in the decoding algorithm. The results are generalized into concept of methodology for the convolutional codes safety assessment in compliance with standards for safety-related applications.*

Keywords

Binary symmetric channel, convolutional structures, error probability, Matlab, modelling, safety code, safety-related communications systems, signal graph.

1. Introduction

Convolutional codes are a part of a large set of channel self-repair codes, which are used in many practical applications requiring immediate (real-time) repair of broken data [1]. Correction properties of convolutional codes for different types of encoders should be verified using models in appropriately chosen software. The most commonly used way to describe the coding and decoding of the convolutional code structures is the trellis diagram, which combines elements of the tree and state diagram [2]. The basis for determining the convolutional code error probability is the signal graph.

The structure of the linear convolutional code is represented by (n, k, L), where n is the number of bits of coded word, k is the number of information bits at the input and L is the constraint length representing the number of bits in encoder memory. Convolutional code is described also by its transfer function tf. Decoding algorithm is not described by the circuit connection, but using the Trellis diagram. This algorithm uses the principle of Maximum-Likelihood Decoding (MLD). The most common method using this principle is the Viterbi method [3]. There are also decoding techniques using tree expression.

Convolutional coding and decoding techniques can be implemented by software in conjunction with the technical implementation (hardware). It is not possible to clearly determine which principle is the most appropriate. The choice depends on the requirements on code from the perspective of output device used and the required transmission speed via a channel with a predefined error structure.

For the simulation of data transmission secured by convolutional codes, the Matlab software along with Simulink and Communications System Toolbox libraries is very useful [4]. This environment allows to create a communication system model from individual function blocks and to adjust their parameters according to the needs of the specific problem. Using this model, it is possible to create a time simulation of transmission with the option to analyse the system, which is used for data transmission through a communication channel and optimization of the convolutional codes parameters such as Decoding Window (DW) or Safety Interval (SI) affecting the error probability of one bit during the transmission. The library contains so called M-function for most function blocks and this M-function is possible to combine with the system model created in Simulink environment. There is not always a corresponding M-function or vice versa, some functions do not have an equivalent in the simulation library.

Matlab provides, except for Viterbi decoding algorithm, two decoders for the convolutional codes decoding. One of them is the APP decoder working on the „posteriori“ probability principle. The second option is an effective way of decoding using the Turbo decoder, which decodes the input signal using the parallel concatenated decoding scheme, which is also working with „posteriori“ probability [5].

The effort of the authors is the suggestion of using the chosen convolutional structures for safety-related applications. Therefore it is necessary to provide a quantitative safety assessment of the transmission based on the detailed codes analysis with focus on choosing the optimal connection regarding the Hamming distance d_{free}, the safety interval between errors and the error probability calculation in the required bit error rate range by using the created model.

The results obtained can be used for generalization of knowledge and for creation of the methodology for verification of the correction properties of convolutional codes in compliance with standards for safety-related applications.

2. Safety Code Requirements for Safety-Related Applications

If the Safety-Related Electronic System (SRES) includes the information transmission between two and more safety-related devices, then the communication system is the integral part of the SRComS and it has to be proven that transmission on the output devices level is safe in compliance with standards, e.g. [6] and [7]. The requirement for the integrity of security against accidental failure is reported in European standards and as a Tolerable Hazard Rate (THR), which is taken as a parameter in the design of binding parameter when constructing, for example, railway security equipments, nuclear power plants or chemical plants. THR values are assigned to the Security Integrity Level (SIL) according to the international standard IEC 60 518. More can be found in [6], [7], [8], and [10].

It is known, that in order to increase the transmission quality, detection (or correction) techniques of transmission codes are used in protocols of conventional transmission systems for detection (or correction) of simple errors or clusters of errors. Although these transmission codes may be very effective, they may fail due to HW faults, interference from the external environment or the occurrence of systematic errors. These transmission codes in SRComS are considered untrustworthy. Therefore, it is required to use the additional safety code for detection of the Safety-Related Message (SRM) damage. Its position in the model of message transmission (Fig. 1) is in layer providing functions of transmission protection [8].

