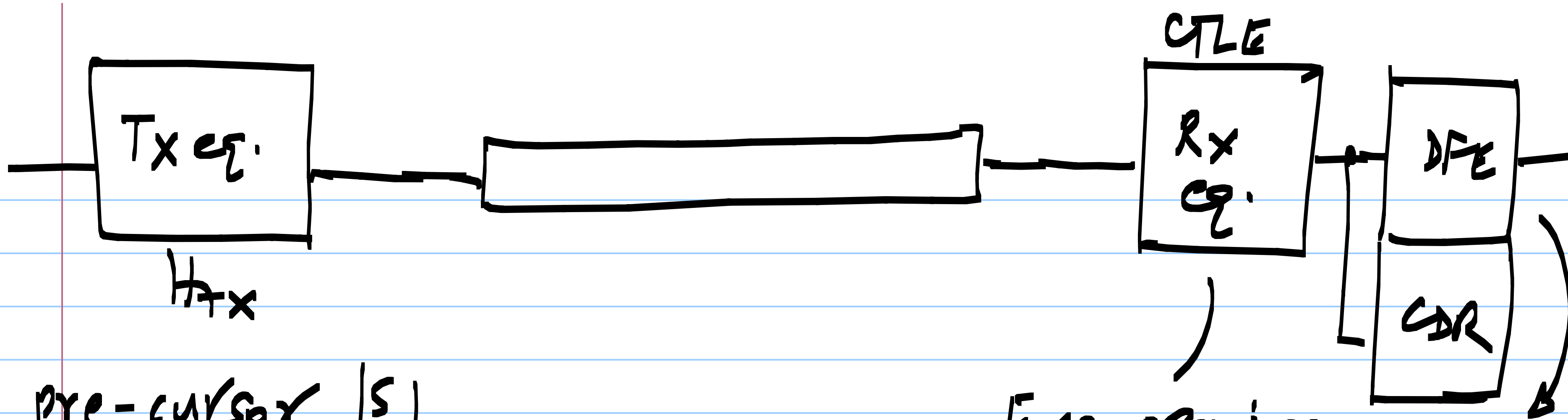


pre-cursor ISI

Eye-opening w/o relying CDR

post cursor ISI

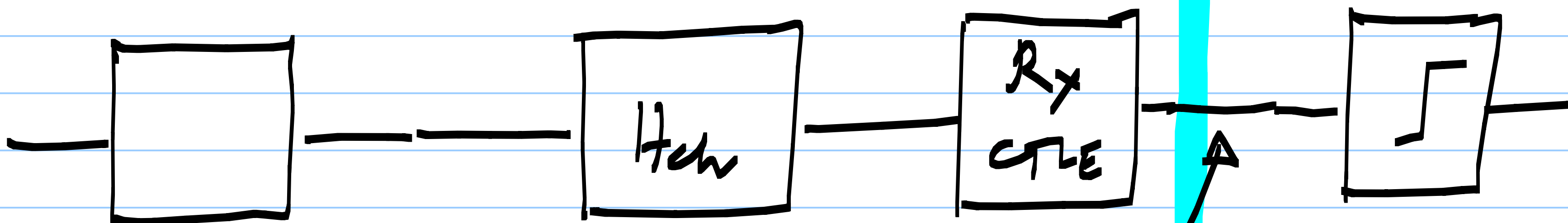
Speculative DFE:

- Eliminates the fb loop with the slicer
- only a MUX in the fb loop

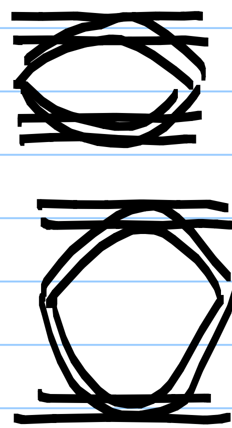
Closing the loop within a cycle → Speculative DFE

