EMBEDDING NOISE PREDICTION INTO LIST-VITERBI DECODING USING ERROR DETECTION CODES FOR MAGNETIC TAPE SYSTEMS

SUAYB S. ARSLAN, JAEWOOK LEE
TURGUY GOKER

QUANTUM CORPORATION

IRVINE, CA

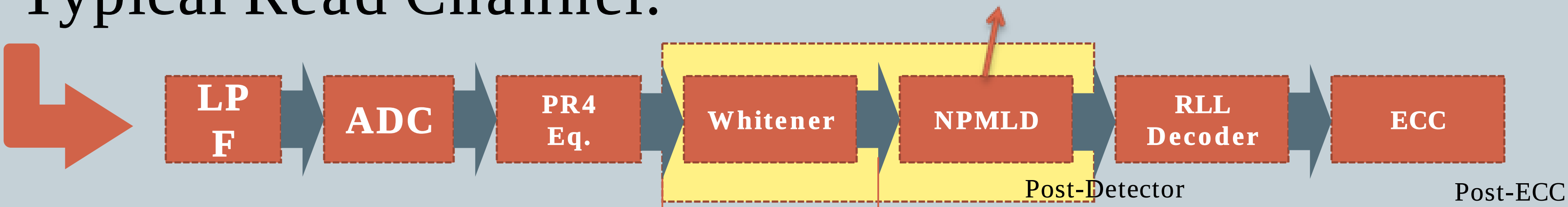
Outline

- Background
- System diagram
- Problem statement and goals
- Proposed solution:
  - Noise predictive List-Viterbi Algorithm
- Numerical results.

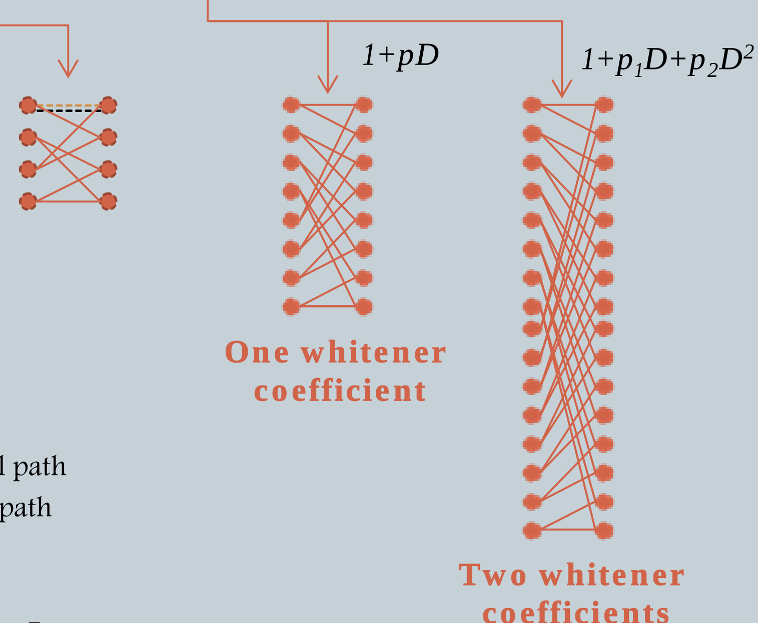
Background

E. Evangelos and W. Hirt, "Improving performance of PRML/EPRML through noise prediction," *IEEE Trans. Magn.*, vol. 32, no. 5, pp. 3968–3970, Sept. 1996.

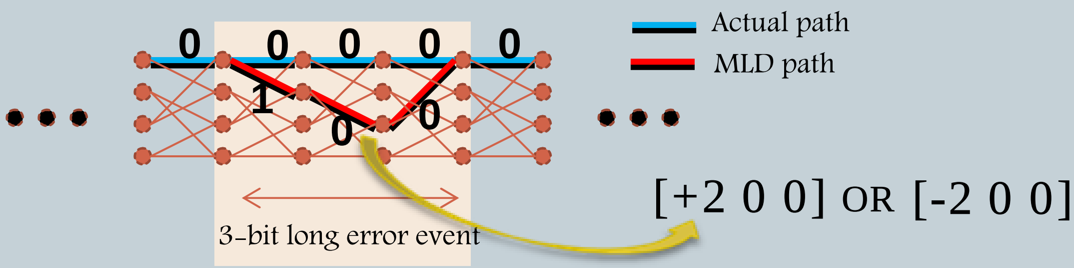
- Typical Read Channel:



- Noise predictive MLD:
  - As the number of whitener coefficients increase, the size of the trellis increases exponentially.
  - We use past decisions to reduce the size of the trellis.



