CHANNEL MEASUREMENT

Channel measurement doesn't help for single bit transmission in flat Rayleigh fading.

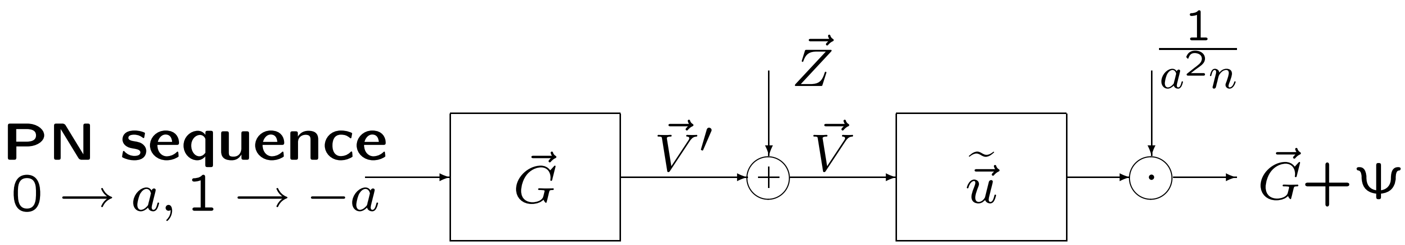
It helps (as we soon see) in detection with multi-tap fading, multiple frequencies, multiple antennas, etc. (i.e., in the presence of diversity).

It helps the transmitter (if known there) to adjust power and rate.

Pseudonoise (PN) PROBING SIGNALS

A PN sequence $\vec{u}$ is a binary sequence that appears to have iid components.

A maximal-length binary shift register of k stages generates all 2^k-1 binary non-zero k-tuples and is periodic with length $2^k - 1$.



$\vec{u}$ is $\approx$ orthogonal to each shift of $\vec{u}$ so

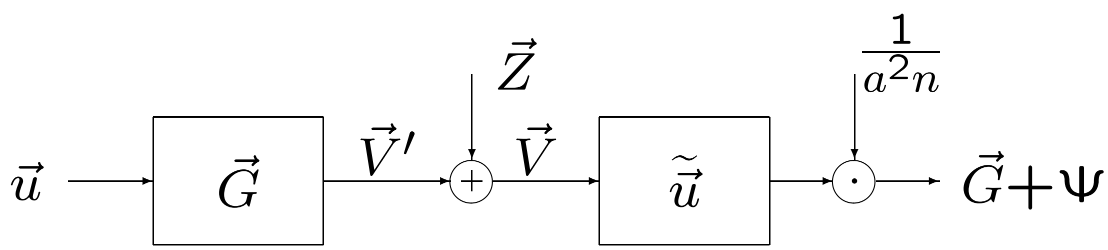
$$\sum_{m=1}^{n} u_m u_{m+k}^* \approx \begin{cases} a^2 n & ; \quad k = 0 \\ 0 & ; \quad k \neq 0 \end{cases} \qquad = \qquad a^2 n \delta_k$$

If $\tilde{\vec{u}}$ is matched filter to $\vec{u}$, then $\vec{u} * \tilde{\vec{u}} = a^2 n \delta_j$.

If $\vec{u} * \widetilde{\vec{u}} = a^2 n \delta_j$**, then**

$$\vec{V}' * \widetilde{\vec{u}} = (\vec{u} * \vec{G}) * \widetilde{\vec{u}} = (\vec{u} * \widetilde{\vec{u}} * \vec{G}) = a^2 n \vec{G}$$

A PN sequence has the same effect as using a single input (of n **times the power) surrounded by zeros.**



The response at time m **of** $\tilde{u}$ **to** $\vec{Z}$ **is the sum of** n **iid complex rv's** $\mathcal{N}_c(0, a^2 N_0 W)$**. .**

The sum has variance $a^2 n N_0 W$**. After scaling by** $1/(a^2 n)$**,** $\mathsf{E}[|\Psi_k|^2] = \frac{N_0 W}{a^2 n}$**.**

