UNIVERSITY OF CAMBRIDGE
Department of Engineering
Signal Processing and Communications Laboratory

# Non-linear constraints for frame synchronisation

or: Why I thought frame synchronisation was the most boring problem in the world and why I am beginning to change my mind

Jossy Sayir

Coding & Modulation Workshop, Munich, 31 July 2015

# Why is synchronisation boring?

Little mention of synchronisation in most books (Wozencraft & Jacobs, Gallager, Lapidoth, Sklar, Madhow)

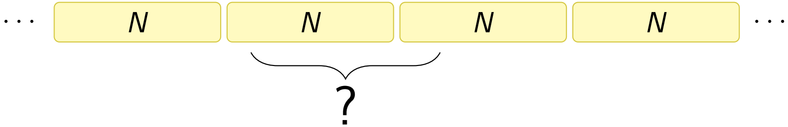
# Why is synchronisation boring?

Little mention of synchronisation in most books (Wozencraft & Jacobs, Gallager, Lapidoth, Sklar, Madhow)

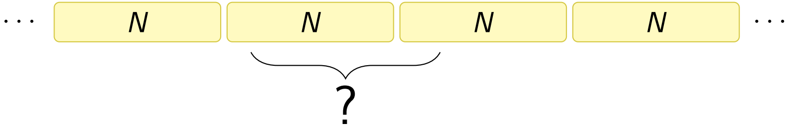

and

Gerhard used to think it wasn't very exciting!

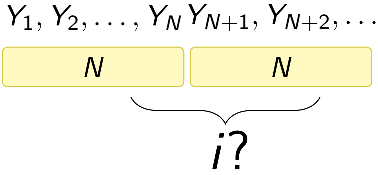
# Periodic frame synchronisation



# Periodic frame synchronisation



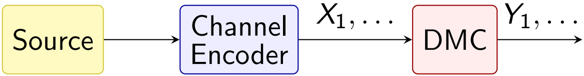

For each period,





Without loss of generality, we assume that $i = 1$ is the correct decision.

# Code-only synchronisation




where every block of length $N$ is a codeword in $\mathcal{C}$. Log likelihood synchroniser:

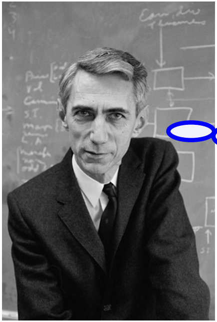
$$\hat{i} = \text{argmax}_i \log \frac{\mathrm{P}(y_i^{i+N-1} | X_i^{i+N-1} \in \mathcal{C})}{\mathrm{P}(y_i^{i+N-1} | X_i^{i+N-1} \notin \mathcal{C})}$$

ML synchroniser:

$$\hat{i} = \text{argmax}_i \max_{x \in \mathcal{C}} \mathrm{P}(y_i^{i+N-1} | X_i^{i+N-1} = x)$$

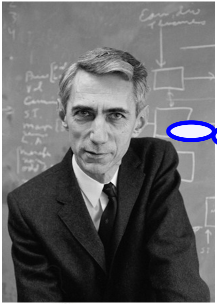
If we use a code family with exponential error decay $\mathrm{P}_e \leq 2^{-\mathcal{E}N}$, then the argument of the max in the ML synchroniser will be on the order $1 - 2^{-\mathcal{E}N}$ for $i = 1$ and $2^{-\mathcal{E}N}$ for $i \neq 1$. The union bound gives a probability of synchronisation failure on the order $P_{\text{sync failure}} \lesssim N2^{-\mathcal{E}N}$.

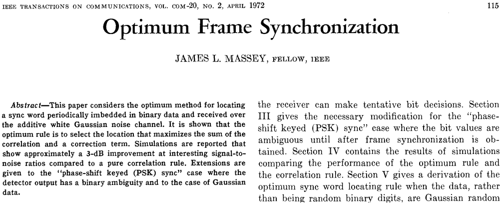
# Information theory and Synchronisation



There is no
synchronisation problem
in information theory!