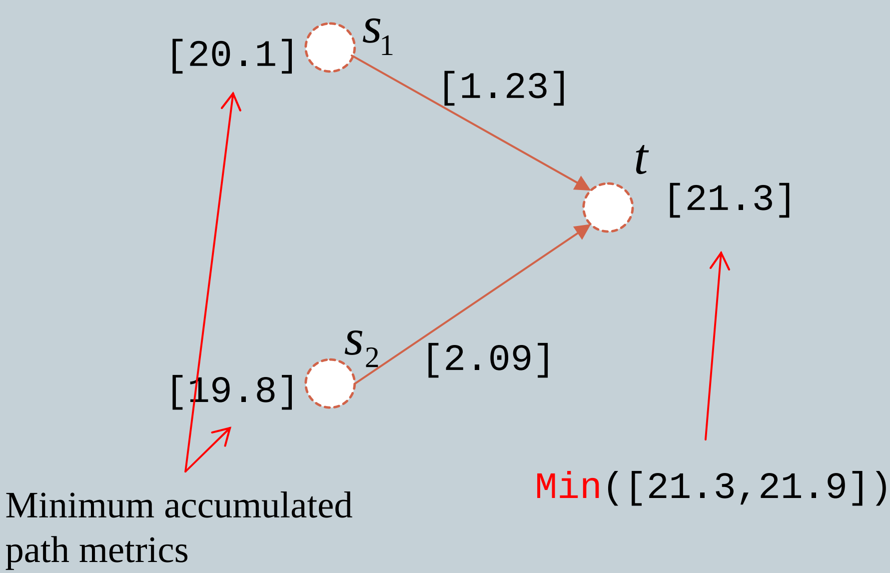
- Error Events on Trellis:



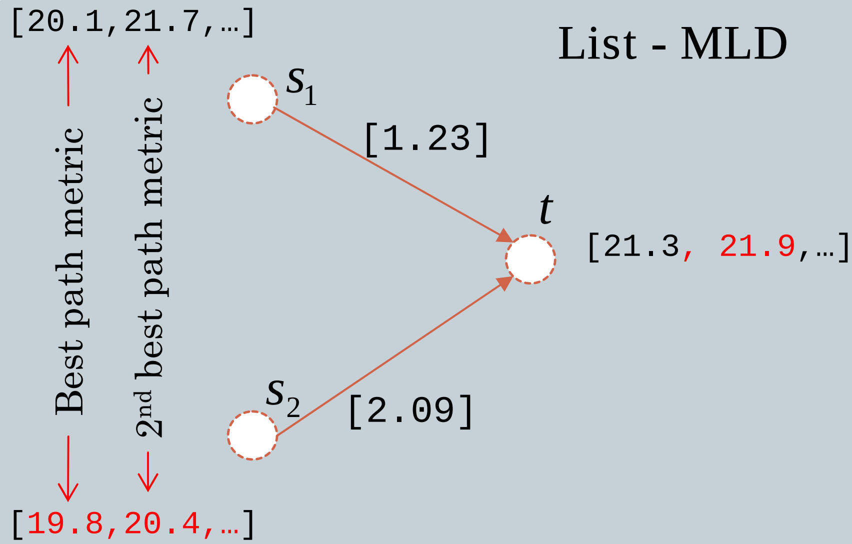
Background

Conventional MLD

Each path metric corresponds to a path and its associated decoded bit stream.



List - MLD



For state t →

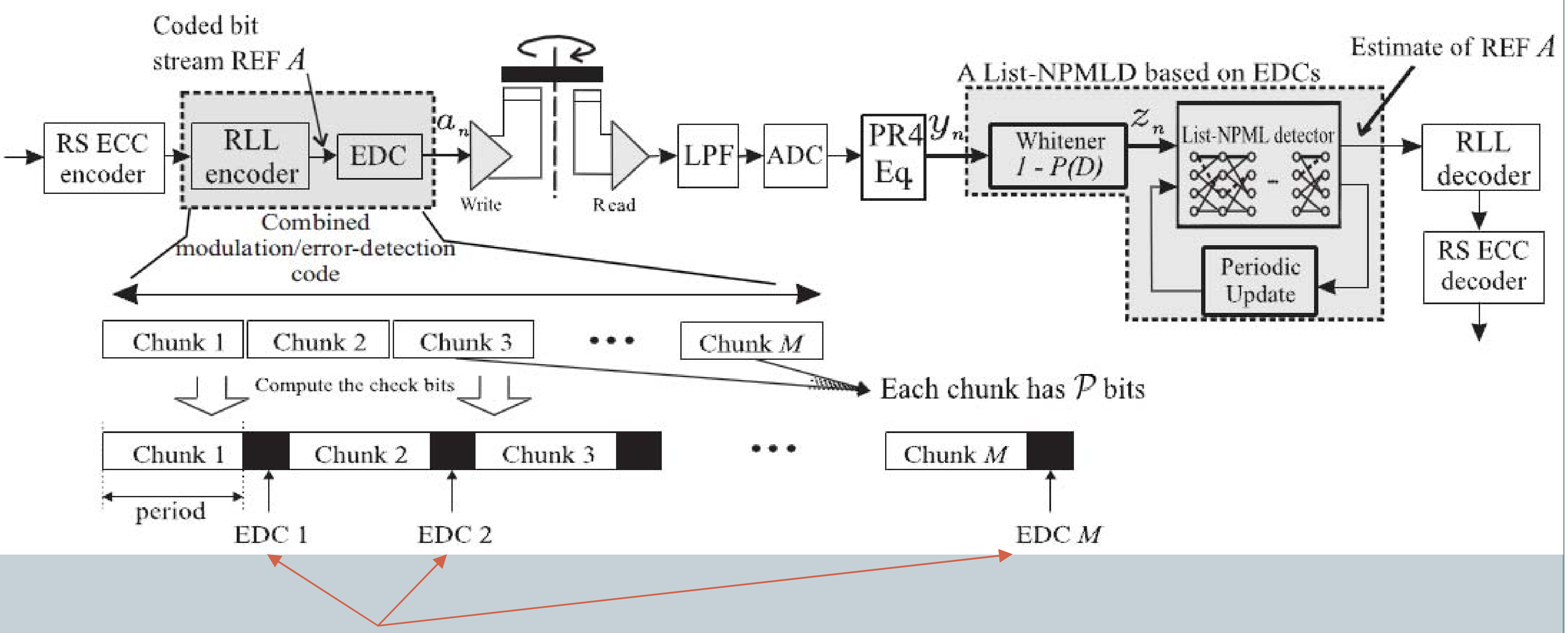
Best path metric: `Min([21.3,21.9]) = 21.3`