RAKE RECEIVER

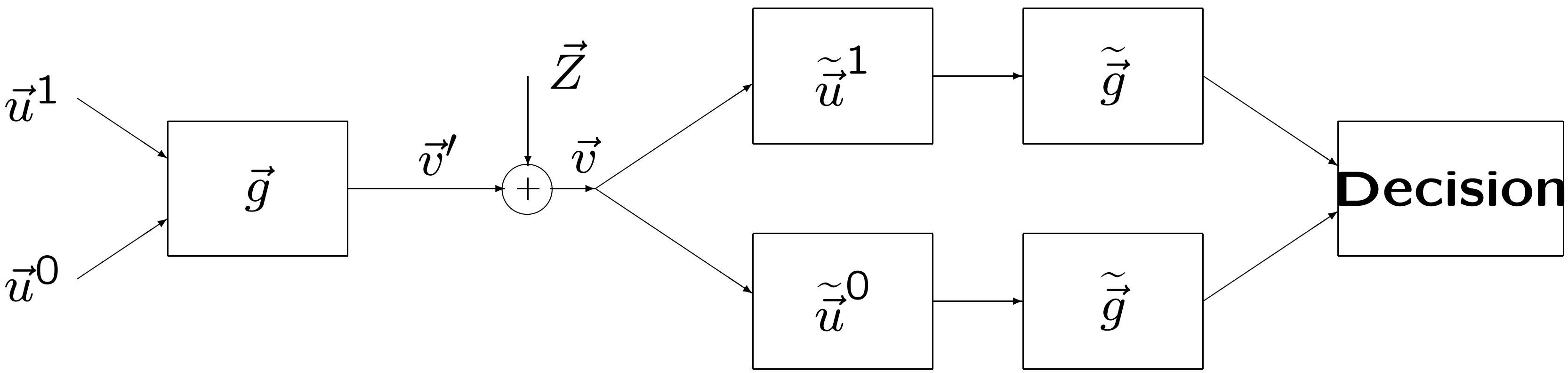
The idea here is to measure the channel and make decisions at the same time.

Assume a binary input, $H{=}0 \rightarrow \vec{u}^0$ **and** $H{=}1 \rightarrow \vec{u}^1$

With a known channel $\vec{g}$**, the ML decision is based on pre-noise inputs** $\vec{u}^0 * \vec{g}$ **and** $\vec{u}^1 * \vec{g}$**.**

