Principles of Digital Communication

May 5, 2021

Reading was

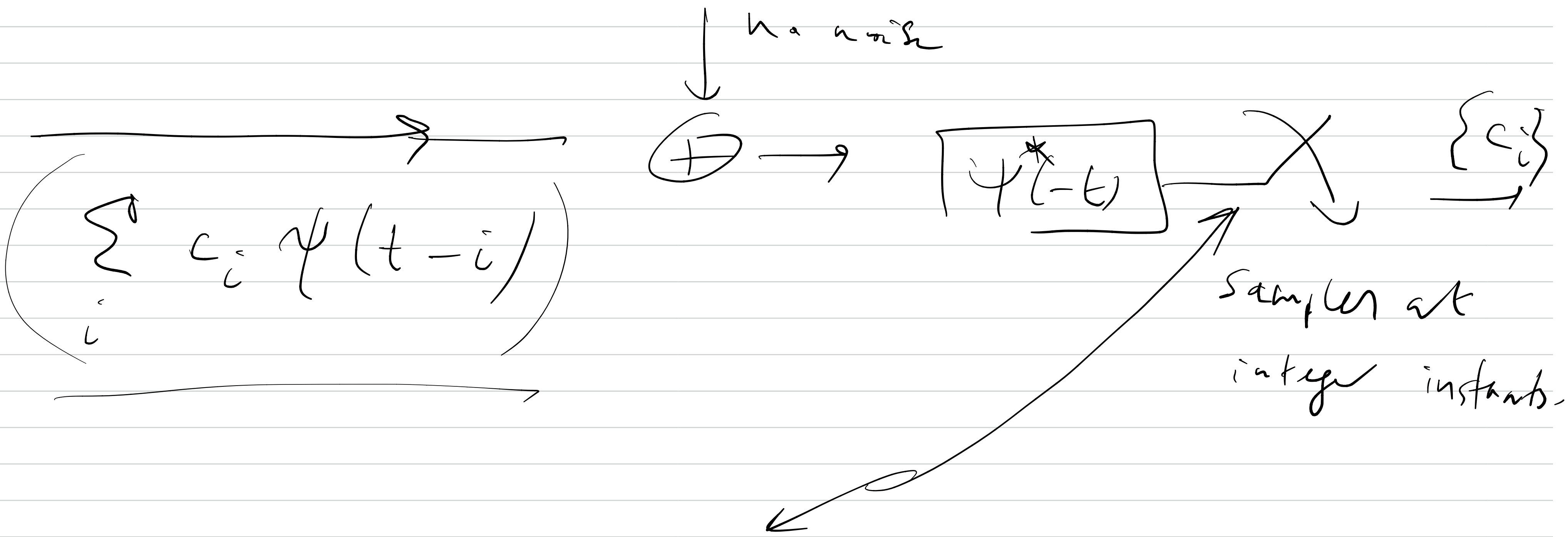
- Root-raised cosine pulses
- Eye diagrams

} Ch 5

- Convolutional Codes

} Ch 6 (1, 2, 3).

RRC: motivation:



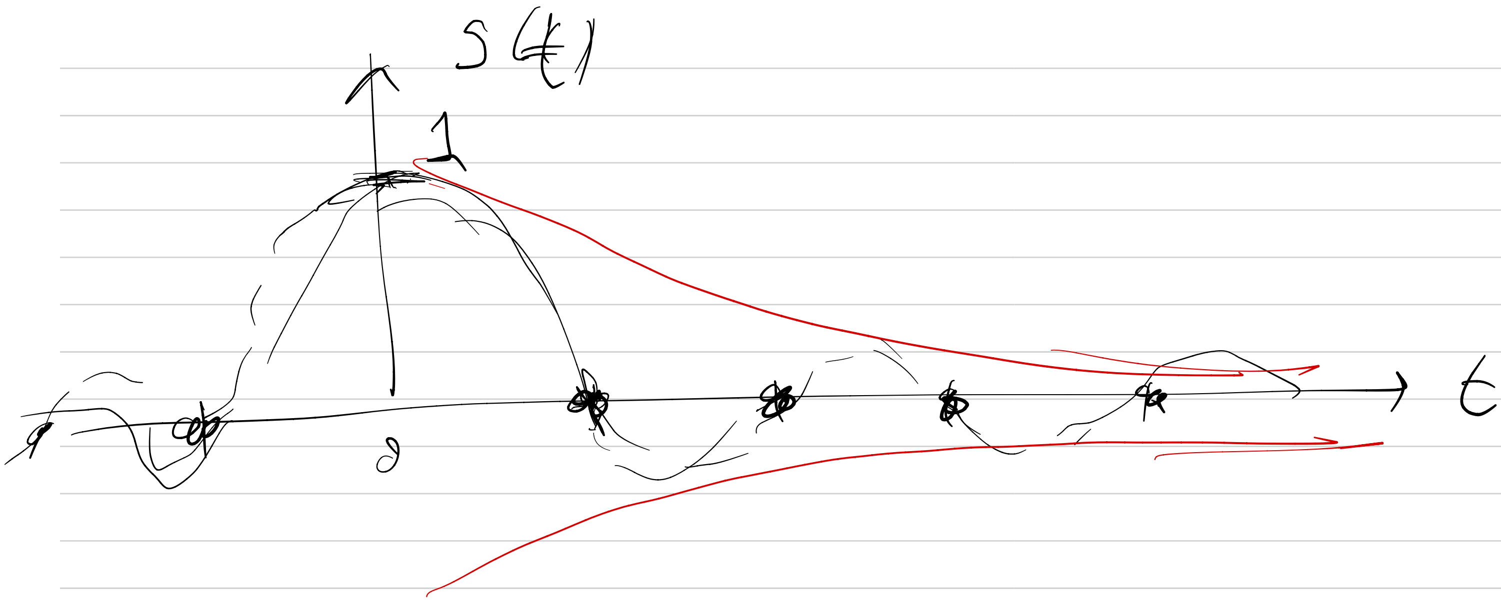
$\sum_i c_i g(t-i)$ $\quad$ g = convolution of $\psi(t)$ with $\psi^*(-t)$

$$g(t) = \int \psi(\tau)\, \cancel{g(t-\tau)}\, d\tau = \int \psi(\tau)\, \psi^*(\tau - t)\, dt$$