Second best path metric:

`Min([21.9, 22.9, 22.5]) = 21.9`

Claim: Once can show that if we like to find the first N best sequence, it suffices to save the most likely N best accumulated metrics in each state.

System diagram



Error Detection Code parities.

Problem statement and goals

- Most of the error events at the output of an NPMLD are caused by a specific subset of all the error events.
- Large number of error events at the output of the NPMLD impacts the post-ECC error rates quite dramatically.
- **Goals and Challenges:**
  - Reduce bit errors by a way of detecting and correcting the error events.
  - False alarm rate for error detection and/or correction shall be minimized.
  - If additional detection codes (e.g. CRC, parity-check code) are required to detect error events reliably, they should not add too much redundancy to the coded channel data.
  - Proposed error detection codes (e.g. CRC, parity-check code) should be concatenated efficiently with existing RLL codes and/or RS codes used for LTO format.
  - Complexity of the proposed scheme should be reasonable.

Noise predictive List-Viterbi detection

- General procedure:
  1. Noise predictive List-Viterbi detection algorithm simultaneously produces a rank ordered list of the N globally best candidates for each state of the trellis.
     - Traditional Viterbi algorithm finds just one most likely candidate.
  2. Select a correct sequence/path out of N candidates based on error detection codes (EDCs):
     - Most of the time, the 1st most probable sequence will check the EDC, but others will not check it.
     - However, in the case of error events, the 1st most probable sequence will not check the EDC, but one out of others may check (N depends on the coverage of the error events)
     - Any sequence that checks the EDC will be the final output of the algorithm.

Noise predictive List-Viterbi detection

- Consider an example with 4-state trellis and one CWI-4 is divided into M equal size chunks. Following figure shows how we do the updates:

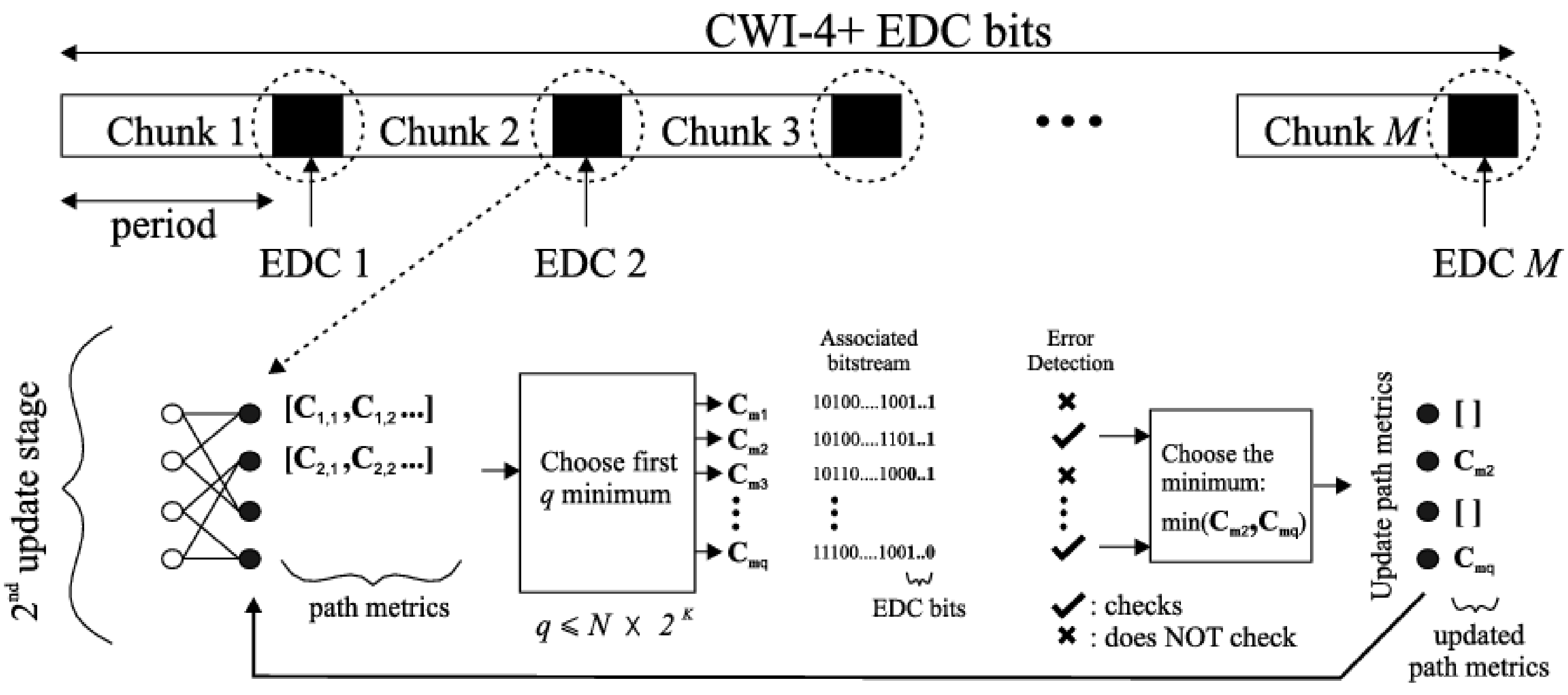
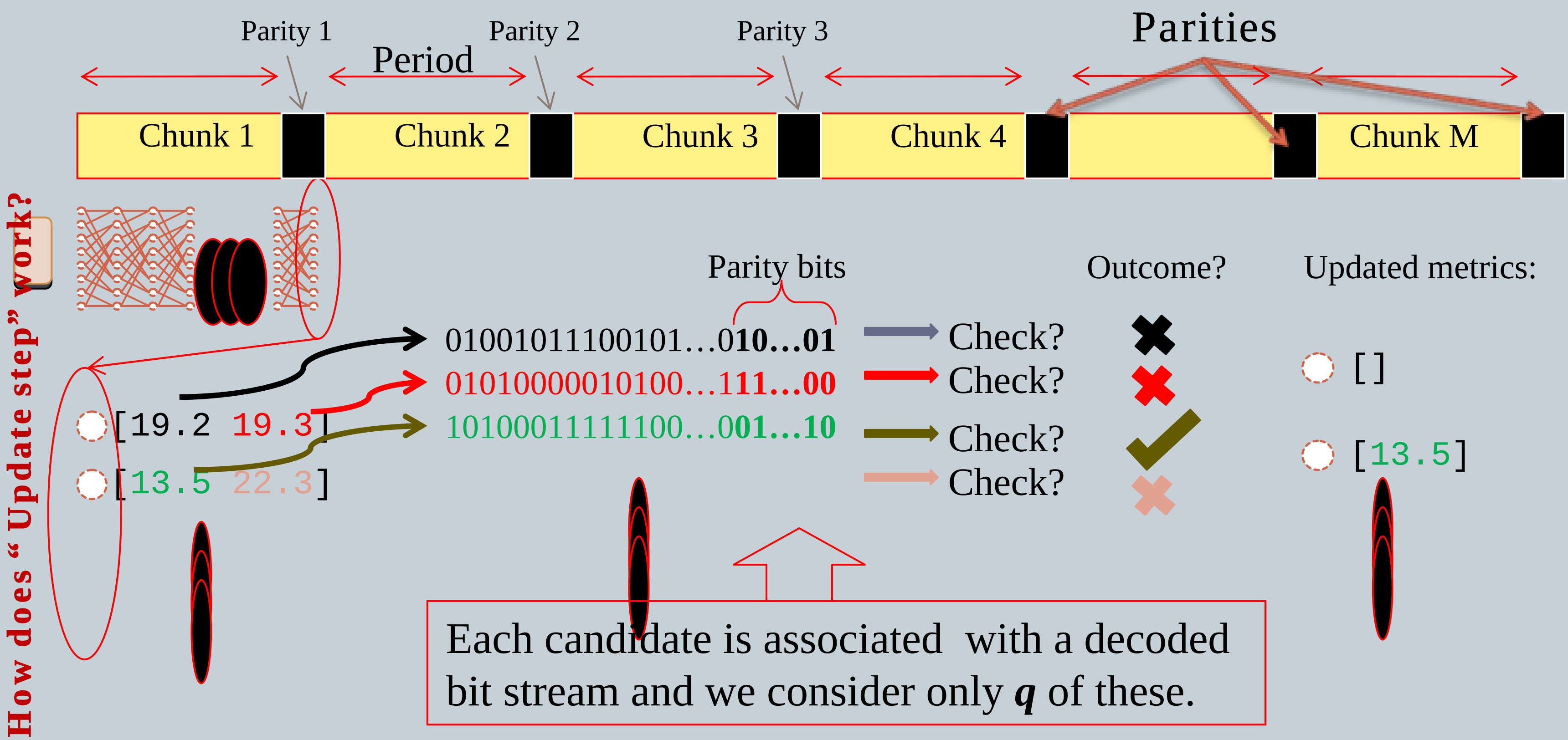


Fig. 4. Update stage of the proposed algorithm. EDC: Error detection code.