Processing has to be linear up to the slicer

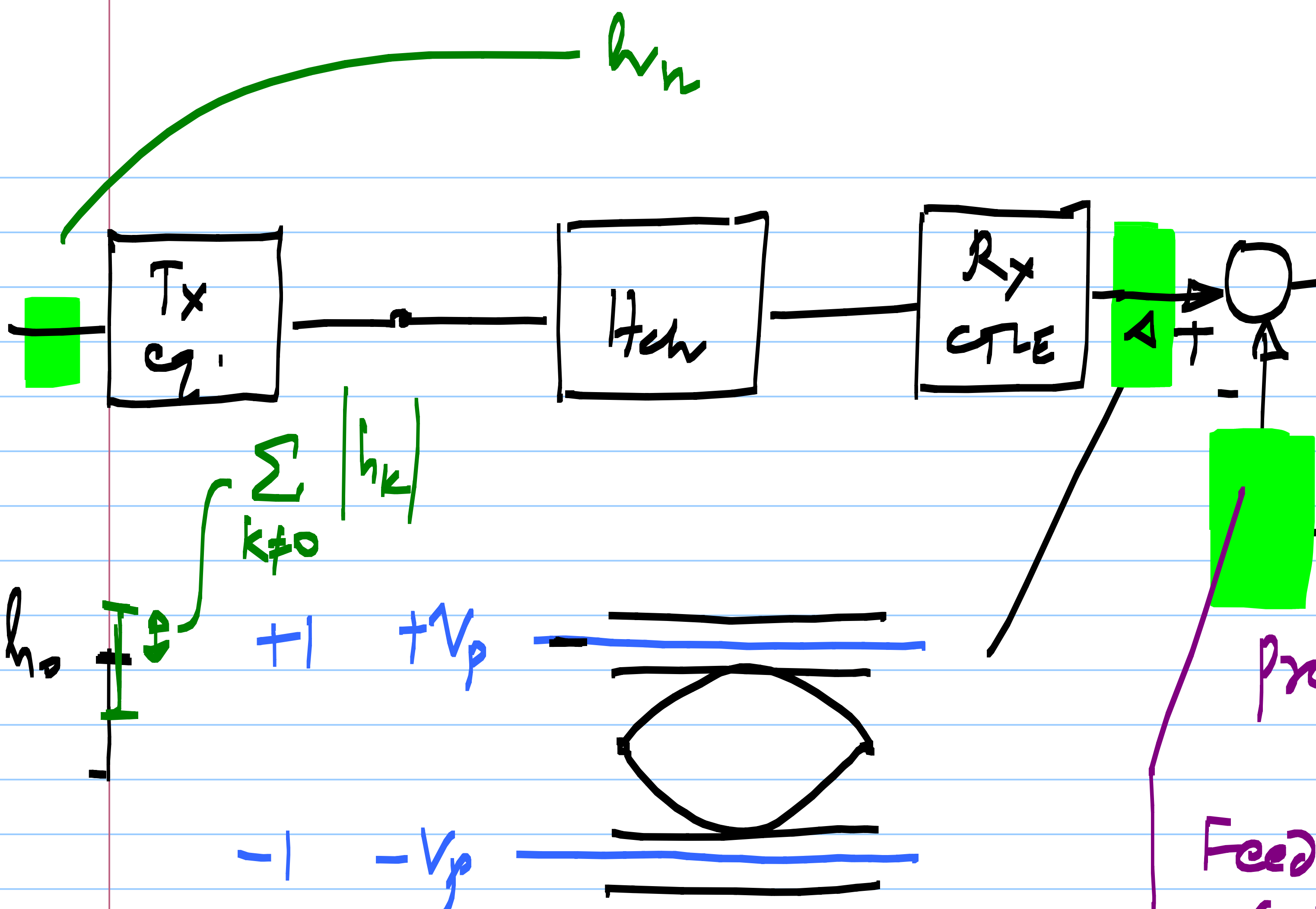
Circuits should behave linearly up to this point



