THROUGHPUT ANALYSIS OF STATISTICALLY SYNCHRONIZABLE CYCLIC CODES

Jakov Stojanović¹, Dragana Bajić²

¹Network Planning Centre, Mobtel Serbia, Bulevar Nikole Tesle 42a, 11070 Belgrade, Serbia and Montenegro, e-mail: jakov.stojanovic@mobtel.co.yu
²Department of Communications and Signal Processing, Faculty of Technical Sciences, University of Novi Sad, Fruskogorska 11, 21000 Novi Sad, Serbia and Montenegro e-mail: LMCDRA@eunet.yu

1 INTRODUCTION

Numerous families of block-codes have been designed for synchronization recovery, most of them with error correcting possibilities [1, 2]. However, if powerful forward error correction (FEC) is main purpose of coding, synchronization error correcting codes are not solution as their performance is poorer. On the other hand, a simple insertion of a deterministic marker that indicates the code-word boundaries would be a waste for block-codes (and other codes as well), since their inherent redundancy is sufficient for synchronization acquisition and recovery. The statistical procedures are based upon the assumption that the code-words would be erroneous if their position within the received data is not chosen correctly (Fig. 1).

Fortunately, a minimal trellis decoding procedure of block-codes [3, 4 and 5] overcomes these difficulties. At the receiving end, a re-arrangement of the incoming bits is necessary in order to construct an appropriate trellis. Therefore, a stream of bits can be scrambled at the transmitting end, with no consequences for the receiver (just a different re-arrangement of the look-up table), but receiving offset $O \neq 0$ would not be likely to simulate a code-word. Furthermore, this decoding procedure yields Hamming distance of the survived path (code-word) as one of the decoding results, a feature important for distinguishing erroneous (“bad”) from “good” code-words for synchronization purposes in a case of multiple-error correcting codes.

This assumption is not true considering the cyclic codes, since each cyclic shift forms another code-word. If the slip of a single bit has occurred ($O=1$ in Fig. 1), the probability of a code-word simulation might reach 0.5 if the code is without a parity bit. Further on, if a code is constructed to correct more than a single error, their frequency is expected to occur, thus increasing the uncertainty whether the errors are a consequence of channel distortion or of a synchronisation slips. It would be quite unreasonable to implement a powerful correcting code and then have a frequent false “lost synchronisation” alarm that starts re-alignment procedure, only because a code-word with a single (correctable!!!) error occurs. So it was long assumed that statistical synchronisation is not suitable for cyclic codes.

A $(n,k) = (24,12)$ Golay code will be used as an illustrative example within this paper. Fig. 2 shows Hamming distances of survived paths for different offsets. The plots are obtained by brute force; all offset positions of all possible pairs of code-words were decoded. If a scrambler has not been applied (a), at small offset positions the paths with small Hamming distance would be the most probable, while the $H>2$ paths are either less probable, or do not exist. Therefore, a criterion for a “bad” code-word would be each code-word with $H>0$, an obvious nonsense since the code can correct more than a single error! On the other hand, if a scrambler is applied (b), the $H=3$ and $H=4$ result would be the most probable at all $O \neq 0$ offsets. Therefore, these distances might be used as a criterion for “bad” code-words.

Fig. 1: Offset in case of synchronization loss

Fig. 2: Probability distribution function of detected Hamming distances: a) cyclic code (upper); b) scrambled code (lower)
The purpose of this paper is to perform a throughput comparison of various algorithms for statistical synchronization, considering the channel bit error rate, probability of cycle-slip, criterion for “bad” code-words, and also to compare the reliability of statistical synchronization to the one when a deterministic synchronization marker is inserted [6].

II EXPECTED VALUE OF HOLDING AND OOF TIMES

For synchronization monitoring purposes, “bad” code-words are counted until a specific value $M$ is reached. Monitoring algorithm has to fulfill two contradictory requirements: to announce synchronization failure alarm as soon as possible if a synchronization failure has occurred (offset $O$<0) and to hold out against channel errors thus preventing false alarms as long as possible ($O$=0). Classical communication systems implement single dimensional counters [7], a $M$-successive counter ($M$ consecutive erroneous words announce an alarm), or, less usual, random-walk counter, with the expected number of code words between the alarm announcements (entering state $M$) is [8-10]:

$$
E[n] = \left\{ \begin{array}{l}
\frac{q}{p} \left[ \frac{1}{1 - q} \frac{1 - (1-q)^M}{1 - q^M} \right] \\
(2 - (2^M - 1) + M - K) (M - K + 1), \quad t = \frac{q}{p} = 1
\end{array} \right.
$$

(1)

where $p$ is a probability of erroneous code word, $q=1-p$.