$$\psi^*(\tau - t)$$

$$\underline{g(t)} \leq \sqrt{\int |\psi(t)|^2 dt} \sqrt{\int |\psi^*(t)|^2 dt}$$

$$= \int |\psi(t)|^2 dt = g(0).$$



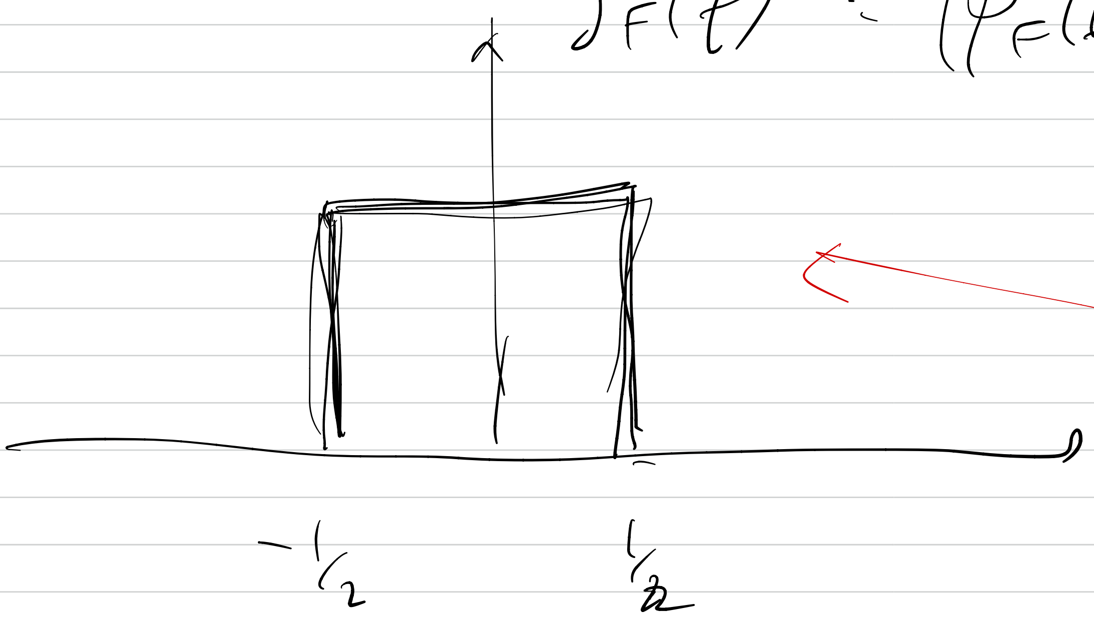
If the filter output is sampled not at integer times but at slightly wrong times, e.g. $i+\delta$ then the i'th sample will be

$$\sum_j c_j\, g(t-j)\Big|_{t=i+\delta} = c_i\, g(\delta) + \sum_{j: j\neq i} c_j\, g(i-j+\delta)$$

A fast decaying (in time) $g(\cdot)$ will ensure that is small, so we will have some protection against timing errors.

Our difficulty is:

$g_F(f) = |\psi_F(f)|^2$ should satisfy Nyquist.