Semi-digital implementation using switching diff. pairs

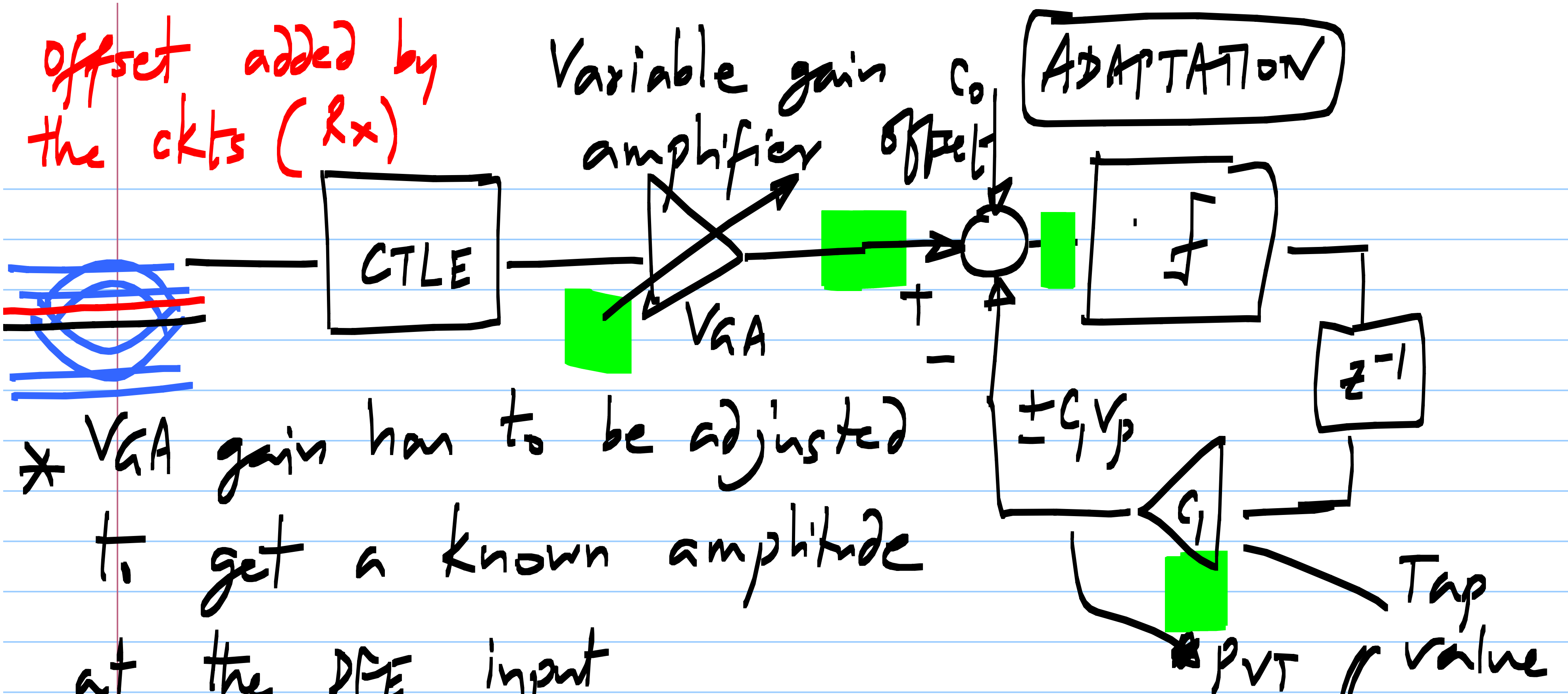


Amplitude here doesn't matter as long as the eye-opening is large enough (for binary data)