Lower part of Eq. (1) is valid for cyclic codes without a parity-check bit, for offsets next to the correct position.

![Fig. 3: A single dimensional counter](image)

Values of $K$ equalling to $M$-1 or to 1 cover the mentioned counter types (Fig. 3).

Statistical synchronization solutions prefer two-dimensional types, [11], probably because the very first one (based upon the statistical properties of sampled speech) [12] was of this kind. One of the two counters counts erroneous code words and sets the alarm if its maximum $M$ is reached. The other one counts correct code words only [11] (“square counter”, Fig. 4a) or all the code words (“diamond counter”, Fig. 4b). Reaching the maximum $N$, both counters are reset. The corresponding expected number of code words is evaluated as:

$$
E[n] = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left( \left( \frac{q}{p} \right)^j \cdot q^{M-1-j} \right)
$$

Sliding window ($M$ out of $N$) counter announces the alarm if $M$ out of $N$ last received code words are “bad”. For each pair ($M$, $N$) a separate closed form formula would have to be derived, or an original simple solution in matrix form used [13]. This evaluation is shown in the Appendix.

In all the previous equations, $q$ is a probability that a codeword would be assumed to be correct (in synchronization monitoring sense). It depends upon the offset, allowed distance $H$ and channel bit-error rate $P_e$, i.e. $q = q(O,H,P_e)$ and $p = 1-q$.

If there is no synchronization failure ($O = 0$), a false alarm is announced due to channel errors. Probability that the code word would be presumed correct depends upon the well known distance spectrum $A_j$ of the Golay (24,12) code [e.g. 1] and can be evaluated as:

$$
q(O = 0,H,P_e) = \sum_{j=0,6,12,24} A_j \left( P_e^j \cdot Q_e^{24-j} + \sum_{e=1}^{H} \sum_{i=0}^{\delta_e} \frac{1}{1 + \delta_e} \left( \frac{24-j}{e-i} \right) \cdot P_e^{-r_{e-i}} \cdot Q_e^{24-j-e_{i}} \right)
$$

(3)

$O$>0, probability $q$ has to be evaluated for each position separately, using the Hamming distance p.d.f. from Fig. 2. The results considering the expected values of holding and OOF times are shown in Figs 5-10.

![Fig. 4: Two-dimensional counters: a) “square”; b) “diamond”](image)
Fig. 5: Logarithm of holding time for different counter capacity $M, H=0$

Fig. 6: Logarithm of holding time for different maximal Hamming distance

Fig. 7: Out-of-sync. time for different counter capacity, worst case $O=1$

Fig. 8: Out-of-sync. time for different allowed Hamming distance, worst case $O=1$

Fig. 9: Out-of-sync. time for different offset position – classical code-word

Fig. 10: Out-of-sync. time for different offset position – scrambled code-word
Since the values of expected value of holding time $E\{n\}$ range from 1 to $10^9$, its logarithm is plotted in Figs. 5 and 6. Single-dimensional counters are the best considering this criterion, while “square” one shows the poorest performances. However, fast alarm announcement is an advantage for out-of-sync time (Figs. 7 and 8) that should be as small as possible. This reveals another reason of frequent implementation of “square” counter for statistical synchronization of optical line codes. $H > 0$ considerably improves time between false alarms, with no remarkable influence upon out-of-sync time if a code word is scrambled: its expected value approximately equals to counter capacity $M$. Bit error rate (BER) shortens holding time. Bigger BER values set alarms frequently if $H = 0$. Increase of counter capacity $M$ increases holding time but also out-of-sync time, so its choice must be a careful compromise. Scrambling the code word has, obviously, no effect to holding time. But, if a code word has remained cyclic, an alarm might never be announced at $O = 1$ or 23, as $q(0=1, H=2, P_e = 0) = 1$ (Fig. 2a)!

Even if $H = 0$, values for $E\{n\}$ are a couple of times bigger than the corresponding ones of scrambled words. On the other hand, implementing statistical synchronization with $H = 0$ lets a system to force a resynchronization if several code words have remained cyclic, an alarm might never be announced, while a code itself would be capable of correcting this several errors within a single word! Figs. 9 and 10 show out of sync. time at various offset position for the quickest (“square”) and slower (“M-successive”) counters. While scrambled code keeps this value reasonably well only for $5 < O < 19$. This justifies the implementation of code scrambling at the transmitting end.