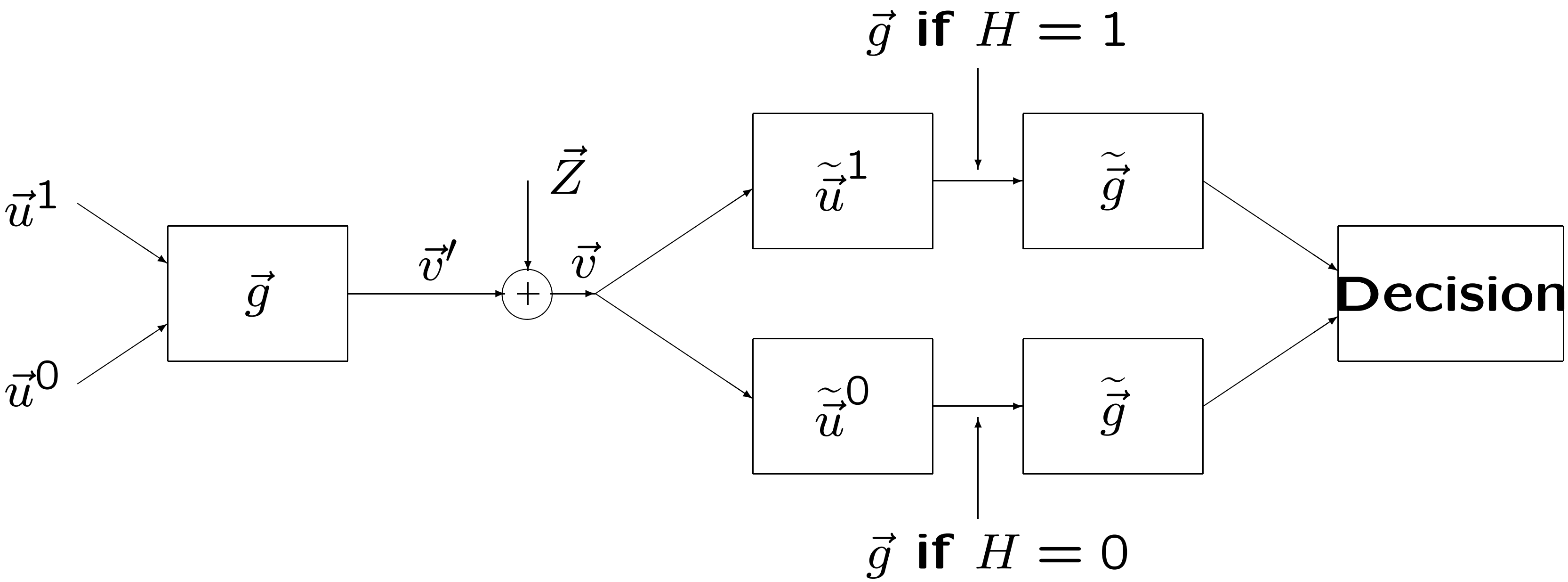
$$\Re(\langle \vec{v}, \vec{u}^0 * \vec{g} \rangle) \;\underset{\tilde{H}=1}{\overset{\tilde{H}=0}{\gtrless}}\; \Re(\langle \vec{v}, \vec{u}^1 * \vec{g} \rangle).$$

We can detect using filters matched to $\vec{u}^0 * \vec{g}$ **and** $\vec{u}^1 * \vec{g}$



Note the similarity of this to the block diagram for measuring the channel.

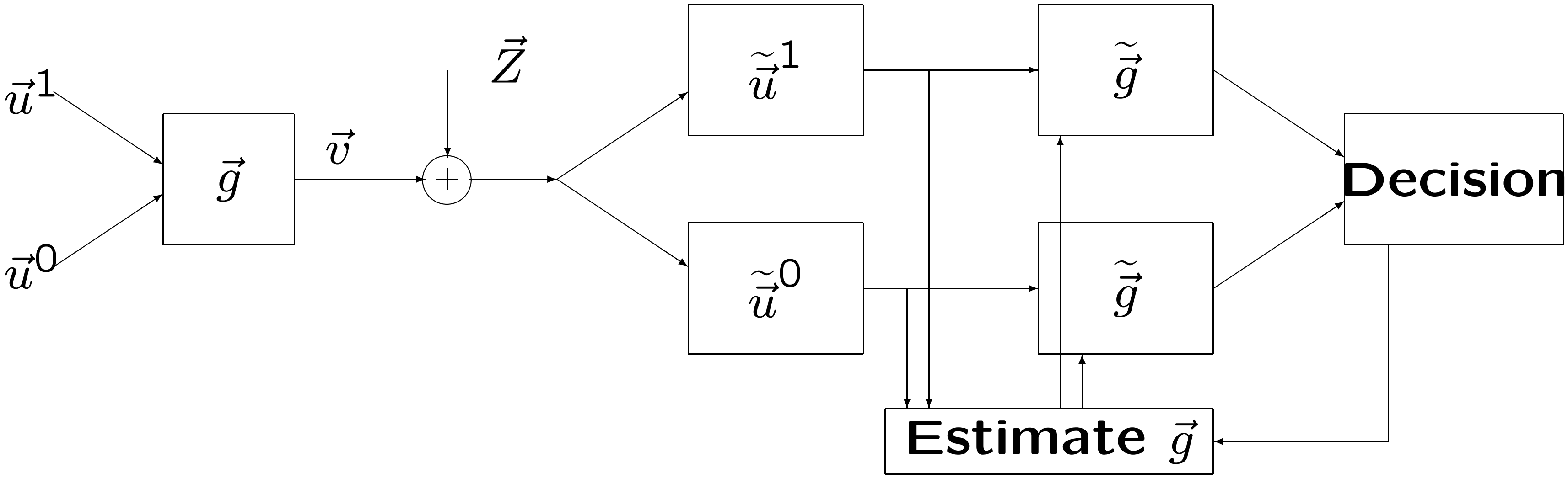
If the inputs are PN sequences (which are often used for spread spectrum systems), then <u>if the correct decision</u> can be made, the output of the corresponding arm contains a measurement of $\vec{g}$.