In case of using a safety code, its ability to detect all expected types of errors with defined values of the detected errors probability has to be proven (proving the fulfilment of safety requirement related to the defined maximum value of the undetected errors probability). Although the standard [6] recommends using detection codes only, there are known applications from the set of block codes, such as [10] and [11], which use coding structures of the correction codes with good properties. These codes accelerate the transmission and have advantageous properties, while its HW or SW implementation is based on its detection property only (e.g. Hamming or Reed-Solomon codes).

The requirements for choosing the safety code parameters depend on the required safety properties of the SRComS (especially on tolerable intensity or probability of undetected broken messages). If the SRComS is used for transmission of Safety-Related and also Non-Related Messages (SRM and SNRM), then these messages have to differ by their structure.

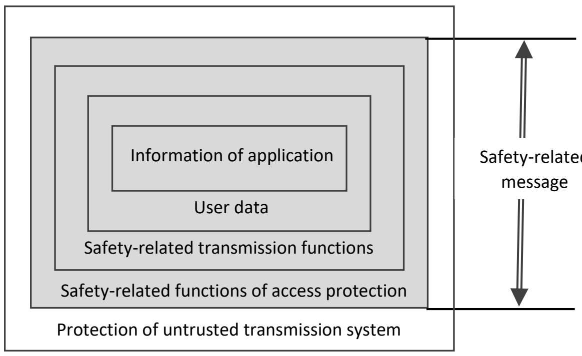

Fig. 1: The location of safety code in SRM model.

The fulfilment of safety requirements, which are defined for SRComS, requires the safety code to have the ability to detect and appropriately react to typical faults of the untrustworthy transmission system. At least these failures or errors (causing the faults) have to be taken into account:

- interrupted transmission line,
- all message bits with logical 0,
- all message bits with logical 1,
- message negation,
- synchronization loss (in case of serial transmission),
- random errors in message,
- clusters of errors in message,
- systematic errors, e.g. repeated types of errors,
- combination of above stated failures, resp. errors.

The important request, which affects the choice of the safety code parameters, is the undetected (resp. uncorrected) error probability, so-called residual error rate of the code, which has to be under defined (required) level.

The trend in the safety-related applications, which presently use wireless technologies (open transmission system), is from the safety code point of view the using of codes with fast coding and decoding algorithm. In many cases, these are real-time applications with strong detection, resp. correction properties (especially in the case of applications with increased industrial interference). The idea of using well-proven correction codes from the non-safety application areas is also current due to the practical problems encountered with the block codes used so far (it is a group of cyclic codes working on the Cyclic Redundancy Check (CRC) principle). The main problems according to [10] are as follows:

- Code length - it is usually required to have the ability to secure messages of different length by one code. The most of the theoretical knowledge about the codes is bound to the certain recommended construct code lengths, which is mostly fulfilled in practice (abbreviated or expanded codes are used). It is important to note that the code of each length has specific detection properties and from that resulting safety properties.
- Minimal Hamming distance ($d_{\min}$) is bounded by requirement that the code should detect errors to a certain multiplicity. Most codes detect simple (independent errors) and errors with i-th multiplicity (dependent errors - clusters). In practical applications, the $d_{\min}$ is not always known and if it is known, so it is valid for code of certain construct length. If the code length is exceeded, $d_{\min}$ may drop sharply.
- Detection of chosen error patterns – some types of errors can occur more often and therefore the code is required to detect them (e.g. repeated clusters of certain length, all bits „1“ or „0“ or message negation). Mostly, however, the probability of error patterns occurrence is unknown.

For these reasons, using the original convolutional structures or modified convolutional codes for real-time applications (after detailed analysis) is very interesting. However, it should be noted that the separation of the detection process from the correction process in case of MLD decoding is not as simple as in the case of block systematic codes using the syndromic decoding technique, although it is possible to reach the software solutions.

3. MLD Decoding-Based Convolutional Code Design in Safety-Related Applications

The basic objective is to analyse the correction properties of chosen convolutional structures via created model and modify the algorithm of MLD decoding. It is assumed that the Viterbi algorithm with hard decision principle is used. The aim is to remove error correction from the decoding process and keep the code usable in safety-related applications in compliance with standard recommendations [6].

This approach is explained using a simple example of linear convolutional code with transfer function $tf = [5\ 7]$. It is an optimal connection regarding the maximal value of the free Hamming distance $d_{free} = 5$, with correction properties of double errors cluster and with safety interval $SI = 10$ bits and if $DW = 5$ tact. The Trellis diagram for decoding this linear convolutional code with the coding rate $r = 1/2$ and redundancy $R = 50$ % is shown in Fig. 2.