Noise predictive List-Viterbi detection

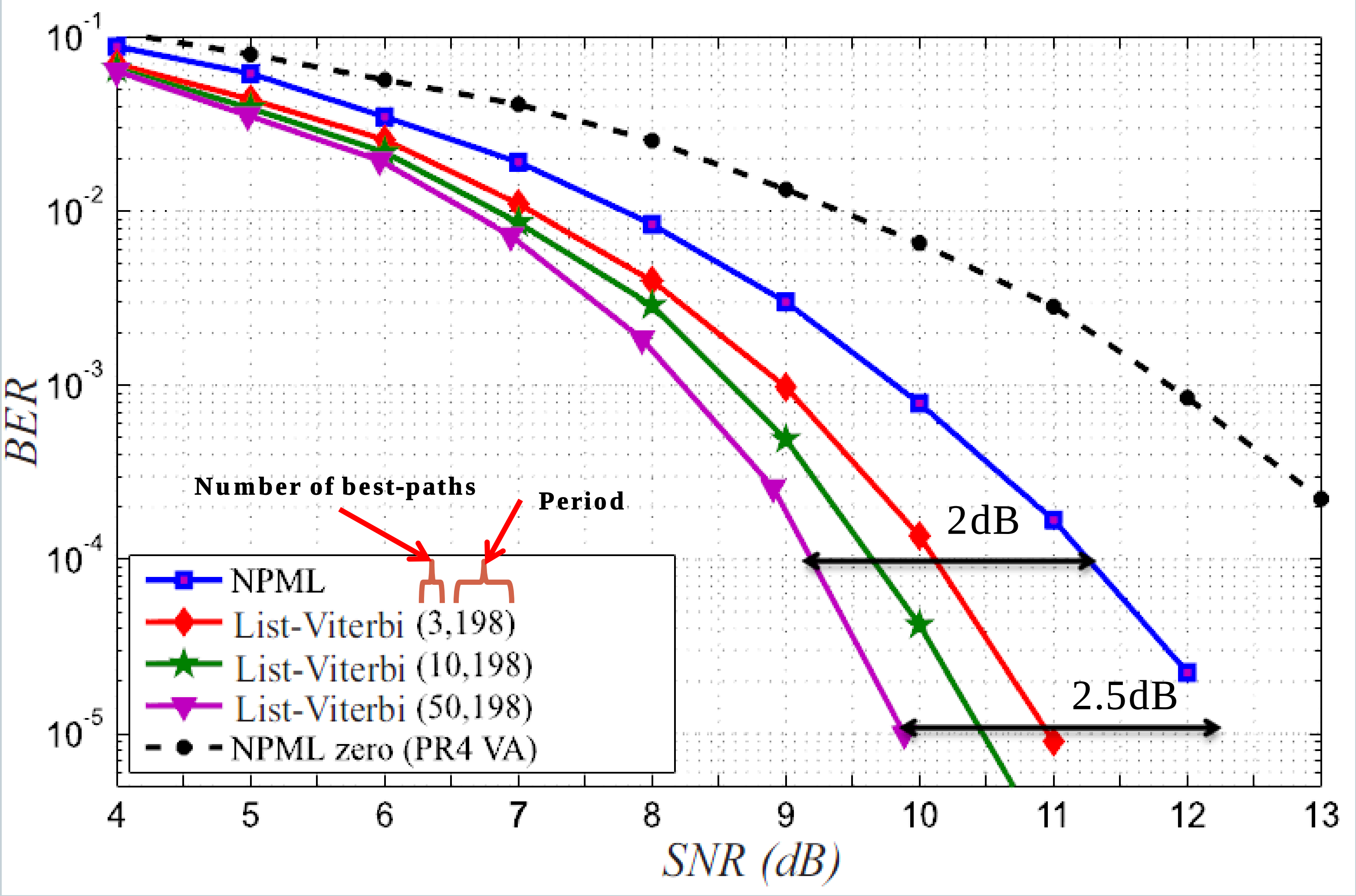
- After each **update step**, accumulated metrics of each state is updated according to the EDC check results.