$\vec{u}^1$ and $\vec{u}^0$ are non-zero from time 1 to n. $\vec{v}'$ is non-zero from 1 to $n+k_0-1$.

$\tilde{\vec{u}}^1$ and $\tilde{\vec{u}}^0$ are non-zero from $-n$ to -1 (receiver time).

If $H = 1$ or $H = 0$, then $\vec{g}$ plus noise appears from time 0 to $k_0 - 1$ where shown. Decision is made at time 0, receiver time.



If $\widehat{H} = 0$, then a noisy version of $\vec{g}$ probably exists at the output of the matched filter $\vec{u}^0$. That estimate of $\vec{g}$ is used to update the matched filters $\tilde{\vec{g}}$.

If $\mathcal{T}_c$ is large enough, the decision updates can provide good estimates.

Suppose there is only one Rayleigh fading tap in the discrete-time model.

Suppose the estimation works perfectly and $\vec{g}$ is always known. Then the probability of error is the coherent error probability $Q(\sqrt{E_b/N_0})$ for orthogonal signals and $E_b = a^2 n|g|^2/W$.

This is smaller than incoherent $\text{Pr}(e) = \frac{1}{2}\text{exp}\{-E_b/(2N_0)\}$.

Averaging over G, incoherent result is $\frac{1}{2+\overline{E}_b/N_0}$ and coherent result is almost the same.

Measurement helps in allowing antipodal transmission.

Case study: CDMA, IS95

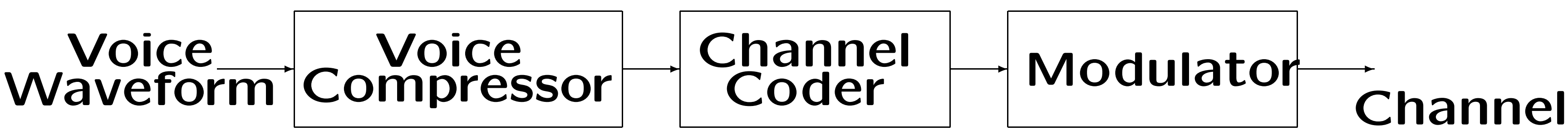
Uses frequency band from 800 to 900 MH.

800-850 used for reverse channel (cell phones $\rightarrow$ base).

850-900 for forward channel.

Need for considerable separation (45mH) to avoid self noise.

Individual subbands of 1.25 MH.



Voice compressor

Input voice segmented into 20ms segments.

Each segment compressed independently into 172 bits. This encodes voice into 8.6 kbps.

Old fashioned voice digitization uses 4 kH. nominal BW (8 ksps), 8 bit quantization, 64 kbps

12 parity checks per segment for error detection.

Another 8 zeros per segment are added to terminate subsequent convolutional code.

Result is 192 bits per 20 ms (9600 bps).

All timing in encoder and decoder keyed to 9600 bps and 20 ms. intervals.

Source/channel separation violated by 20ms.