the minimal BW $g_F(\cdot)$

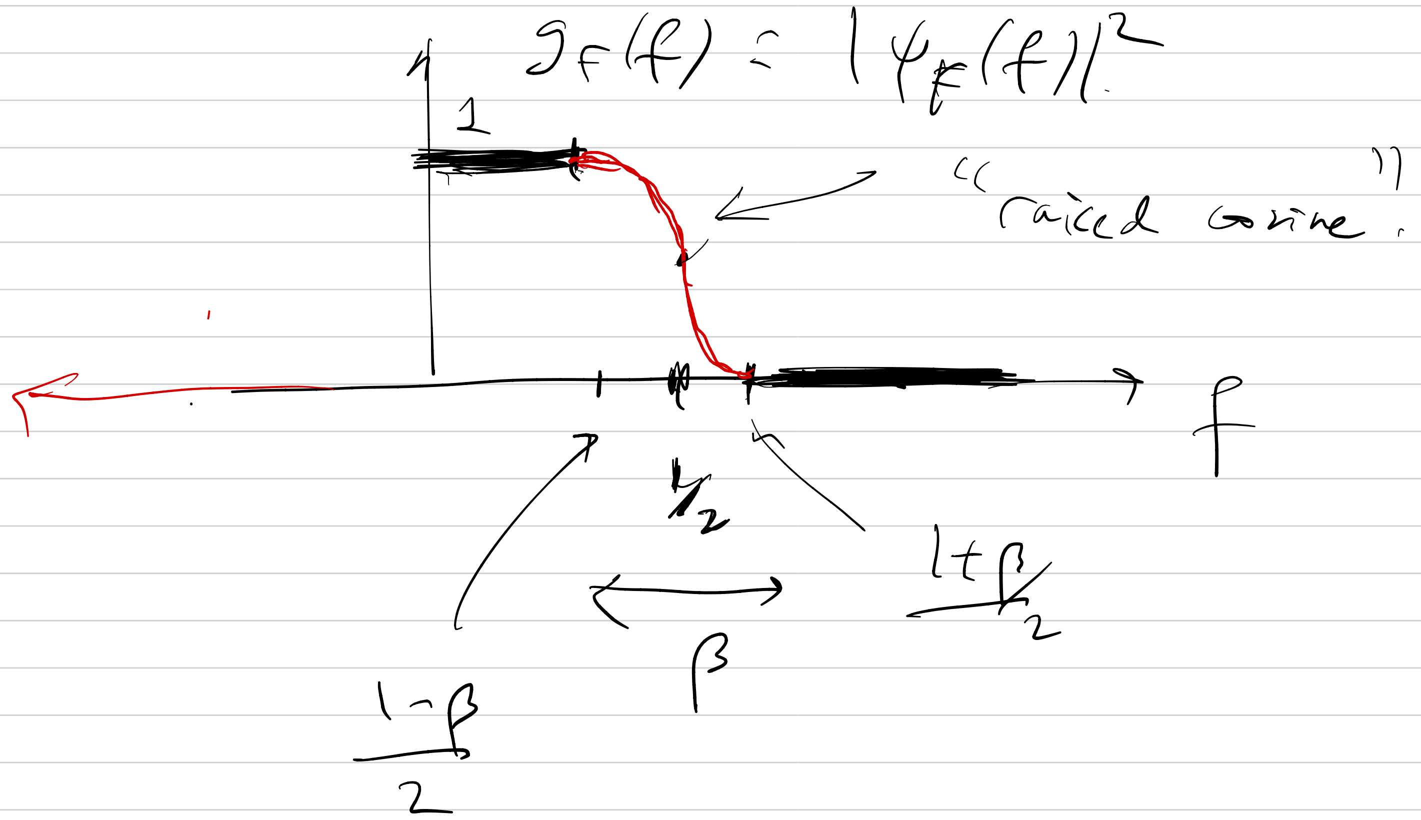
$\to g(t) = \mathrm{sinc}(t)$

decays slowly. We saw last week that how fast $g(t)$ decays

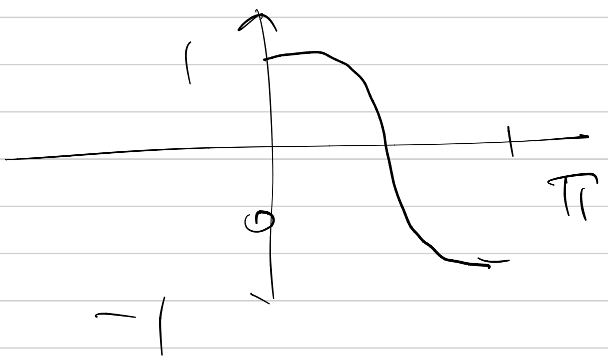
$\Longleftrightarrow$ how smooth $g_F(f)$ is.

This leads to an ad-hoc choice (raised cosine)

$g_F(f) = |\psi_F(f)|^2$



β: excess % bandwidth



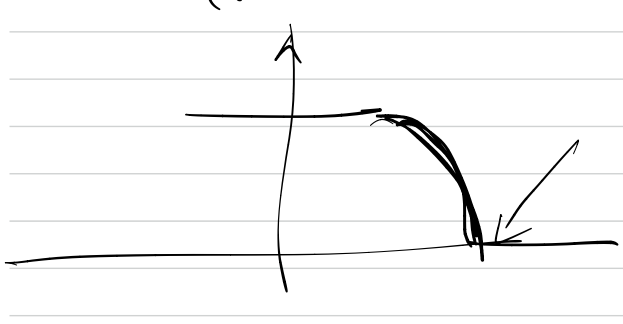
Since $\frac{1+\cos x}{2} = \cos^2 \frac{x}{2}$

So we have a simple expression for $\psi_F(f)$

So we expect

$g(t)$ to decay like $\left(\frac{1}{t^3}\right)$

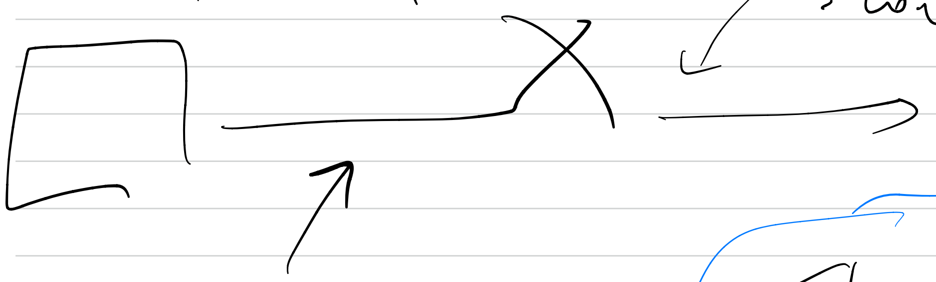
$\psi_F(f)$



so $\psi(t)$ decays like $1/t^2$.

Eye diagrams: A way to judge how immune a system with a given $\psi(t)$ is against timing errors (samples taken not exactly at integers) & noise.