# Information theory and Synchronisation



IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-20, NO. 2, APRIL 1972 115

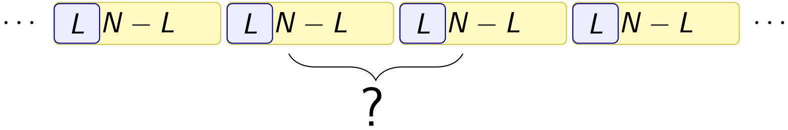
## Optimum Frame Synchronization

JAMES L. MASSEY, FELLOW, IEEE

*Abstract*—This paper considers the optimum method for locating a sync word periodically imbedded in binary data and received over the additive white Gaussian noise channel. It is shown that the optimum rule is to select the location that maximizes the sum of the correlation and a correction term. Simulations are reported that show approximately a 3-dB improvement at interesting signal-to-noise ratios compared to a pure correlation rule. Extensions are given to the "phase-shift keyed (PSK) sync" case where the detector output has a binary ambiguity and to the case of Gaussian data.

the receiver can make tentative bit decisions. Section III gives the necessary modification for the "phase-shift keyed (PSK) sync" case where the bit values are ambiguous until after frame synchronization is obtained. Section IV contains the results of simulations comparing the performance of the optimum rule and the correlation rule. Section V gives a derivation of the optimum sync word locating rule when the data, rather than being random binary digits, are Gaussian random

# Preamble Synchronisation



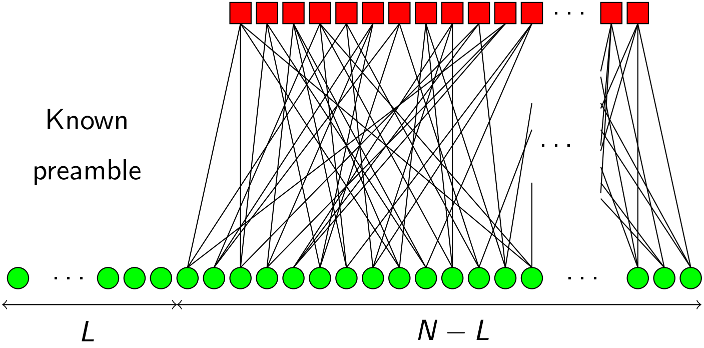

- Codeword $X_{L+1}, \ldots, X_N$ from an $(N-L, K)$ code with rate $R_c = K/(N-L)$
- Fixed preamble $X_1^L = p_1^L$ known to transmitter and receiver
- Overall rate:

$$R = R_C \frac{N-L}{N} = \frac{K}{N}$$

- Preamble = wasted rate (does not contribute when decoding data)
- Optimal (ML) synchroniser:

$$\hat{i} = \text{argmax}_{i=1\ldots N} \prod_{k=1}^{L} \mathrm{P}_{Y|X}(y_{i+k-1}|p_k)$$

# Code aided synchronisation




- Use a subset of $m$ linear constraints to enhance performance
- ML synchroniser:

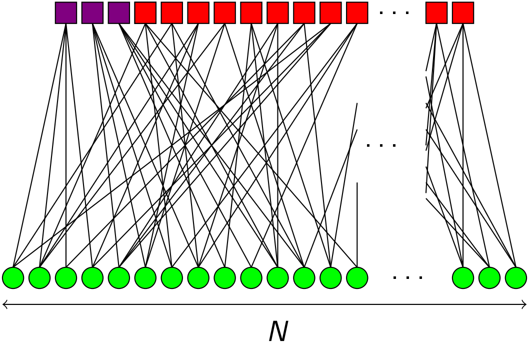
$$\hat{i} = \mathrm{argmax}_{i=1\ldots N} \prod_{k=1}^{L} \mathrm{P}_{Y|X}(y_{i+k-1}|p_k)\,\mathrm{Pr}\{y_L^N|X_L^N \in \mathcal{C}_m\}$$

where $\mathcal{C}_m$ is the set of sequences that satisfies the $m$ constraints

# Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | 5 | | | | | |
| 6 | | | 2 | 1 | | 5 | | |
| | | | 8 | | 9 | | | 2 |
| | | 8 | | | | 3 | | |
| | | 7 | | | | 1 | | |
| | | 5 | | | | 9 | | |
| 4 | | | 3 | | 8 | | | |
| | | 3 | | 2 | 1 | | | 9 |
| | | | | | 4 | | | 7 |