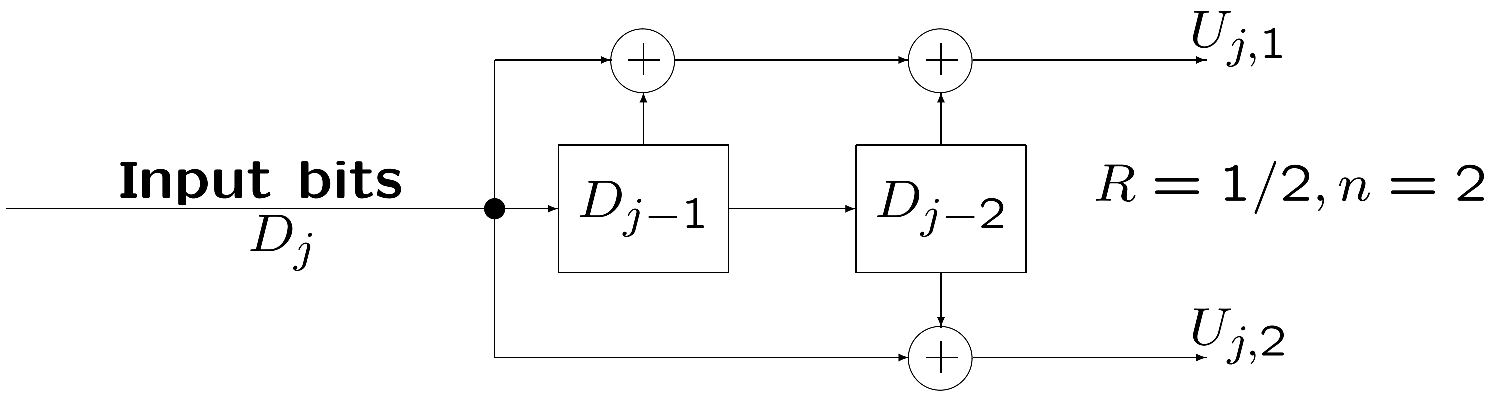
Caused by the strange nature of voice.

Small delay is critical for voice. Larger delay violates conventions for human conversation.

Voice has many statistical constraints lasting far longer than 20 ms, but compression can't use them.

Channel coding could use longer codewords effectively, but can't in voice systems.

A convolutional encoder, $R = 1/3$, constraint length $n = 8$ follows the source compression.