Let the data sequence 00 11 11 00 10 on the decoder input be influenced by the error sequence 00 00 10 00 00, which is artificially applied on the coded sequence (single error). According to the Viterbi algorithm, a path with better metrics is searched in every tact using the Trellis diagram. The path metrics with the lowest value determine the most probable path, corresponding to the input sequence (blue path in the Fig. 3). Procedure of original MLD decoding according the Viterbi algorithm for connection with

$tf = [5\ 7]$ are tabled in Tab. 1, Tab. 2, Tab. 3 and Tab. 4.

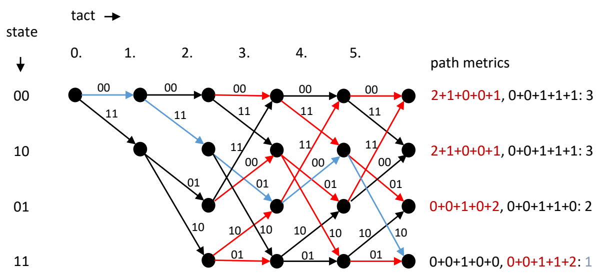

Fig. 2: Example of original procedure of MP decoding according to the Viterbi algorithm ($DW = 5$ tacts).

Tab. 1: Paths metrics.

state / tact	0.	1.	2.	3.	4.	5.
00	0	0	2	3	3	3
10	-	2	0	2	1	3
01	-	-	3	1	2	2
11	-	-	3	1	2	1

Tab. 2: Non-excluded previous states.

state / tact	0.	1.	2.	3.	4.	5.
00	00	00	00	01	00	01
10	-	00	00	00	01	01
01	-	-	10	10	11	11
11	-	-	10	10	11	10

Tab. 3: Map of transitions between states.

actual state / next tact	00	10	01	11
00	0	1	-	-
10	-	-	0	1
01	0	1	-	-
11	-	-	0	1

Tab. 4: Decoded sequence.

tact	0.	1.	2.	3.	4.	5.
state	00	00	10	01	10	11

From the procedure stated above is obvious that the error was removed during the decoding. The decoding window parameter in original procedure is very important to achieve the optimal decoding. The higher the DW, the higher the probability of correct decoding and the higher the safety interval between errors. However, increasing the DW value slows the decoding process down [12].

It has been empirically proven that in order to achieve the optimal code correction properties, this value is determined as follows:

$$DW = 5 \cdot L, \tag{1}$$

where DW is size of decoding window or, in other words, the decoding depth, and L is the constraint length. Correction properties of encoders are often increased by increasing the constraint length L.

The proposed procedure for modification of the original decoding algorithm is based on the evaluation of metrics of each DW, while the path with zero metrics is determined. In case of different (higher) metrics during the SRM transmission an error occurred (an error was detected), the message has to be transmitted again. It is therefore necessary to use the Automatic Request Question (ARQ) mode for message repetition, which is standardly used in SRComS.

In the observed example of decoding the sequence (Fig. 2), a path with zero metrics would not be found after DW processing (the best path in the example is with $d = 1$) and the receiver would issue a command for message repetition. It would be advantageous if both the original and the modified algorithm were processed and the results would be compared on the receiving side, thereby increasing the eliability of the transfer. The message would be evaluated as valid only if the two decoding algorithms match.

This proposed approach appears to be effective for shorter messages (the so-called short message format) where the telegram can be divided into fewer decoding windows to ensure system availability. In most cases, in SRComS (compared to conventional systems) it is a typical message format being used. If the message length is, for example, 2B, it is the short message format for safety profile used in the safety related applications that communicate via Common Industrial Protocol (CIP), so called CIPsafety [7]. The message length 16-bits and the processing corresponds to the two decoding windows after adding the padding. Regarding the transmission speed of decoding (tact) and simple algorithm (shift and mod2 function), it is a very fast process.

The modified procedure of decoding appears to be effective in safety related applications for the message length up to 4B. For longer formats, it is necessary to simulate the time dependences of the decoding for generated error patterns or to use the original algorithm after a detailed safety analysis, especially from the perspective of calculating the uncorrected error probability within the defined bit error range, to meet the tolerable rate of dangerous failures caused by electromagnetic interference.