# Permutation constraints for frame synchronisation




- $q$-ary LDPC code with $m$ added non-overlapping permutation (non-linear, SUDOKU-like) constraints, each involving $q$ variables in the length $K$ systematic part
- Rate: $R = \dfrac{\log_q((q!)^m \times q^{K-mq})}{N} = \dfrac{m \log_q q! - mq + K}{N}$

# Synchronisation with permutation constraints

- ML synchronisation:

$$\hat{i} = \operatorname{argmax}_i \prod_{n=0}^{m-1} \sum_{x_1^q \in \mathcal{S}_q} \prod_{k=1}^{q} \mathrm{P}_{Y|X}(y_{i+nq+k}|x_k) \qquad (1)$$

  where $\mathcal{S}_q$ is the symmetric group of permutations of $\{1, 2, \ldots, q\}$.
- Notation: let $\underline{y}_k = y_i^{k+q-1}$, and $\mathrm{P}(\underline{y}_k|\underline{X})$ be the matrix whose $(i,j)$ entry is $\mathrm{P}_{Y|X}(y_{k+i-1}|j)$ where we assume that matrix indices are numbered from 1 to $q$, and for this definition we momentarily assume that $X$ is defined over an alphabet $\{1, 2, \ldots, q\}$
- Then we can express (1) as follows

$$\hat{i} = \operatorname{argmax}_i \prod_{n=0}^{m-1} \operatorname{perm} \mathrm{P}(\underline{y}_{nq+1}|\underline{X})$$

  where $\operatorname{perm}(A)$ denotes the permanent of the matrix $A$.

# Examples on GF(5) for the erasure channel

Known preamble:

| | | | | |
|---|---|---|---|---|
| 4 | 3 | 1 | 0 | 2 |

Preamble sync:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 3 | 2 | 4 | 3 | 1 | 0 | 2 | 1 | 0 | 3 | 0 | 2 | 1 | 3 |

Linear constraint:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 3 | 2 | 4 | 3 | 1 | 0 | 2 | 1 | 0 | 3 | 0 | 2 | 1 | 3 |

Permutation constraint:

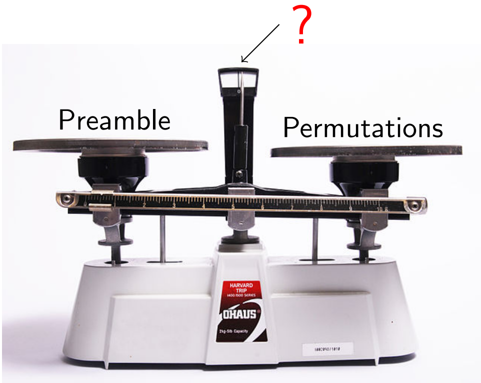
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 3 | 2 | 4 | 3 | 1 | 0 | 2 | 1 | 0 | 3 | 0 | 2 | 1 | 3 |

# Examples on GF(5) for the erasure channel

| | | | | | |
|---|---|---|---|---|---|
| Known preamble: | 4 | 3 | 1 | 0 | 2 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Preamble sync: | 2 | | 3 | 2 | 4 | 3 | | 0 | 2 | 1 | 0 | | 0 | 2 | | 3 |
| Linear constraint: | 2 | | 3 | 2 | 4 | 3 | | 0 | 2 | 1 | 0 | | 0 | 2 | | 3 |
| Permutation constraint: | 2 | | 3 | 2 | 4 | 3 | | 0 | 2 | 1 | 0 | | 0 | 2 | | 3 |

# Fair comparisons: the devil is the detail!




- Synchronisation performance?
- Rate?
- Block length?
- Decoding performance?

- Rate and synchronisation performance can be computed (or approximated) analytically.
- Calibrate all techniques to equal rate, equal synchronisation probability and equal block length and compare on the basis of decoding performance.