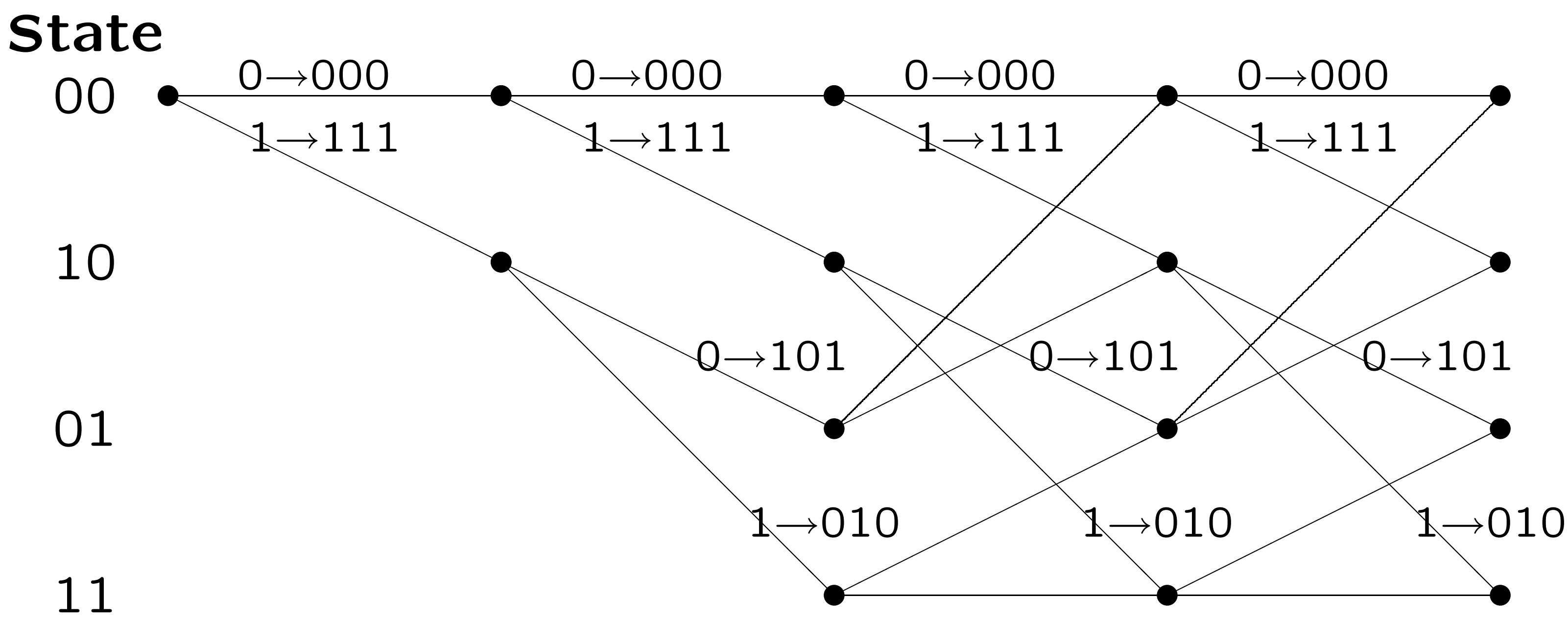


$$
\begin{aligned}
U_{j,1} &= D_j \oplus D_{j-1} \oplus D_{j-2} \\
U_{j,2} &= D_j \qquad\qquad\ \oplus D_{j-2}
\end{aligned}
$$

It needs n bits at end of block to return to state 0.

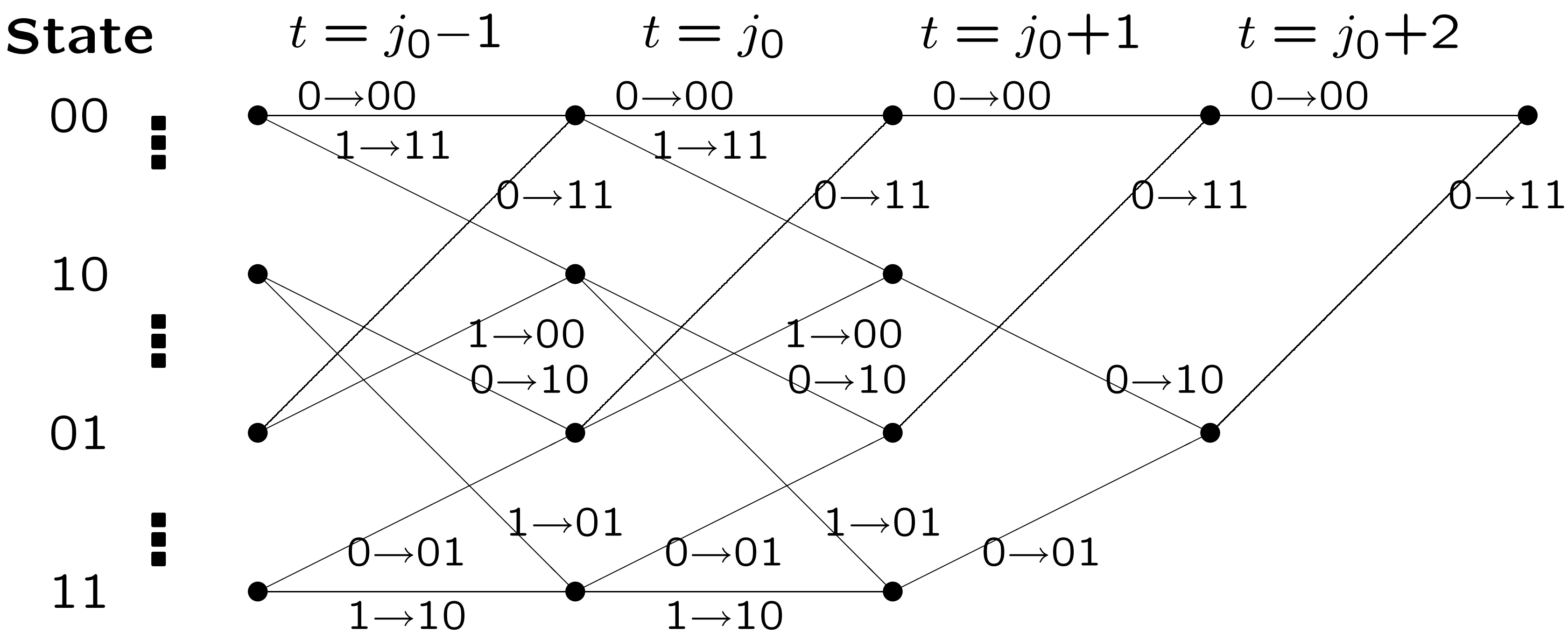
Viterbi algorithm used for decoding; complexity $\sim 2^n$.