### III EXPECTED VALUE OF LOST DATA

A model that enables average percentage of lost data (a parameter complementary to throughput) evaluation is shown in Fig. 11 (for the universal single-dimensional counter; for the others it can be drawn accordingly). It consists of two sets of states: in synchronism (black ones), and out of sync. (grey ones) for unnoticed occurrence of synchronization loss. Dwelling times of these states is one code word. State “resynchronization” corresponds to acquisition process. Its dwelling time equals to expected duration of resynchronization $\tau$ that depends mainly upon the processor resources, and might range from 4 to 5 to 30 code words for a (24,12) code (analysis not included into this paper). Resynchronization starts immediately upon entering states $M$, so they are incorporated into resynchronization and their dwelling time is 0. The probability that a real sync loss would occur is denoted with $a$ and depends highly upon the robustness of the chosen transmission system.

Probabilities $p_1$ and $p_2$ are joint probabilities evaluated as:

$$p_1 = Pr\{in\, sync.\} \cdot Pr\{erroneous\, code\, word, O=0\} = \frac{1-a}{a} \cdot p$$

$$p_2 = Pr\{out\, of\, sync.\, occurred\} \cdot Pr\{err.\, code\, word, O>0\} = a \cdot p_1$$

$$q_1 = \frac{1-a}{a} \cdot q$$

$$q_2 = \frac{a \cdot p_1}{a}$$

$$p_1 = Pr\{FAW \, in \, error\}, 0 < \text{offset} < F$$

$$q_1 = Pr\{FAW \, correct\}$$

(4)

It is easy to obtain a state – transition matrix from the state diagram and evaluate vector of state – selection probabilities

$$\pi = [\pi_0, \pi_1, \ldots, \pi_{40}, \pi_{41}, \pi_{42}, \pi_{43}, \pi_{44}, \pi_{45}, \pi_{46}, \pi_{47}, \pi_{48}, \pi_{49}, \pi_{50}, \pi_{51}, \pi_{52}, \pi_{53}, \pi_{54}, \pi_{55}, \pi_{56}, \pi_{57}, \pi_{58}, \pi_{59}, \pi_{60}, \pi_{61}, \pi_{62}, \pi_{63}, \pi_{64}, \pi_{65}, \pi_{66}, \pi_{67}, \pi_{68}, \pi_{69}, \pi_{70}, \pi_{71}, \pi_{72}, \pi_{73}, \pi_{74}, \pi_{75}, \pi_{76}, \pi_{77}, \pi_{78}, \pi_{79}, \pi_{80}, \pi_{81}, \pi_{82}, \pi_{83}, \pi_{84}, \pi_{85}, \pi_{86}, \pi_{87}, \pi_{88}, \pi_{89}, \pi_{90}, \pi_{91}, \pi_{92}, \pi_{93}, \pi_{94}, \pi_{95}, \pi_{96}, \pi_{97}, \pi_{98}, \pi_{99}, \pi_{100}]$$

$$= [\pi_0, \pi_1, \pi_2, \pi_3]$$

where indices 1 and 0 correspond to in-sync and out-of-sync sets of states. Vector of state dwelling times [number of frames] is:

$$\tau = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$= [\tau_1 \ \tau_0 \ \tau_R]$$

The transition matrix is:

$$P = \begin{bmatrix}
P_1 & P_2 & R_{12} \\
R & P_3 & R_3
\end{bmatrix}$$

$$R = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}$$

$$R_{12} = [0 \ 0 \ 0 \ 0 \ p_1 + p_2]$$

$$R_3 = [0 \ 0 \ 0 \ 0 \ p_3]$$

(5)

From these probabilities steady state probabilities are evaluated [14], knowing the state dwelling time vector $\tau = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$. Summing steady probabilities of out-of-sync states including resynchronization, average number of lost code-words vs. all the received code-words is obtained.

$$\text{lost data} = \frac{\sum_{i=0}^{N} \tau_{0i} \cdot \pi_{0i} + \pi_R \cdot \tau_{R}}{\sum_{i=0}^{N} \pi_{L} \cdot \tau_{Li} + \sum_{i=0}^{N} \pi_{0i} \cdot \tau_{0i} + \pi_R \cdot \tau_{R}} \times 100\%$$

Some of the results are shown within the Fig. 12. Data loss is a criterion that incorporates holding time, out-of-frame time and acquisition time. It is plotted against various parameters. It is interesting that, although the 2-dimensional counters are implemented statistical synchronization with $H = 2$, cyclic code behaves reasonably well only for $5 < O < 19$. This justifies the implementation of code scrambling at the transmitting end.
ones present logarithms as the real values are too big. Therefore, out-of-sync time is dominant factor and the counters optimal for it would remain optimal.