4. Model Implementation

The correction properties of chosen convolutional structures were tested using a model. The model includes a data source (Bernoulli Binary Generator)

Tab. 5: Results of SI for chosen optimal connections of encoders with the coding rate 1/2 and the constraint length $L = 3$, 5, and 8.

	Type of error				
	a)	b)	c)	d)	e)
$SI\,(L=3)$	7	15	-	-	-
$SI\,(L=5)$	7	18	31	43	-
$SI\,(L=8)$	9	15	26	36	44

a) Periodically repetitive simple error (100..00100..001..).
b) Periodically repeating burst length of 2 (1100..001100..0011..).
c) Periodically repeating burst of 3 (11100..0011100..00111..).
d) Periodically repeating burst of 4 (111100..00111100..001111..).
e) Periodically repeating burst of 5 (1111100..001111100..).

generating bits with a defined rate, convolutional encoder/decoder (Convolutional Encoder, Viterbi Decoder) and a model of binary symmetric channel (Binary Symmetric Channel) [13]. Two functional blocks of the BSC are used. In the lower branch, the data goes directly to the function block BSC2, without using the safety code. In the upper branch of the model, data is initially secured by safety code and then goes to the block BSC1. In the dialog window of the BSC1, the error rate is set identically to BSC2. The outputs of both branches are compared with input data in the Error Rate Calculation ERC1 and ERC2 evaluators.

The tests were performed in faulty and fault-free operation in compliance with the requirements for optimal connections of several linear convolutional codes with $tf = [5\ 7]$, [23 35] and [247 371], while their correction properties considering the increasing constraint length L were examined. In the Tab. 6, these code parameters are stated: d_{free} and the number of corrected errors p_{cor}. The choice of encoder connections can be changed in the encoder/decoder dialog window by changing the transfer function. Different error patters (from single error up to error clusters of required length) were simulated in compliance with the requirements. The script for generating the clusters of length 2 up to 5 is shown in Fig. 3. The cluster length is adjustable in the script.

```
clc;                                % vyčistenie Command Window
len=1000;                           % nastavenie dĺžky chybovej postupnosti
deklen=2*len;                       % dĺžka zakódovanej postupnosti
vstup=randi([0,1],len,1);           % generovanie vstupnej postupnosti
trellis=poly2trellis(3,[5 7]);      % trellis štruktúra => tf[5,7]
CH=comm.ErrorRate;                  % konštruktor chybovosti
chyby=zeros(deklen,1);              % chybový vektor
for i = 1:17:deklen                 % definovanie konkrétneho typu chyby
chyby(i,1)=1;
chyby(i+1,1)=1;
end
kod=convenc(vstup,trellis);                                 % kódovanie
chybnykod=xor(kod, chyby);                                  % pripočítanie XOR chybového vektora
vystup=vitdec(chybnykod,trellis,15,'trunc','hard'); % dekódovanie => Viterbiho dekóder
vystup=cast(vystup,'double');                               % zmena typu z logical na double
errorStats=step(CH,vstup,vystup);                           % výpočet chybovosti
fprintf('Viterbiho dekoder:\n');                            % výpis
fprintf('Error rate = %f\nNumber of errors = %d\n',errorStats(1),errorStats(2));
fprintf('\n');
```

Fig. 3: Script for generating the error structures during transmission [13].

In the simulation realized in Matlab version 8.5 (also known as R2015a), we worked with the following types of errors a) - e) described in Tab. 5 and we used their periodic repetition at a chosen length of error sequence of 2000 bits. The safety interval determines the number of flawless symbols within the error sequence. The results of the safety interval are stated in Tab. 5.

The script allows to test the convolutional structures with inappropriate transfer function, so called catastrophic connections, which cannot be used in the form of safety code due to a dysfunctional MLD. For the simulated time transmissions of secured data, the safety interval between errors was examined.

Tab. 6: Correction properties of chosen convolutional codes considering the increased constraint length L.