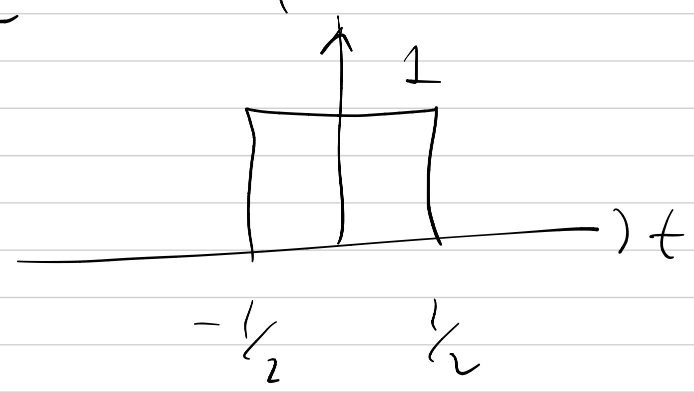
matched filtr



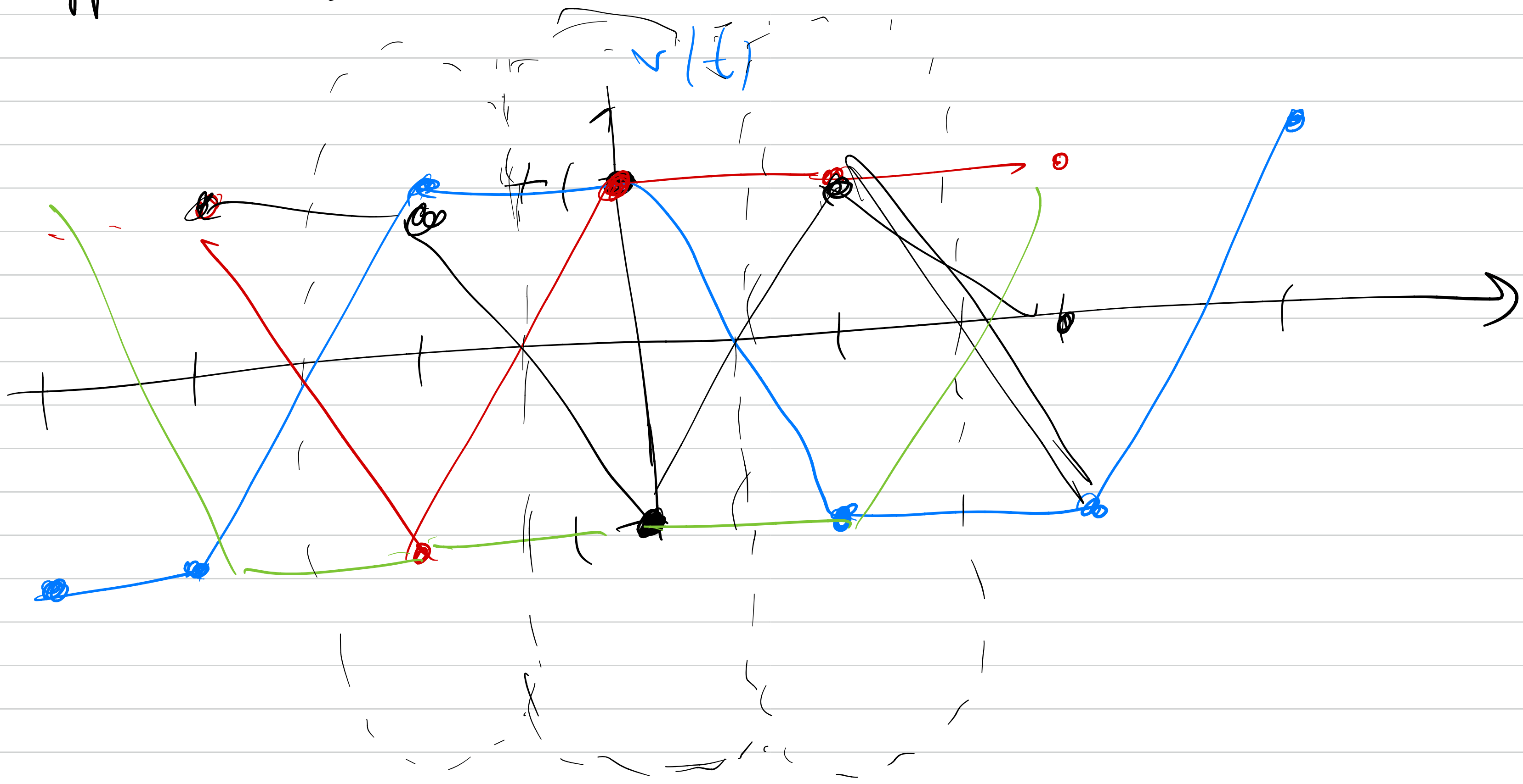
should the samples at integer times.

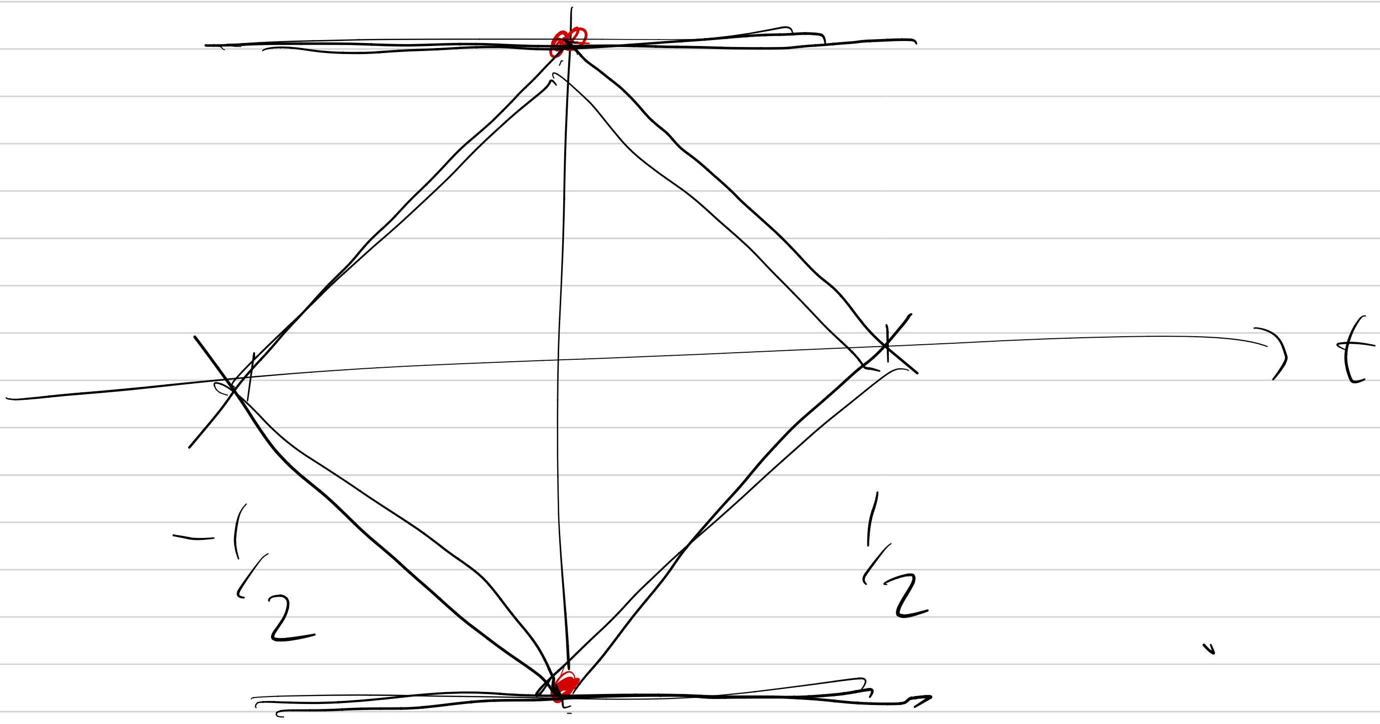
without noise $= \boxed{\sum_j{}' c_j\, g(t-j)} = v(t)$

Ex: $\psi(t)$ $g(t)$



suppose c_i's are ± 1's.