Trellis for Viterbi decoder



Conventional soft decoder: use log likelihood ratio (against a reference) for each input bit. Sum of LLR's is LLR for a sequence.

The code is terminated with n bits.



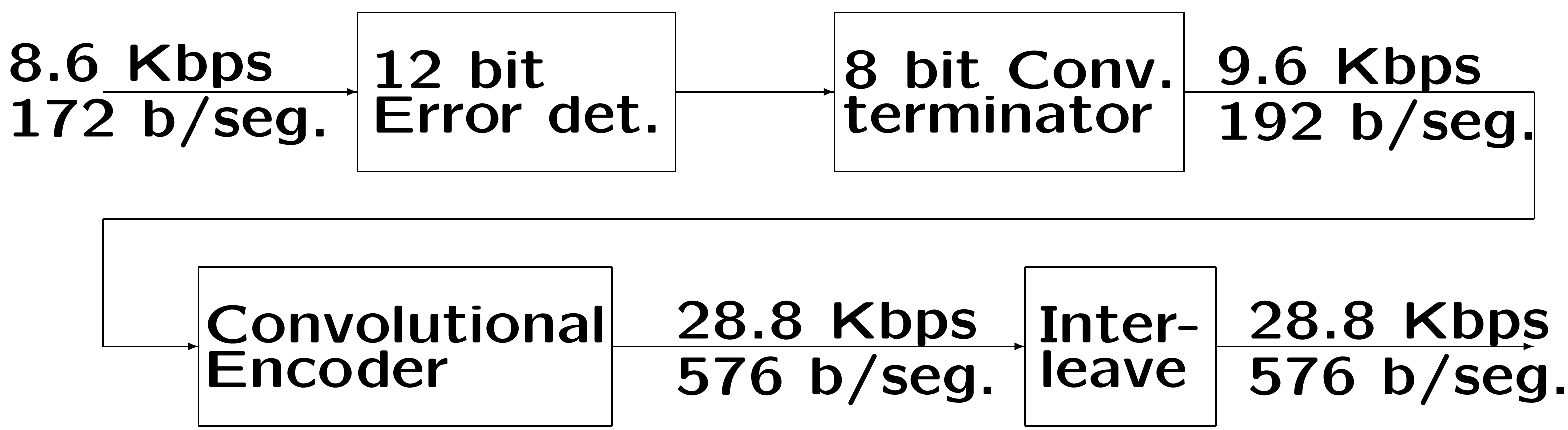
Codeword length is $3(j_0 + n)$. The $n = 8$ termination bits was added to input.

Voice compression and Channel coding

20 ms voice segments compressed to 172 bits.

20 bit overhead, then rate 1/3 code.

576 bit block output then interleaved



Modulation by Hadamard matrix

Interleaver output is segmented into 6 bit blocks.

Map 6 bit blocks to 64 bit orthogonal code-words.

0	0
0	1

$b = 1$

0	0	0	0
0	1	0	1
0	0	1	1
0	1	1	0

$b = 2$

0000	0000
0101	0101
0011	0011
0110	0110
0000	1111
0101	1010
0011	1100
0110	1001

$b = 3$

Generate H_{b+1} from H_b: put H_b at top left, top right, lower left, and put complement $\overline{H}_b$ at lower right.

Modulation

Segment interleaver output into 6 bit segments.

