surely detects up to 4 incorrectly transmitted bits. For input blocks with length 7 or 8 symbols, the code surely detects up to 2 incorrectly transmitted bits. When the length of the input block is greater than 8 symbols, the code detects for sure 1 incorrectly transmitted bit.

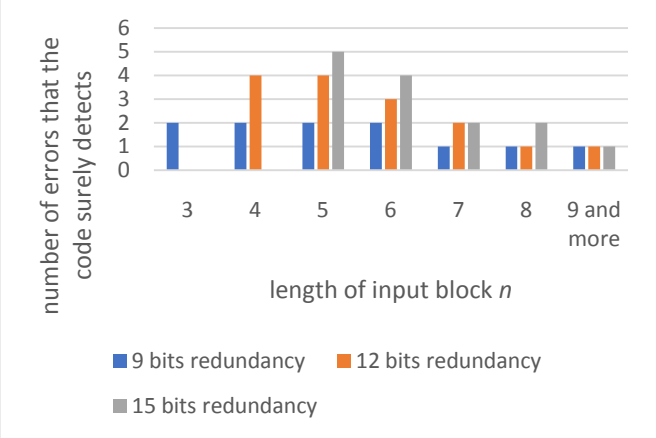

Fig. 4 *Number of incorrectly transmitted bits that the code surely detects when the length of the input blocks is n symbols from the alphabet ∑ and the redundancy is 9, 12 and 15 bits.*

In Fig. 4 and Fig. 5 are presented the numbers of incorrectly transmitted bits that the code surely detects when the redundancy is 9, 12 and 15 bits. On *x*-axis in Fig. 4 is given the length of the input block, on *y*-axis is given the number of errors that the code surely detects, while the color of each pillar represents the length of the redundancy.

From Fig. 4 we can see that when the length of the input block is fixed, if longer redundancy is added to the input blocks, then the number of incorrectly transmitted bits that the code detected for sure is greater or at least equal to the number of surely detected incorrectly transmitted bits when a shorter redundancy is added. This is expected result since longer redundancy means that each information symbol is controlled by more redundant symbols.

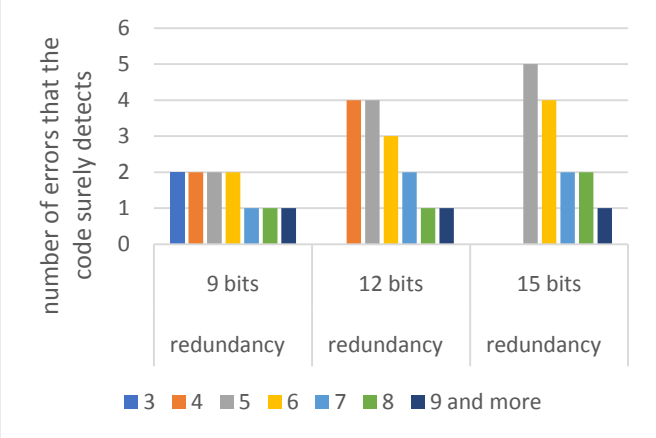

Fig. 5 *Number of incorrectly transmitted bits that the code surely detects when the length of the input blocks is n symbols from the alphabet ∑ and the redundancy is 9, 12 and 15 bits.*

On Fig. 5 on *x*-axis is given the length of the redundancy, while the length of the input blocks *n* is represented with the color of the pillars. As we can see from Fig. 5, regardless of the length of the redundancy, when the length of the input block increases and the length of the redundancy is fixed, the number of errors that the code surely detects decreases or remains the same.

As we can see from Fig. 4 and Fig. 5, the best result from the aspect of the number of errors that the code surely detects is achieved when the length of the input blocks is 5 symbols from the alphabet ∑ and the redundancy has length 15 bits. This means that from the aspect of the number of errors that the code surely detects, it is best to divide the input message into blocks of length 5 symbols from ∑ and to code these blocks such that the redundancy is 15 bits (i.e., to choose the parameter *r* in the model to be 4). In this case the code will detect for sure every incorrectly transmitted coded block with up to 5 incorrectly transmitted bits.

4. Conclusion

The results for the number of errors that the code surely detects when the given binary matrices *A*, *B* of order 3×3 and zero matrix *C* of order 1×3 are used for coding are the following:

In the case when the redundancy has length 9 bits, the code surely detects up to 2 incorrectly transmitted bits when the length of the input block is smaller than or equal to 6 symbols from ∑. For input blocks with length greater than or equal to 7 symbols, the code surely detects 1 incorrectly transmitted bit.

When the redundancy is 12 bits, the code surely detects up to 4 incorrectly transmitted bits when the length of the input block is 4 or 5 symbols from the alphabet ∑, up to 3 incorrectly transmitted bits when the input block has length 6 symbols, up to 2 incorrectly transmitted bits when the input block has length 7 symbols and 1 incorrectly transmitted bit when the input block has length greater than or equal to 8 symbols.

When the redundancy has length 15 bits, the code surely detects up to 5 incorrectly transmitted bits when the length of the input block is 5 symbols, up to 4 incorrectly transmitted bits when the length of the input block is 6 symbols, up to 2 incorrectly transmitted bits when the length of the input blocks is 7 or 8 symbols. The code surely detects 1 incorrectly transmitted bit when the length of the input block is greater than or equal to 9 symbols.

When the length of the input blocks is fixed, the number of errors the code surely detects does not decrease with increasing redundancy length. If the length of the redundancy is fixed, then when the length of the input block increases, the number of errors that the code surely detects decreases or remains the same.

In order to achieve largest number of surely detected incorrectly transmitted bits, the input message should be divided into blocks of length 5 symbols and each block to be coded separately such that the parameter of the model is *r*=4.

5. References

[1] Ziemer R. E., Tranter W. H., "Principles of communication: systems, modulation, and noise," Wiley, 2014.
[2] Bossert M., "Channel Coding for Telecommunications," Wiley, 1999.
[3] Fenwick P., "Checksums and Error Control," Introduction to Computer Data Representation, Bentham Science Publishers, 2014.
[4] Verhoeff, J., "Error Detecting Decimal Codes (Tract 29)," The Mathematical Centre, Amsterdam, 1969.
[5] Peterson W. W., Brown D. T., "Cyclic Codes for Error Detection," Proceedings of the IRE. 49 (1), 1961, pp. 228–235.
[6] Koopman P., "32-Bit Cyclic Redundancy Codes for Internet Applications," Proceedings of The International Conference on Dependable Systems and Networks, 2002, pp. 459–468.
[7] Bakeva V., Ilievska N., "A probabilistic model of error-detecting codes based on quasigroups", Quasigroups and Related Systems, vol. 17, 2009, pp. 135-148.
[8] Ilievska N., "Proving the probability of undetected errors for an error-detecting code based on quasigroups," Quasigroups and Related Systems, vol. 22, 2014, pp. 223 – 246.
[9] Ilievska N., Gligoroski D., "Simulation of some new models of error-detecting codes," Proceedings of the 22nd Telecommunications Forum Telfor, Belgrade, Serbia, pp. 395-398, 2014.
[10] Klove T., "Codes for Error Detection," World Scientific, 2007.