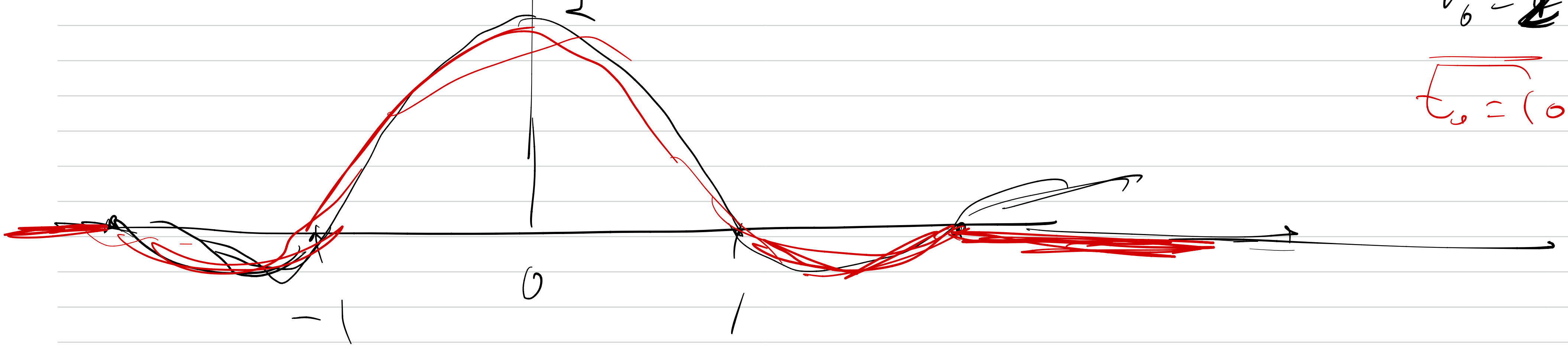


The book consider

$$\psi(t) = \mathrm{RRC}_\beta(t)\,\mathbb{1}\{|t| < t_0\}$$

$t_0 = 2$

$t_0 = 10$



And plots the Eye-diagrams p 174.

β large $\rightarrow$ "more open eye"

small $\rightarrow$ "more closed eye"

t_0 small $\rightarrow$ "

t_0 large $\rightarrow$

Graded HW: already on Moodle

Due: 16th

"Group solutions"

Chapt 6.

recall that we saw

k = # of bits in our message $k = \log_2 m$

n : # of dimensions (Duration × BW)

k/n : bit rate (bits/sec/Hz)

efficiency per dimension

$\mathcal{E}_b$ = energy / bit. $\|c_i\|^2 / k$

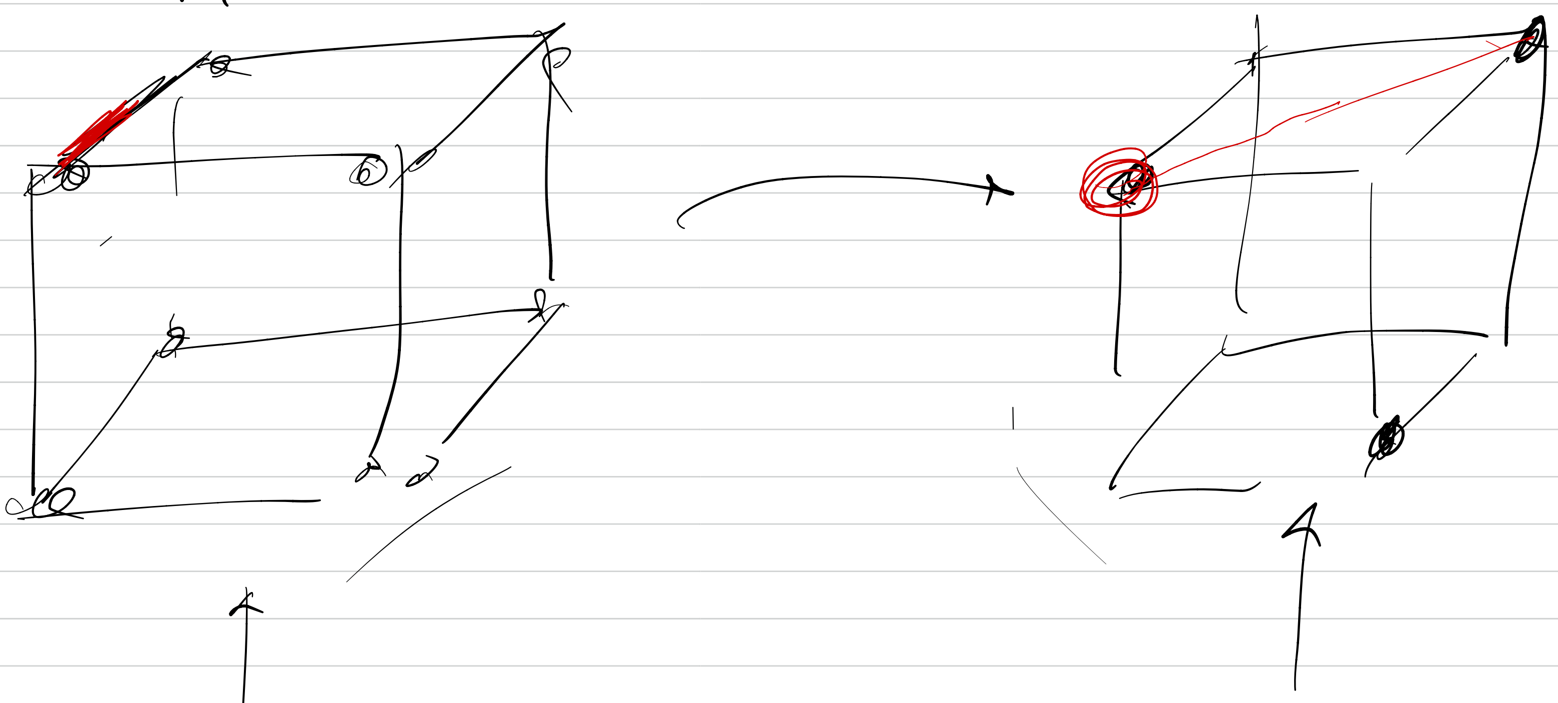
$b_1 \ldots b_k$ to be sent ($\{+1, -1\}$ each of them).

4.4.7 $w(t; b_1 \ldots b_k) = \sum_i b_i \psi(t-i)$

has $k/n = 1$

for large k this has high P(message error).

To rectify this:



bit-by-bit method using all 2^k corners of the hypercube as signal points.

use a fraction of the vertices 2^k corners of a $(2k)$ dim cube (2^{2k} corners).

We hope to gain reliability by reducing bit-rate.

How to do this:

Crazy idea: pick 2^k points among $2^n = 2^{2k}$ "at random"!! If we do so, there is an excellent chance that we end up with a "good" selection. ($P(\text{error})$ will not be very different than the "best" choice of the 2^k points).

Ex: $k=20$ we have $2^{20} \simeq 10^6$ points.