Previous symbol
= 1
Feedback for
current symbol
= C1 × error

Known amplitude relationship
between input & ~~feedback~~ $= c_1 \cdot V_p$

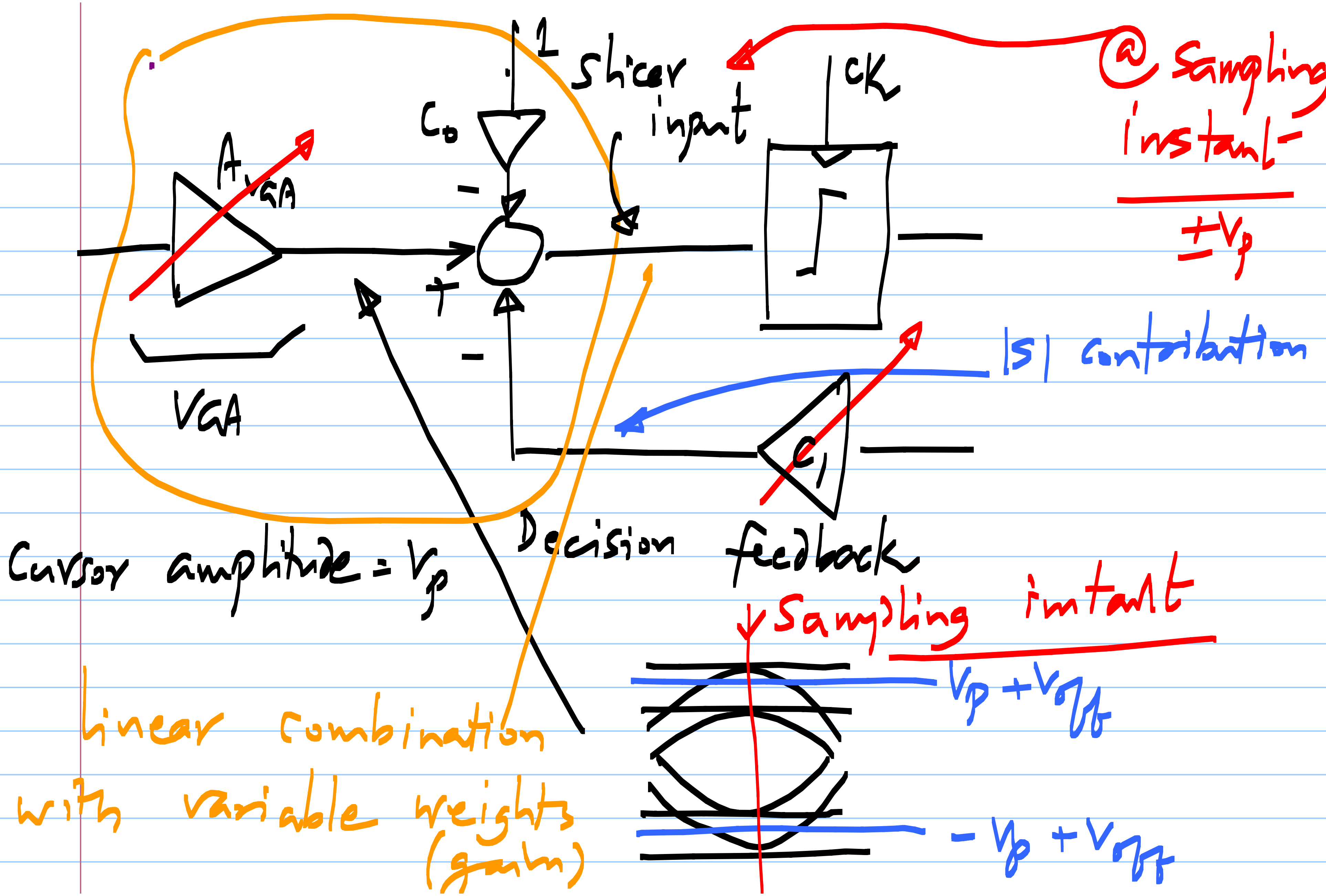


* VGA gain has to be adjusted to get a known amplitude at the DFE input

* DFE input amplitude:
- channel
- Tx eq. settings
- ckt variations (PVT)

Adaptation for VGA gain, offset cancel, DFE taps:

- DFE input amplitude : variable ⟵ channel / Tx eq. settings / PVT variations (Rx)
  - ⟶ Fix the amplitude using a VGA [fixes the cursor value]
- offset at DFE input — Circuits (Rx, Tx)
  - ⟶ Add offset cancellation
- DFE taps depend on ISI ⟵ channel / Tx. eq. settings


1
slicer input
CK
@ Sampling instant
$\pm V_p$
C_0
A_{VGA}
VGA
|S| contribution
C_1
Decision feedback
Cursor amplitude = V_p
Sampling instant
$V_p + V_{off}$
$-V_p + V_{off}$
linear combination with variable weights (gain)

DFE: feedback of a digital signal } $c_1 \cdot V_p$

- Voltage levels depend on DFE circuits (bias current of the tap diff. pairs)