## Bounds for preamble-based synchronisation

- Threshold synchroniser, metric

$$\mu(i) = -\frac{1}{L}\sum_{k=1}^{L} \log \mathrm{P}_{Y|X}(y_{i+k-1}|p_k)$$

- finds the set $\mathcal{D} = \{i : \mu(i) \leq \theta\}$,

$$\begin{cases} \text{if } |\mathcal{D}| = 1, \text{ pick } i \in \mathcal{D} \\ \text{otherwise declare a failure.} \end{cases}$$

- Chernoff bound for the probability of successful synchronisation:

$$\begin{aligned} \mathrm{P}_{\text{success}} &= \Pr\{\mu(1) \leq \theta \text{ AND } \mu(i) > \theta, \forall i \neq 1\} \\ &\geq \left(1 - e^{-L\mathcal{E}_1(\theta,\hat{\gamma}_1)}\right)\left(1 - e^{L\mathcal{E}_2(\theta,\hat{\gamma}_2)}\right)^{N-1} \end{aligned}$$

where $\mathcal{E}_1(\theta,\hat{\gamma}_1) = \hat{\gamma}_1\theta - \log\sum_y (\mathrm{P}_{Y|X}(y|x))^{1-\hat{\gamma}_1}$ and $\mathcal{E}_2(\theta,\hat{\gamma}_2) = \hat{\gamma}_2\theta + \log\sum_y \mathrm{P}_Y(y)(\mathrm{P}_{Y|X}(y|x))^{\hat{\gamma}_2}$.

# Exact expressions for the BSC

- Threshold synchroniser:

$$P_{\text{success}} = \left( \sum_{k=0}^{k_\theta} \binom{L}{k} p^k (1-p)^{L-k} \right) \left( 1 - \frac{1}{2^L} \sum_{k=0}^{k_\theta} \binom{L}{k} \right)^{N-1}$$

where $k_\theta = \max_k \{ p^k (1-p)^{L-k} > e^{-L\theta} \}$

- ML synchroniser:

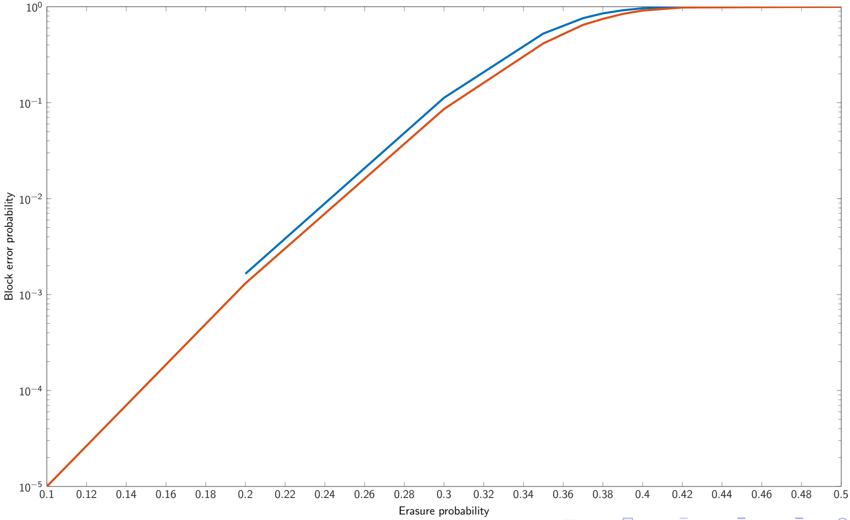
$$P^{\text{ML}}_{\text{success}} = \sum_{k=0}^{L} \binom{L}{k} p^k (1-p)^{L-k} \left( 1 - \frac{1}{2^L} \sum_{j=0}^{k-1} \binom{L}{j} \right)^{N-1}$$

# More about bounds

- We derived similar bounds and exact expressions for permutation constraints based synchronisers. These bounds do not reduce to single-variable expressions and require sums over $q!$ terms.
- Exact expressions for both regimes can only be computed for small values of $L$
- For example, for $N = 400$, $L = 40$, BSC $p = 0.1$, $\theta = 0.6$, the Chernoff bound for preamble-based synchronisation yields $\mathrm{P}_{\text{success}} \geq 0.4767$ while the exact expression gives $\mathrm{P}_{\text{success}} = 0.8688$
- Although the bounds may be tight asymptotically or even at finite lengths when represented in terms of rate, we need to calibrate for equal probability of synchronisation to get a fair comparison and the bounds seem insufficiently tight for this purpose

# Early performance comparisons

- GF(4), $N = 1000$, $L = 8$, $m = 5$



# Conclusion

- I am still not sure I'm 100% in love with synchronisation
- But it's beginning to grow on me