RCUR: $\arg\min_{1 \le i \le 2^{20}} \|y - c_i\|$

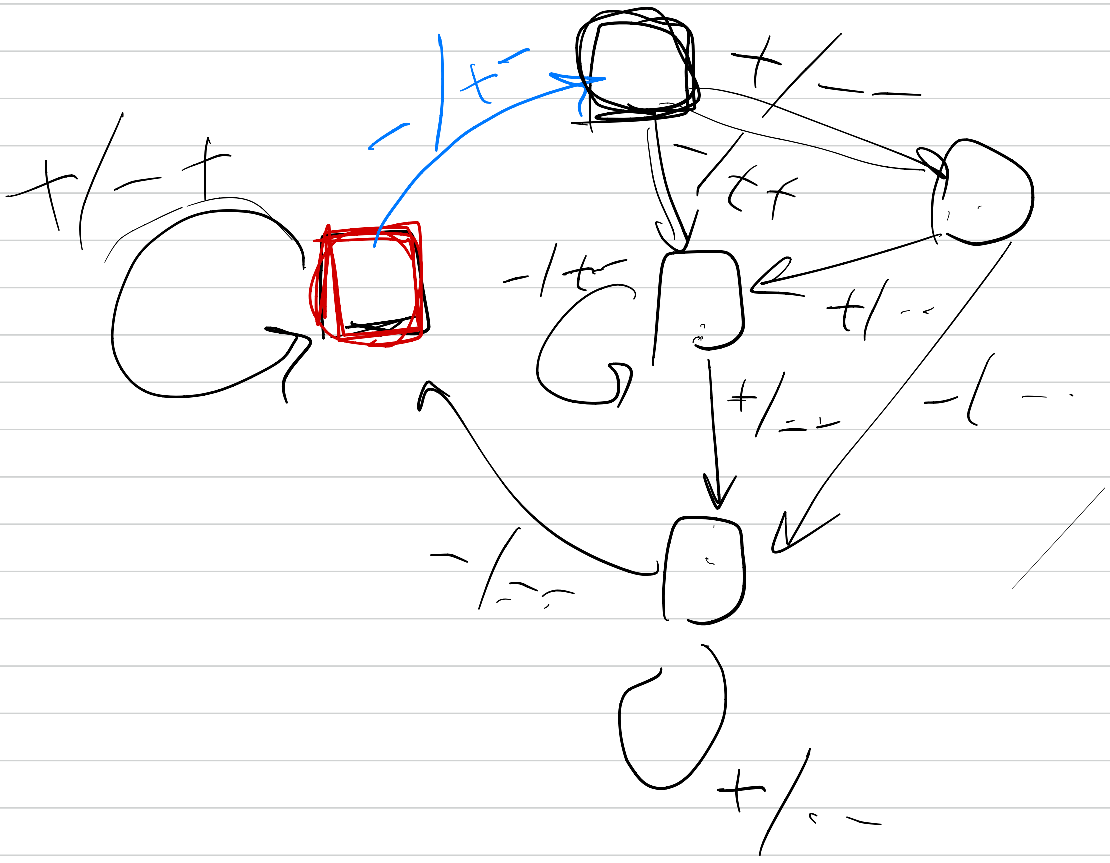
10^6 operations to decode 20 bits.

if $k = 100 \Rightarrow$ deep trouble..

Now what?

Encoding by Finite state Machines.

Ex:



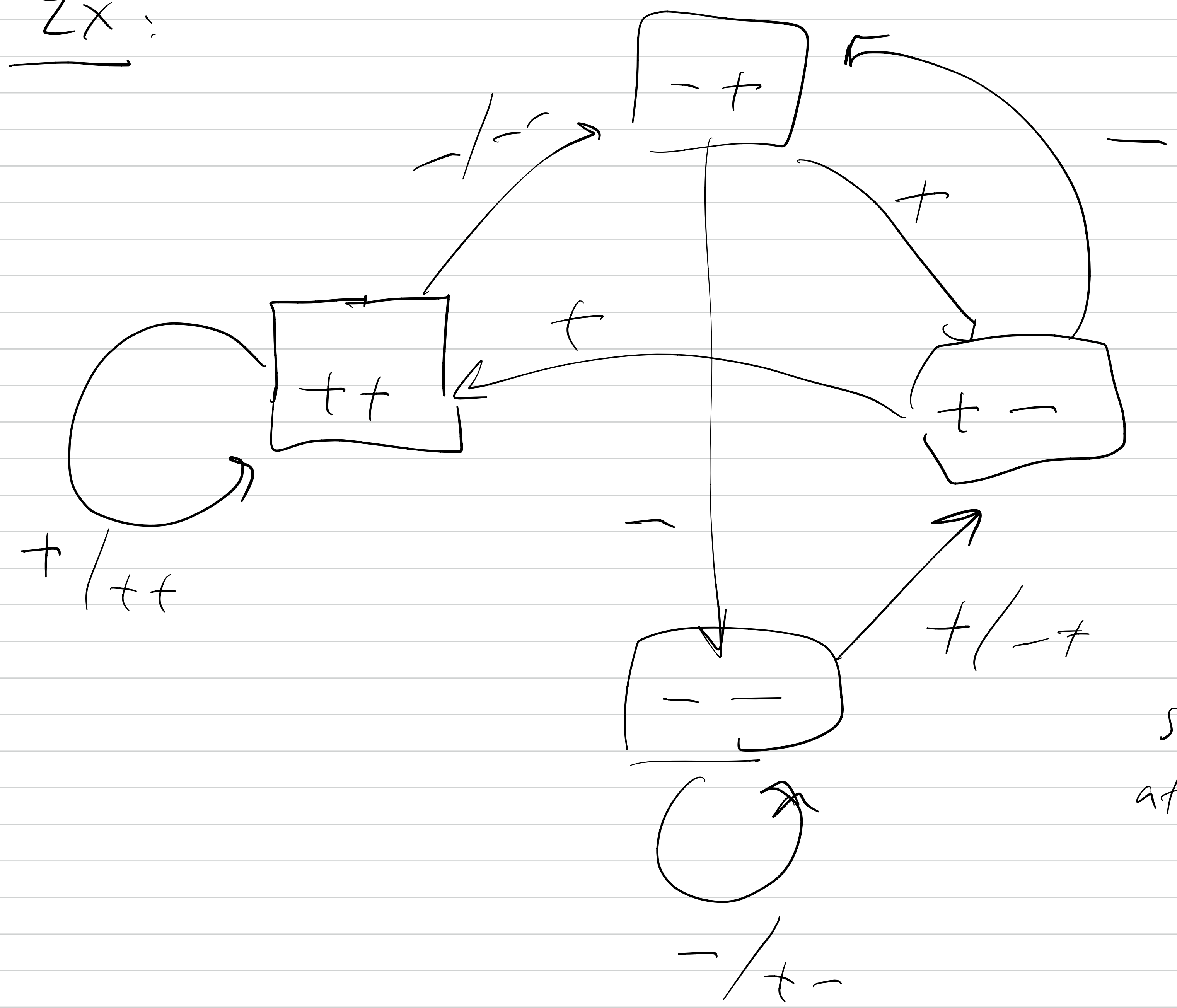
$b_1 b_2 b_3 b_4 = - - + -$

$\downarrow$

$+ - + + - - - - -$

Such a machine maps k bit sequences to $n = 2k$ codewords.

Ex:



state: $b_{j-1} b_{j-2}$ at time j

input $= b_j$

$$\text{output} = \begin{bmatrix} b_j b_{j-2} \\ b_j b_{j-1} b_{j-2} \end{bmatrix}$$

- The description complexity of the encoder does not change with k (compare with "crazy-method" encoder description $= 2^k$)

- The encoding complexity is $\sim k$

How about the receiver?

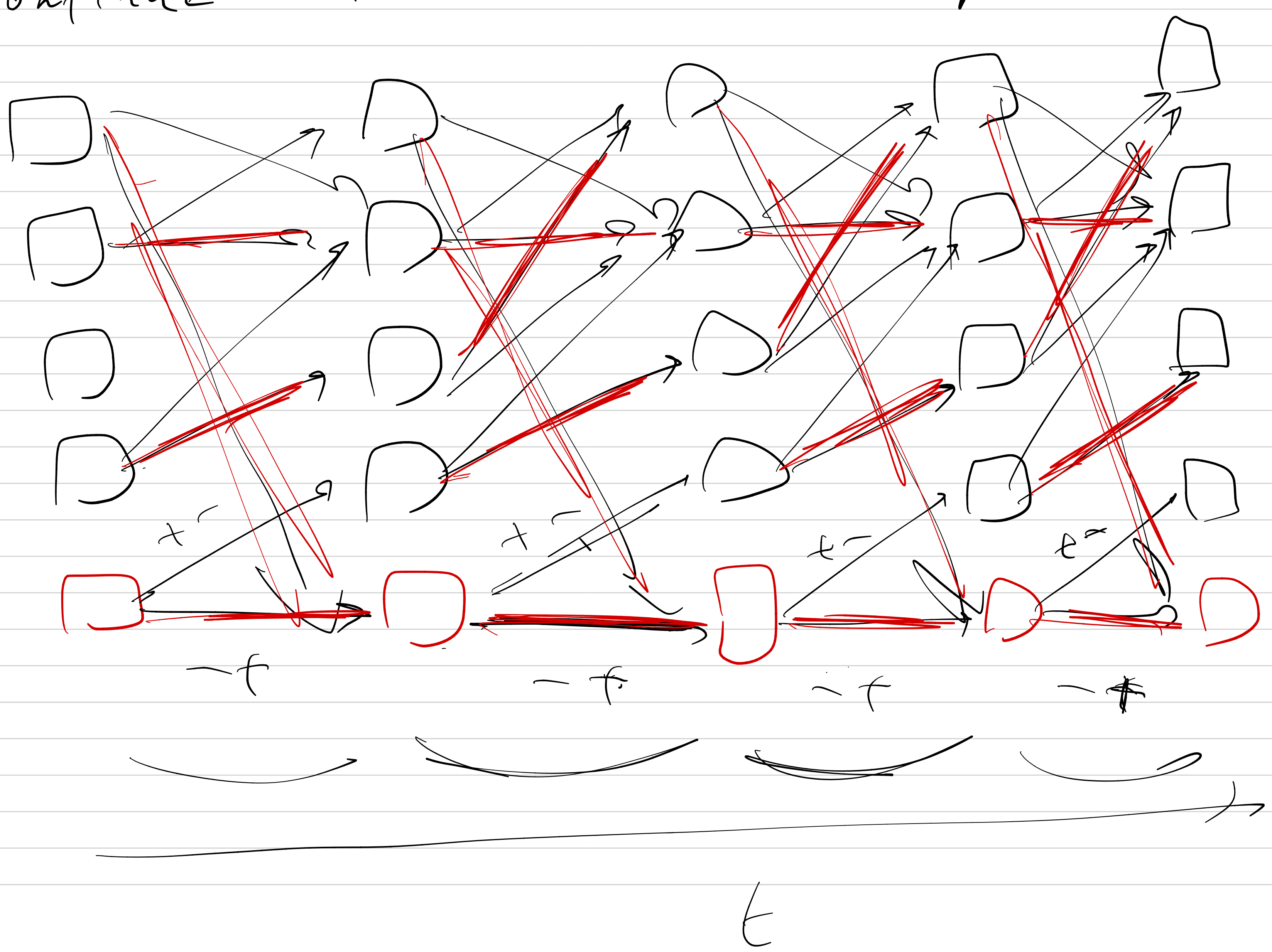
we need to find

$$\operatorname*{argmin}_{b_1 \ldots b_k} \|(y_1 \ldots y_n) - C(b_1 \ldots b_k)\|$$

the encoder output when $b_1 \ldots b_k$ is the input.

This still looks unpractical: there are 2^k things to compare.