L	tf	d_{free}	p_{ch}	DW
3	[5 7]	5	2	15
5	[23 35]	7	3	25
8	[247 371]	10	4	40

5. Results

The basis of the quantitative safety evaluation of the safety code is a detailed analysis of the error probability depending on the bit error rate and the subsequent dangerous failure rate calculation. In compliance with [6], this function has to be monotonically increasing for all values of the bit error rate of the communication channel. For the error rate calculation, a script was created with the possibility of more detailed analysis of graphic dependencies. The results were compared with results obtained using the created model on the basis of time simulation of transmission. The uncorrected error probability for convolutional codes is possible to be determined using the signal graph. The procedure is shown for a linear convolutional code (2, 1, 2) with $tf = [5\ 7]$. This procedure can be generalized for any convolutional encoders defined by the state diagram and consequently a generally valid methodology can be created.

The procedure consists of these steps:

- Creation of the signal graph from the state diagram of convolutional code.
- Allocation of the D, L, N parameters to the signal graph connectors.

- Creation of the transfer function $T(D, L, N)$, which uniquely represents the code states and all paths for the transmission of binary sequence from the input to the output (it is expressed by an infinte geometric series).
- Creation of the geometric series sum of the transfer function.
- Exclusion of parameters of the transfer function, which have a unit value (simplification of the transfer function).
- Calculation of partial derivation of the transfer function according to the parameter N (exponent of the parameter affects the code correction properties).
- Application of the bit error rate p_b for parameter D provided the BSC.
- Determination of upper border of probability of error event occurrence while using the MLD according to Viterbi.
- Analysis of the uncorrected error probability function depending on the p_b (function has to be monotonically increasing for all observed values of the p_b).
- Determination, i.e. estimation of the fault messages number per hour (pessimistic estimation is that all received messages are faulty).
- Calculation of the tolerable dangerous failure rate [h^{-1}].

The signal graph for the example stated above is shown in Fig. 4. The connectors of the signal graph are evaluated by the parameters D, L, and N, whose exponents are determined as follows: the power of parameter D determines the Hamming distance of the output code word from the zero code word, power of parameter L determines the number of branches in the path, and power of parameter N determines the weight of the input sequence (number of ones in the input sequence). Using stated parameters, it is possible to write the transfer function of the $T(D, L, N)$ code, which is determined by the infinite geometric series, Eq. (2).

$$\begin{aligned} T(D,L,N) &= D^5L^3N + D^6L^4(1+L)N^2 + \\ &+ D^7L^5(1+L)^2N^3 + \ldots \\ &= D^{5+k}L^{3+k}(1+L)^kN^{1+k}. \end{aligned} \tag{2}$$

The sum of this geometric series leads to Eq. (3).

$$T(D,L,N) = \frac{D^5L^3N}{1 - DL(1+L)N}. \tag{3}$$

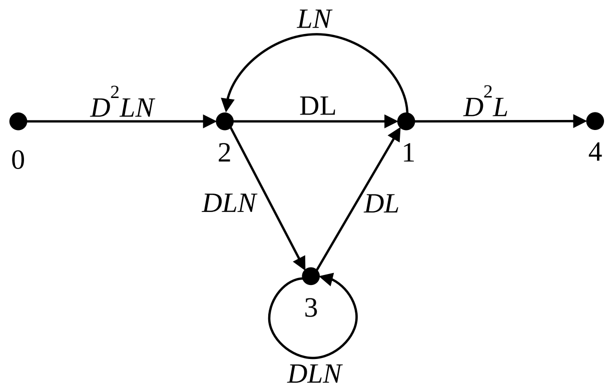

Fig. 4: The signal graph of the convolutional code (2,1,2) with tf = [5,7].

Function $T(D, L, N)$ determines uniquely the convolutional encoder (the state diagram with finite number of states), i. e. through what paths is possible to pass from input node (0) to output node (4) in the signal graph in Fig. 5 when transmissioning the input sequence.