Map 6 bit $\rightarrow$ 64 bits rows of H_6.

Map selected row into antipodal signals.

This gives us orthogonal code of length 64.

28.8 kbps $\rightarrow$ (28.8)(64)/6 = 307.2 kbps

Receiver uses these codewords in a rake receiver.

Rake has 64 branches rather than 2.

Incoherent decisions are used.

For orthogonal code, incoherent reception in WGN

$$\Pr(e) \leq \frac{63}{2} \exp\left[\frac{-E_s}{2N_0}\right]$$

$E_s = 6E_b$, **so** $\Pr(e) \leq \frac{63}{2} e^{-3E_b/N_0}$.

Effect of fading is to change E_b **over time.**

Because of multitap diversity, E_b **changes less than for Rayleigh distribution.**

Typical case: E_b **large, little penalty for incoherent.**

Output of orthogonal decoder could in principle produce 63 LLR's. Want only one LLR for each of 6 bits.

Want $\log\left(\frac{\text{Pr(correct)}}{\text{Pr(error)}}\right)$

Visualize comparing two codewords 0000000 and 1111111 for hard decisions with error probability p.

If errors are independent, $P_e \approx 35p^4$. If perfectly correlated, $P_e = p >> 35p^4$ for p small.

Lesson: Highly dependent errors screw up decoding.

Solution: Interleaving.

Rest of modulator

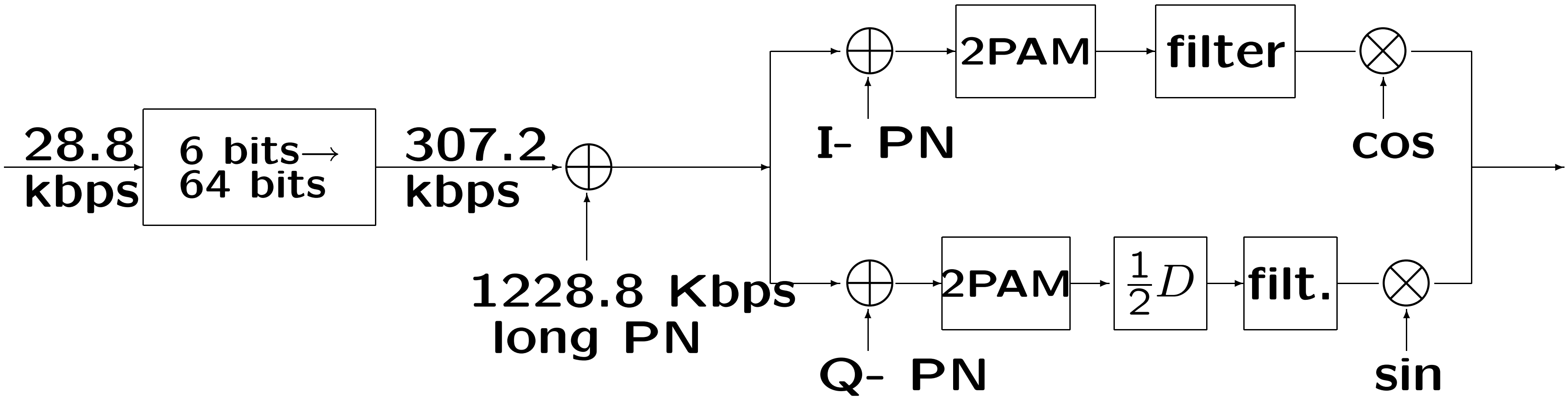
The sequence of Walsh functions from encoder are multiplied by PN sequence at 4 times 307.2 kbps (1.2288mbps)

This is the output of a 42 bit linear shift register.

The period is $2^{42} - 1$, with this many possible starting states.

Each cell phone starts with a different state in this PN sequence. Only difference between different cell phones.

This Mod-2 addition on Hadamard output doesn't effect orthogonality.



The I-PN and Q-PN are used to make the inphase and quadrature components look different.

How can this be received without a phase lock loop?

Why is incoherent reception necessary then?