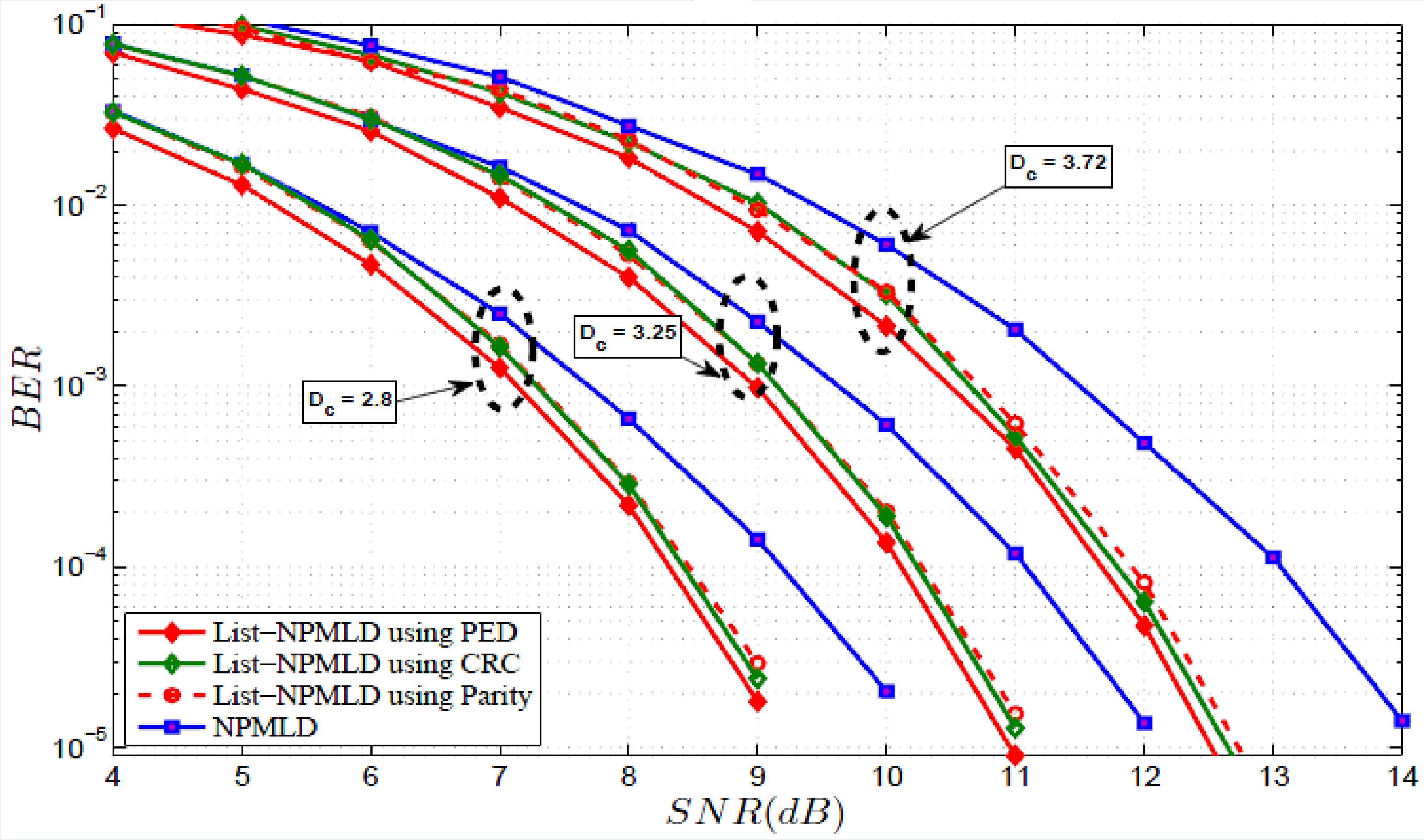
Numerical Results

- Pr4 signaling, 4 state trellis, 3 bits in feedback
- Lorentzian Channel model
  - Both white and colored noise is assumed. Noise power is equally shared.
  - Perfect timing recovery
  - LMS linear prediction for noise.
- We both used Perfect Error Detection (PED), parity as well as CRC check bits for error detection.
- We used different linear densities: $D_c = PW50/T$ where $PW50$ is the pulse width measured at half the peak amplitude of the channel step response, T is the bit period.
  - NP - List-Viterbi: 2.8, 3.25 and 3.72. (with P=198bits)
  - NPMLD: 2.76, 3.2 and 3.66.