Note: The transfer function $T(D, L, N)$ for the observed example of the encoder with tf = [5 7] has parameters $L = 1$ and $N = 1$ and is edited into Eq. (4). From the signal graph is possible to determine the free Hamming distance of the code d_{free}, that is equal to the smallest exponent of parameter D.

$$T(D) = D^5 + 2D^6 + 4D^7 + \cdots + 2^kD^{5+l}. \tag{4}$$

Let us denote the number of code paths when using the MLD, that are acquiring the weight j, as n_j. Then it is possible to modify the Eq. (4) into Eq. (5) and the transfer function $T(D)$ from Eq. (5) can be expressed by the sum Eq. (6) [5].

$$T(D) = \sum_{j=0}^{\infty} n_j D^j, \tag{5}$$

$$T(D) = \sum_{j=5}^{\infty} 2^{j-5} D^j. \tag{6}$$

Let us denote the error event occurrence probability when using the MLD according to the Viterbi p_E. Let the error occurrence probability in the MLD process be p_j. Then the error event occurrence probability for the number of paths n_j with weight j is lower than the probability sum of all possible paths, that will merge in the signal graph with node (0).

$$p_E < \sum_{j=0}^{\infty} n_j p_j. \tag{7}$$

The uncorrected error occurrence probability in one bit (p_{unc}) has to be lower, than the sum of all paths, that are given by the product of the total weight w_j and the error probability during decoding decision p_j (see Eq. (8)).

$$p_{unc} < \frac{1}{k}\sum_{j=0}^{\infty} w_j p_j. \tag{8}$$

For convolutional codes with coding rate $r = k/n$, k symbols from each branch are decoded. In the observed example $k = 1$.

During the message transmission through the binary symmetric channel, the error probability when decoding decision p_j is lower than Eq. (9).

$$p_j < [p_b(1-p_b)]^{\frac{1}{2}}. \tag{9}$$

where p_b is the error probability of one element (bit) in the BSC channel.

According to the [7] the equation for upper border of the uncorrected error probability of one element (bit) is:

$$p_{unc} < \frac{1}{k}\frac{\partial T(D,N)}{\partial N}. \tag{10}$$

If in the Eq. (3) $L = 1$ and the Eq. (10) are applied for $k = 1$, we can calculate the error probability for one tact (1 path) in the DW.

$$\begin{aligned} &\frac{\partial T(D,N)}{\partial N} = \\ &= \frac{D^5}{1-2DN} - \frac{D^5 N}{(1-2DN)^2}\cdot(-2D) = \\ &= \frac{D^5(1-2DN)+2D^6N}{(1-2DN)^2} = \\ &= \frac{D^5-2D^6N+2D^6N}{(1-2DN)^2} = \frac{D^5}{(1-2DN)^2}. \end{aligned} \tag{11}$$

After modification of the equation (Eq. (11)) for $N = 1$ and influence of the BSC, i. e. $D = 2[p_b(1-p_b)]^{1/2}$, we get Eq. (12).

$$p_{unc}\frac{\partial T(D,N)}{\partial N} = \frac{2\left[p_b(1-p_b)^{\frac{1}{2}}\right]^5}{\left[1-4\left[p_b(1-p_b)^{\frac{1}{2}}\right]^2\right]}. \tag{12}$$

The obtained results of the uncorrected error probability for the analysed connection of linear convolutional code with transfer function $tf = [5\ 7]$ are shown in Fig. 5.

The error rate of the analysed connection with the redundancy of 50 % $tf = [5\ 7]$ is comparable with the Hamming block code (7, 4) with redundancy of 42 %. A graphical dependence of the uncorrected error probability for both encoders is shown in Fig. 6.

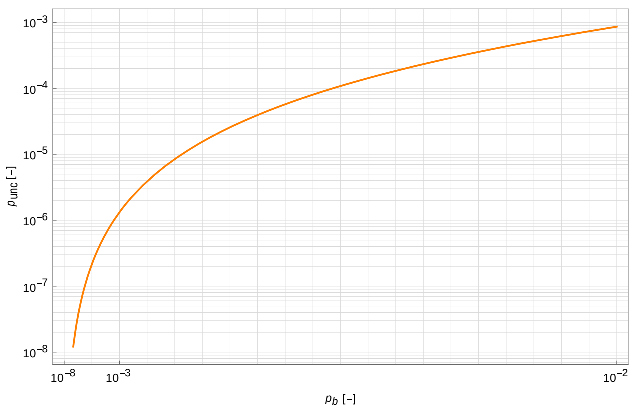

Fig. 5: Graphical dependence of the uncorrected error probability on the p_b. The signal graph of convolutional code (2,1,2) with $tf = [5\ 7]$.