Continue with the 1st example:



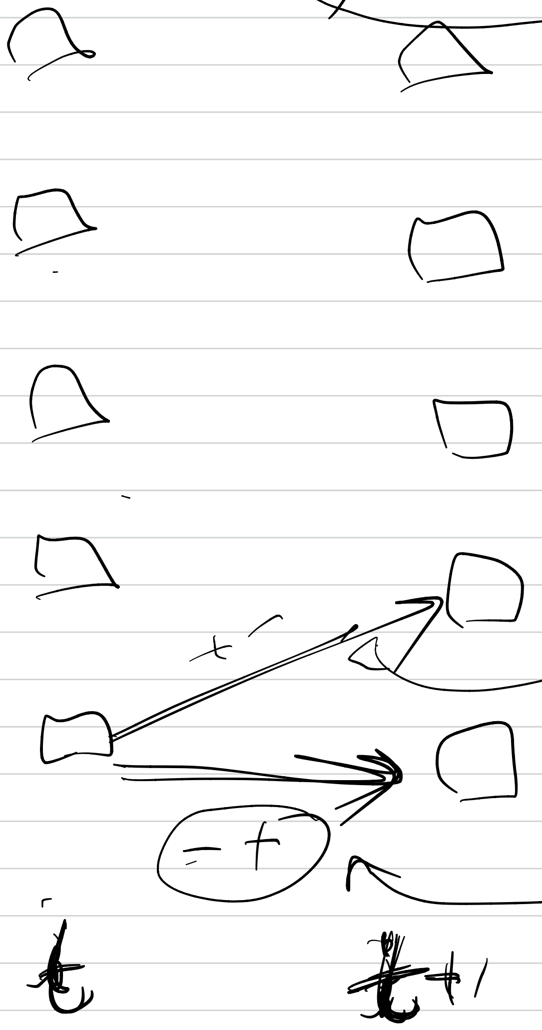
receiver task:

$$\underset{b_1 .. b_k}{\arg\min} \; \underbrace{\|y - c(b_1 .. b_k)\|^2}$$

$$= \|y\|^2 + \|c(b_1 .. b_k)\|^2 - 2\langle y, c(b_1 .. b_k)\rangle$$

does not depend on $b_1 .. b_k$

$$= \underset{b_1 .. b_k}{\arg\max} \; \underbrace{\langle y, c(b_1 .. b_k)\rangle}_{\sum_{j=1}^{n} y_j \, c(b_1 .. b_k)_j}$$

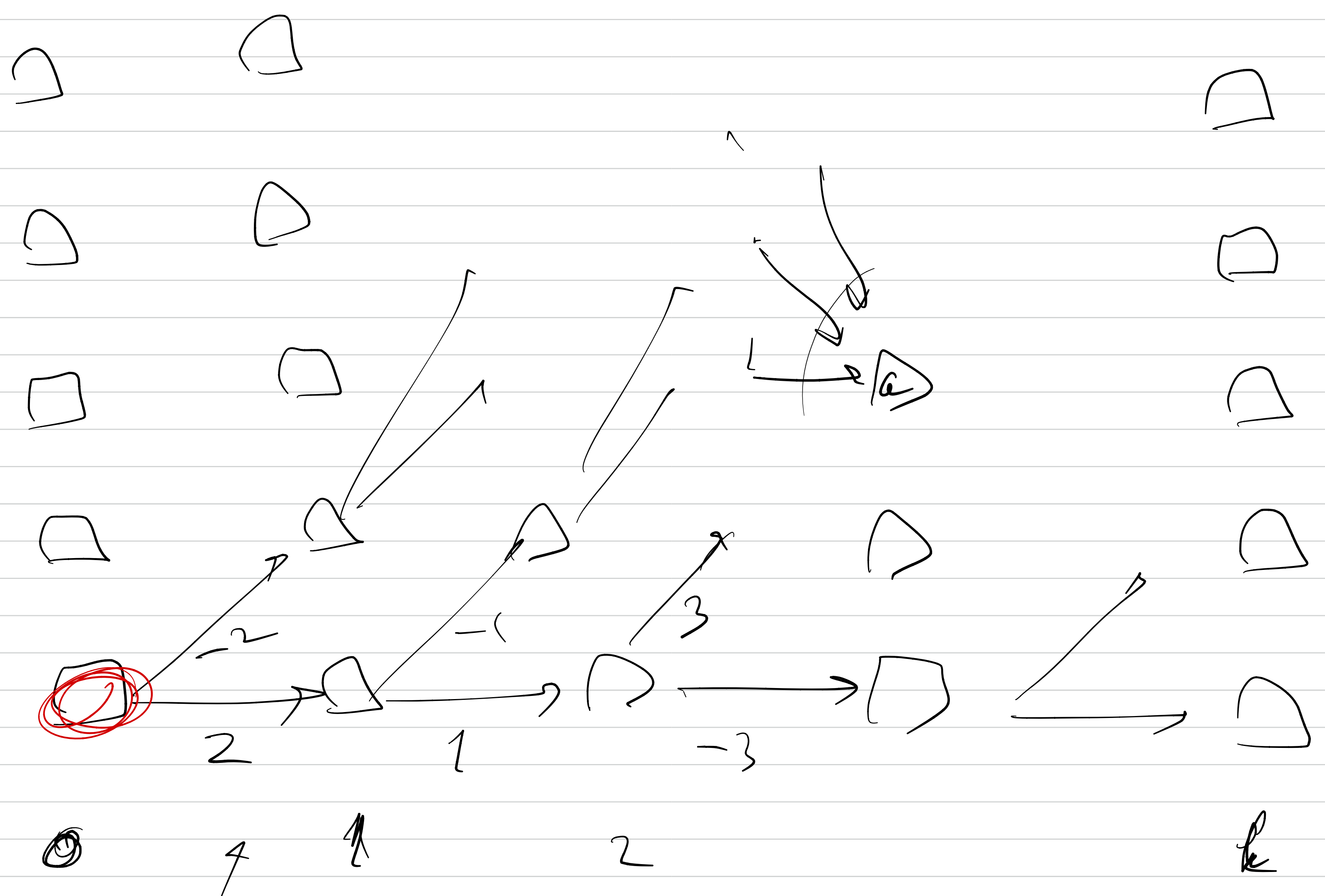


$(+y_{2t+1} - y_{2t+2})$

$(-y_{2t+1} + y_{2t+2})$

$j = 2t+1, \; 2t+2$

contribution to the $\langle , \rangle$ if we traverse that branch.



labeled by the contrib to the inner product

we try to find which path from 0 to the right edge has the largest reward?

Algorithm: go from left to right:

for node a at stage i,

$$\text{reward}(a) = \max_{\substack{b:\ \text{the nodes at stage } i-1 \\ \text{that can lead to } a}} \left(\text{reward}(b) + \text{reward}(b \to a) \right)$$

when this terminates at the right edge nodes.

The path found by the algorithm from the starting node to the most valuable right edge node identifies the MAP decision.

\# of operations prop to (states) $\times$ k.

$\Rightarrow$ the decoding effort is bounded / bits

See figure 6.4 on p. 210 for a fully worked out example.