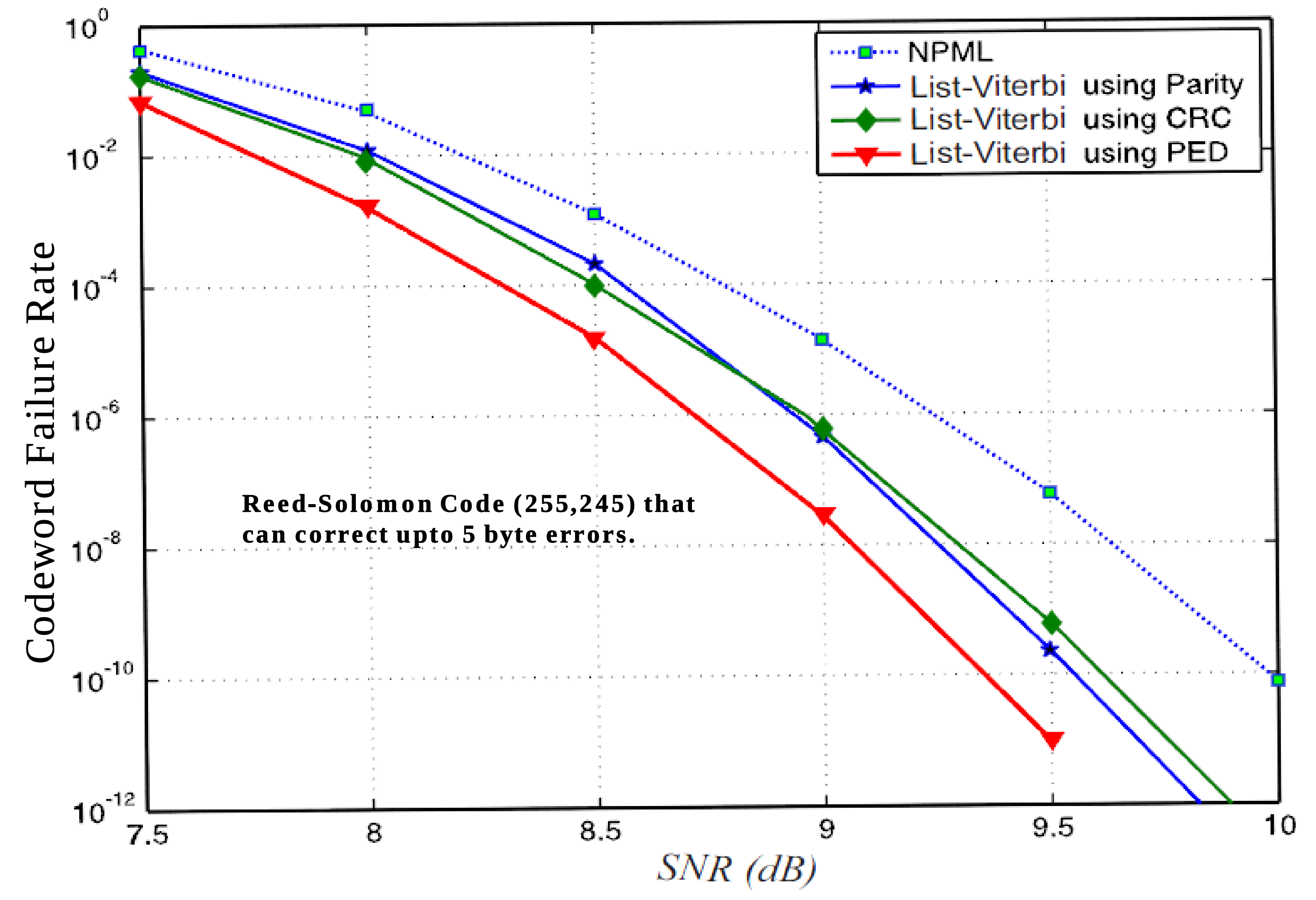
Numerical Results -PED



Numerical Results – EDCs with N=3, P=198.



Numerical Results – EDCs with N=3, P=198, D_c=3.25



Numerical Results – Tape Data

Error Event	NPMLD	LNPMLD(3)	LNPMLD(50)
"1"-bit EE	13365 (53.96%)	5854 (56.36%)	1843 (62.6%)
"2"-bit EE	4744 (19.15%)	2117 (20.4%)	550 (18.7%)
"3"-bit EE	3456 (13.9%)	1146 (11.03%)	194 (6.6%)
"4"-bit EE	1004 (4.05%)	338 (3.25%)	56 (1.9%)
"5"-bit EE	374 (1.51%)	102 (1%)	23 (0.8%)
TOTAL	24769 (100%)	10387 (100%)	2942 (100%)

- P=198 and D_c=2.3.
- We can correct approx. 60% and 90% of the error events using N=3 and N=50, respectively.
- The frequency of these dominant error events are shown to be roughly the same
- Can be combined with using
  - Outcome: Using the proposed scheme with a post processing method targeting a specific error event distribution.