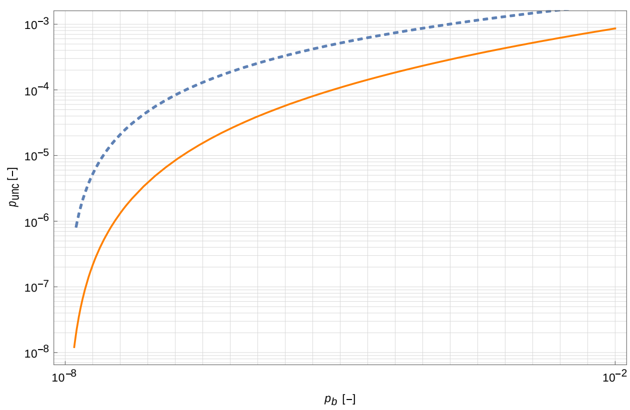

Fig. 6: Comparison of graphical dependence of the uncorrected error probability on the p_b for convolutional code (full line) with $tf = [5\ 7]$ and Hamming code (7,4) (dotted line).

The uncorrected error probability of the Hamming code (7, 4) is calculated according to the (Eq. (13) [4].

$$\begin{aligned} p_{unc} = 21p_b^2(1-p_b)^5 + 35p_b^3(1-p_b)^4 + \\ +21p_b^5(1-p_b)^2 + 7p_b^6(1-p_b)^1 + 1p_b^7 \end{aligned}. \tag{13}$$

As already emphasized, the requirement for the use of a safety code in safety related applications is the monotonous growth of the error probability function over the entire bit error range $p_b \in (0;\ 2^{-1})$. In our observed example of convolutional code with $tf = [5\ 7]$ in Fig. 7, the function is monotonically increasing. However, the error rate graph for the worse quality of the communication channel shows that there is a leap change in the uncorrected error probability. This could cause a critical error in applications with increased safety level when receiving the secured message. A graphical dependence of the uncorrected error probability with $tf = [5\ 7]$ for the worse quality of communication channel is shown in Fig. 7.

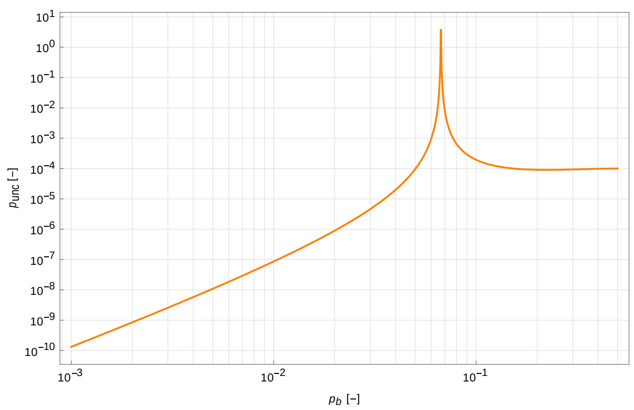

Fig. 7: Example of graphical dependence of the uncorrected error probability with $tf = [5\ 7]$ for communication channel with the worse quality.

6. Conclusion

For safety-related communication, it is necessary to focus on the use of detection channel codes (which, in the case of error detection, require an automatic retransmission mode) or the use of correction code techniques.

In the case of binary transmission, the most frequently used block systematic codes (CRC, Hamming codes) have some disadvantages. Therefore, the authors have attempted to find a solution in a form of using the convolutional codes. These codes belong to group of correction codes and are used in a number of communication applications. The convolutional codes achieve reliable data transmission mainly due to a good burst-error correction ability.

When using the safety code in safety-related applications, it is necessary to quantitatively demonstrate the ability to detect all expected error types with a defined probability value of detected errors or to demonstrate the compliance with the safety requirement related to the maximum probability of undetected errors. The methods of assessing communication security are based on the modelling principle.

In this paper, we verified the proposed methodology for assessing the safety message transmission for the selected convolution structures via models in Simulink and using the created m-files in Matlab. To simulate error transmission over a communication channel, we used a binary symmetric channel model. The results obtained are generalized in the proposal for a methodology for assessing the safety of convolutional codes in accordance with standards for safety-critical applications.

Acknowledgment

This work has been supported by the Educational Grant Agency of the Slovak Republic (KEGA) Number: 016ZU-4/2018 Modernization of teaching methods of management of industrial processes based on the concept of Industry 4.0.

References

[1] JOHANESSON, R. and K. S. ZIGANGIROV. *Fundamentals of Convolutional Coding*. 1st ed. Hoboken: John Wiley, 2015. ISBN 978-1-119-09867-6.