**Fig. 11**: A model for data loss evaluation (single-dimensional)

**Fig. 12**: Expected value of lost data [%] as a function of a) resynchronization time $\tau$; b) Bit-error rate $P_e$; c) Counter capacity $M$; d) probability of synchronization failure $a$.
IV CONCLUSION
A comparative analysis of various synchronisation - monitoring algorithms for block-codes is performed. Various monitoring procedures for classical analytical approach considering the expected values of holding time and expected out-of-synchronisation times are proposed. A model and method for analysis of expected amount of data loss during operation time is proposed as well. Extended (24, 12) binary cyclic Golay code is used as an illustrative example. It was shown that the classical M-successive counter, “borrowed” from telecommunication transport systems, achieves the best results. It was also shown that, if minimal trellis decoding is used, cyclic codes, contrary to the general belief, can be statistically synchronized.

ACKNOWLEDGMENT
The authors wish to thank Ivan Stanojevic, who thought about “diamond” counter, and Prof. Vojin Senk, who has suggested the idea of “M out of N” counter.

REFERENCES


APPENDIX: M out of N COUNTER
The procedure for evaluating the “first passage time” for M out of N counter is shown using the illustrative example N = 4, with 0<M<5.
The complete state-transition diagram is shown in Fig. A1.

To evaluate the first passage time it is necessary to define the start set S and the finishing set F of states, then the set of “forbidden” states (the ones that cannot be reached) and the remaining set of states R. Then, the corresponding new state-transition (kxk) matrix P is to be derived from the complete matrix $P_c$. At last, implementing the idea from [13], state selection probabilities are evaluated from

$$P \cdot P = p$$  \hspace{1cm} (A1)

with the additional condition:

$$P \cdot I^T = 1$$  \hspace{1cm} (A2)

since the last equation of (A1) is statistically dependent. Vectors $p$ and $I$ are:

$$p = \left[ \pi_0 \ \pi_1 \ \pi_2 \ \cdots \ \pi_{k-3} \ \pi_{k-2} \ \pi_{k-1} \right]$$

$$I = \left[ \begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{array} \right]$$

Having evaluated all this, the expected value of number of steps necessary to reach the finishing states is:
\[ E[n] = \sum_{i\in R_S} \pi_i \sum_{j\in F} \Pr\{j/i\} \]  
\[ \text{where } \Pr\{j/i\} \text{ is a transition probability of the derived matrix } \Pi. \]

The state transition diagrams for 4 out of 4, 3 out of 4, 2 out of 4 and 1 out of 4 case is shown in Figs. A2 – A5. The corresponding matrices \( \Pi \), are shown beneath each figure.

\[
\begin{align*}
P &= \begin{bmatrix} q & p \\ 1 & 0 \end{bmatrix} \end{align*}
\]

At last, it is worth mentioning that \( M \) out of \( M \) algorithm actually presents \( M \)-successive counter! Knowing this, a classical expression for \( K=M-1 \) (Fig. 3, Eq. (1)), i.e.

\[
E[n] = \frac{1-p^4}{q \cdot p^4}
\]

can be derived in a more tedious way, hereafter shown for \( M = 4 \):

\[
\begin{align*}
\pi_1 &= q \cdot (\pi_2 + \pi_3) \\
\pi_2 &= q \cdot (\pi_4 + \pi_6) \\
\pi_3 &= q \cdot (\pi_5 + \pi_7) \\
\pi_4 &= q \cdot (\pi_8 + \pi_9) \\
\pi_5 &= q \cdot (\pi_6 + \pi_7) \\
\pi_6 &= q \cdot (\pi_8 + \pi_9)
\end{align*}
\]
Abstract: A comparative analysis of various synchronisation-monitoring algorithms for block-codes is performed. Besides the classical analytical approach considering the expected values of holding time and expected out-of-synchronisation time, a model and method for analysis of expected amount of data loss during operation time is proposed and applied for the analysis. Extended (24,12) binary cyclic Golay code with minimal trellis decoding is used as an illustrative example, showing that cyclic code can be statistically synchronized.

Sažetak: Analizirani su razni algoritmi za nadgledanje sinhronizacije blok-kodova. Pored uobičajenog pristupa koji podrazumeva matematičko očekivane vrijeme držanja i vrijeme ispadanja, uvedeni su i model i metod za analizu matematičkog očekivanja količine izgubljenih podataka. Za ilustraciju pomenutih metoda iskorišćen je (24,12) prošireni Goljev kod sa minimalnim trellis dekodovanjem. Pored ostalog, pokazalo se da, suprotno uvređenom mišljenju, statistička sinhronizacija može da se primeni na ciklične kodove.

ANALIZA PROPUSNOSTI CIKLIČNIH KODOVA SA STATISTIČKOM SINHRONIZACIJOM, Jakov Stojanović, Dragana Bajić