Conclusions

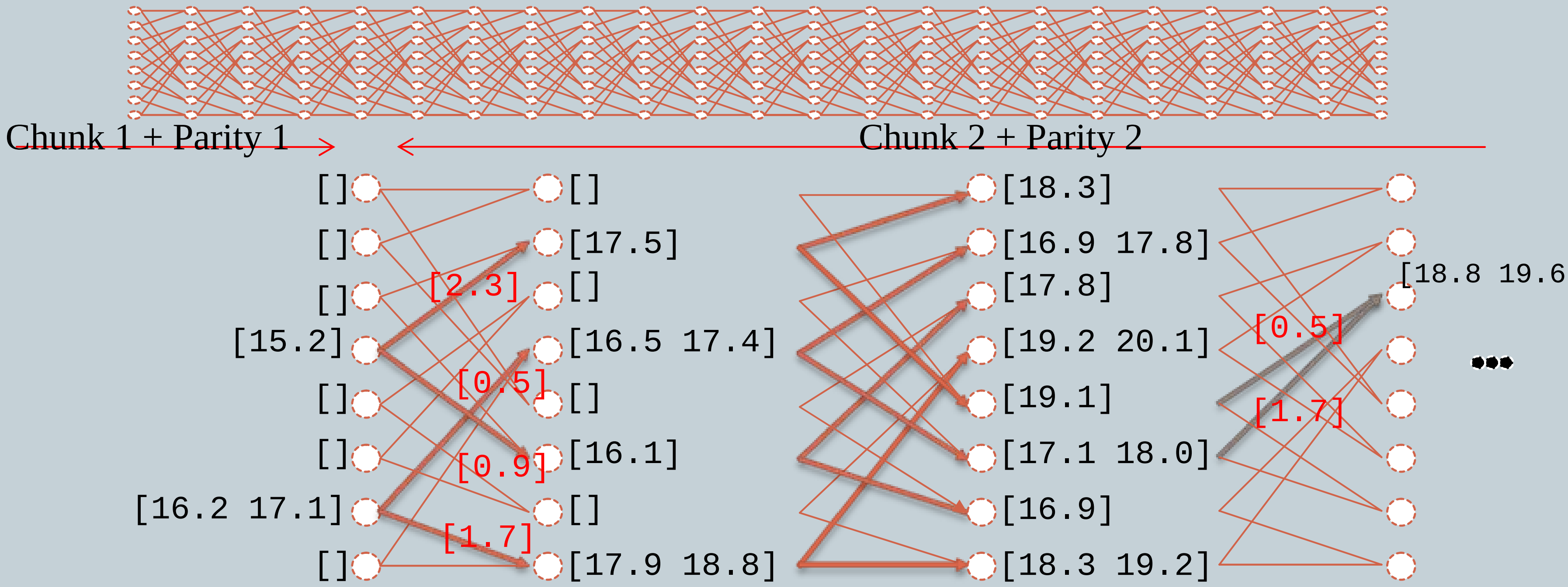
- A noise predictive List-Viterbi algorithm used in conjunction with EDCs is proposed.
- No assumption about error event distributions.
- Significant gains at the detector output.
- Good improvements at the Post-ECC level.
- More details can be found:
  - **S. S. Arslan, J. Lee and T. Goker**, "Error Event Corrections Using List-NPMLD Decoding and Error Detection Codes," available next month, Vol. 49, No. 7, ***IEEE Transactions on Magnetics***, July 2013.

Appendix

MORE ON THE ALGORITHM DETAILS

Parity Update

Trellis structure of 8-state List-Viterbi detector:



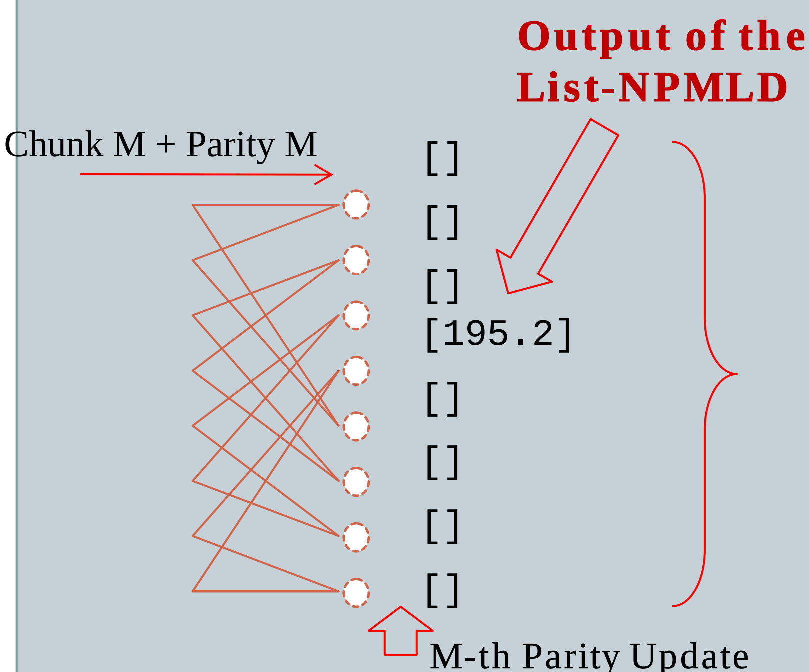
1st Parity Update
(as described in the previous slide)

Second branch metrics are not shown for clarity.

- This example shows the periodic updates using parity error detection in conjunction with Noise predictive List-Viterbi. It also shows how the metric computations are done after each parity update step. It is clear that in each parity update step, the algorithm eliminates (does not record) some of the incorrect paths.

Parity Update

- We keep on updating and compute the accumulated metrics of next states, until we perform the last parity update.
- **In an ideal case:** After the last parity update step, we expect to have the following picture:



- **In any update step, <u>what happens if no candidate path checks the parities</u>?**
  - We choose the path out of all the candidates with the smallest accumulated metric. (We still perform the metric updates for the other paths.)
  - If this happens at the M-th parity update, then the output of the List-NPML detector is the one with the smallest accumulated path metric.