[2] MUZIKAROVA, M. and M. FRANEKOVA. *Teoria informacie a signalov*. 1st ed. Zilina: EDIS-vydavatelstvo, 2009. ISBN 978-80-554-0075-4.

[3] ABUBEKER, K. M., A. R. BUSHARA and S. BACKER. Maximum likelihood DE coding of convolutional codes using viterbi algorithm with improved error correction capability. In: *2013 IEEE Conference on Information & Communication Technologies*. Thuckalay: IEEE, 2013, pp. 135–138. ISBN 978-1-4673-5758-6. DOI: 10.1109/CICT.2013.6558082.

[4] FRANEKOVA, M., R. PIRNIK and L. PEKAR. *Modelling of the data transmission via Matlab, Simulink a Communications Sytem Toolbox*. 1st ed. Zilina: University of Zilina, 2014. ISBN 80-227-0513-6.

[5] FARKAS, P. *Kodovanie a modulacie*. 1st ed. Bratislava: Slovenska technicka univerzita v Bratislave, 1993. ISBN 978-80-554-0896-5.

[6] STN EN 50159. *Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems*. Brussels: CENELEC, 2010.

[7] IEC 61508, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems*. Geneva: International Electrotehnical Commision, 2009.

[8] FRANEKOVA, M., F. KALLAY, P. PENIAK and P. VESTENICKY. *Communication safety of industrial networks*. 1st ed. Zilina: University of Zilina, 2007. ISBN 978-80-8070-715-6.

[9] KLAPKA, S. and M. HARLENDEROVA. Quantitative Assessment of Safety Codes. In: *Proceeding of international conference FORM/FORMAT*. Budapest: EPL, 2008, pp. 249–255.

[10] RASTOCNY, K., M. FRANEKOVA, I. ZOLOTOVA and K. RASTOCNY. Quantitative assessment of safety integrity level of message transmission between safety-related equipment. *Computing and Informatics*. 2014, vol. 33, no. 2, pp. 1001–1026. ISSN 1335-9150.

[11] FRANEKOVA, M. and K. RASTOCNY. Problems of safety evaluation of safety codes in practical applications. In: *International Conference on Transport Systems Telematics*. Berlin: Springer, 2010, pp. 232–242. ISBN 978-3-642-16472-9. DOI: 10.1007/978-3-642-16472-9_26.

[12] FRANEKOVA, M. *Convolutional coding structures*. Zilina, 1995. Dissertation Thesis. Vysoka skola dopravy a spojov v Ziline.

[13] ZDANSKY, P. *Convolution codes in SW Matlab*. Zilina, 2015. Bachelor Thesis. University of Zilina.

[14] IEC 61784-3, *Industrial communication networks – profiles – Part 3: Functional safety Filedbuses – General rules and profile definitions*. Geneva: International Electrotechnical Commission, 2010.

[15] FRANEKOVA, M. and K. RASTOCNY. *Kryptografia v bezpecnostne relevantnych systemoch*. 1st ed. Zilina: EDIS-vydavatelstvo, 2017. ISBN 978-80-554-1310-5.

About Authors

Maria FRANEKOVA was born in Brezno (Slovak Republic) in 1961. She received her Prof. in 2011 in the field of Automation with orientation to "Safety-related Control and Communication Systems". Her research interests include assessment of safety communications on the base of coding and cryptography tools within safety-related applications.

Peter ZDANSKY was born in Nova Bana (Slovak Republic) in 1993. He received his Master (Ing.) degree in 2015 from the University of Zilina, where he is currently pursuing the Ph.D. degree. His research interests include the safety and availability assessment of safety-related electronic systems, especially in the field of railway applications.

Emilia BUBENIKOVA was born in Martin (Slovak Republic) in 1969. She received her Ph.D. from the University of Zilina in 2014 in the field of Automation focused on „Detection of lanes in control applications in road transport“. Her research interests include methods of coding theory and image digital processing and control systems.

Peter HOLECKO was born in Zilina (Slovak Republic) in 1981. He received his Ph.D. in University of Zilina in 2012 in the field of Automation and specialisation on sensor networks. Currently he is an assistant professor and his research interests include security of information systems and sensor networks.