Input to DFE: voltage levels depend on channel loss, Tx eq. settings. - cursor

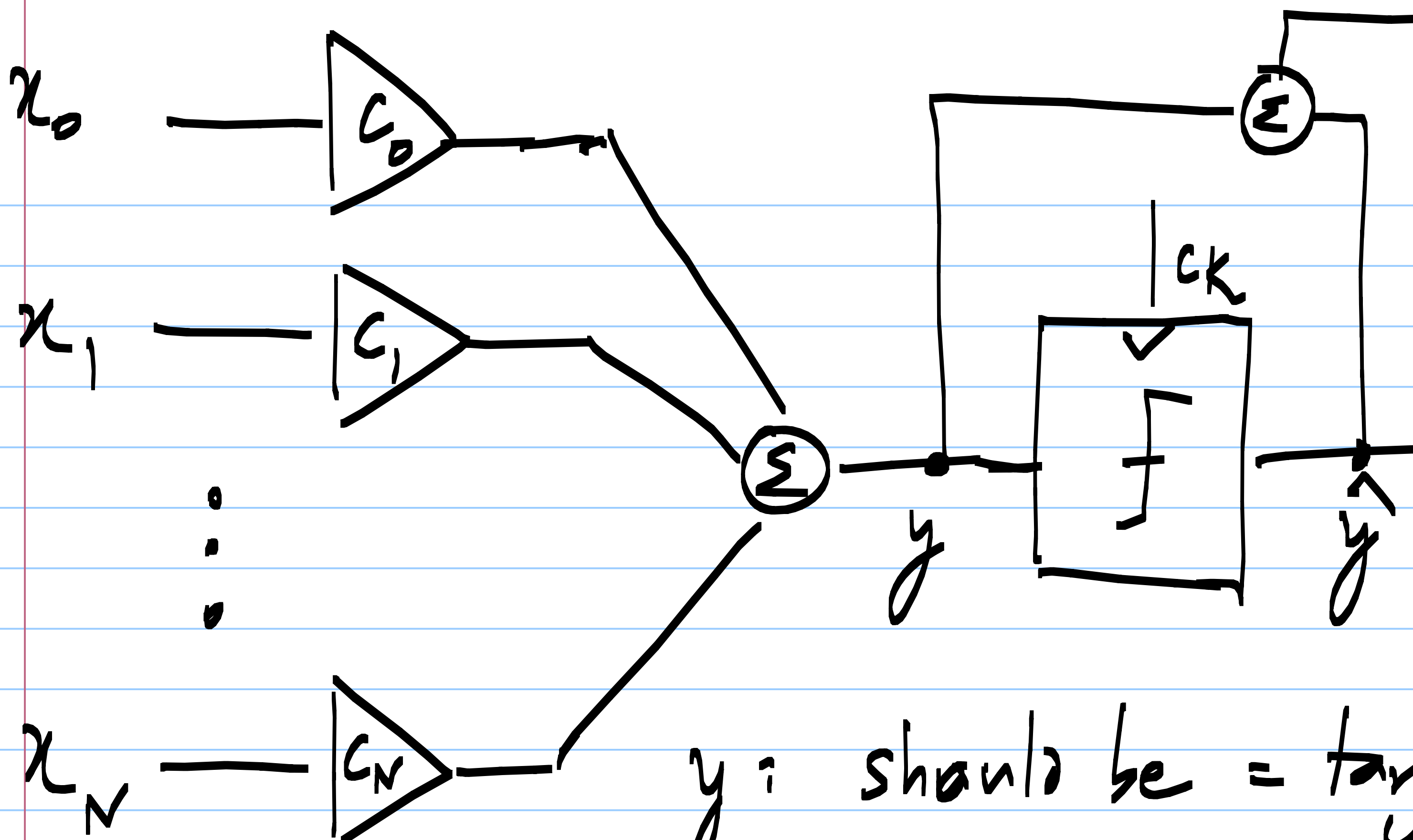
---

Least-mean-square (LMS) Adaptation:

Goal: Achieve a given target

- Have $\pm V_p$ at the slicer input at the sampling instants

→ Minimize the mean-squared error between the actual value and the target



$e = y - \hat{y}$
@ sampling instant

$$y = \sum_{k=0}^{N} c_k x_k$$

y : should be = target ± 1
if decision are correct
$\hat{y} = \pm 1$

Error $e = y - \hat{y} = \left(\sum_{k=0}^{N} c_k x_k - \hat{y} \right)$

Mean-squared error

$$= \overline{e^2} = \overline{(y - \hat{y})^2} = \underbrace{\overline{\left(\sum_{k=0}^{N} c_k x_k - \hat{y} \right)^2}}$$

Minimize $\overline{e^2}$

[Averaged over many symbols]

Determine c_k such that $\overline{e^2}$ is minimized

$$\sum_{0}^{N-1} c_k x_k[l] = \hat{y}_k[l]$$

what we want

l: time index

L:

$L > N$

$L \in \{$:

linear combination of N inputs

N variables

$$\begin{array}{ccc} L \times N & N \times 1 & L \times 1 \\ \begin{bmatrix} x_0[1] & x_1[1] & \cdots & x_{N-1}[1] \\ x_0[2] & x_1[2] & \cdots & x_{N-1}[2] \\ & & & \\ x_0[L] & x_1[L] & \cdots & x_{N-1}[L] \end{bmatrix} & \begin{bmatrix} c_0 \\ c_1 \\ \\ c_{N-1} \end{bmatrix} = & \begin{bmatrix} \hat{y}[1] \\ \hat{y}[2] \\ \\ \hat{y}[L] \end{bmatrix} \end{array}$$

$$A \cdot \bar{c} = \bar{\hat{y}} \qquad \bar{c} \cancel{=} A^{-1}\bar{\hat{y}}$$

$$L \times N \quad N \times 1 \qquad L \times 1$$

Square matrix

$$(A^T A) \cdot \bar{c} = A^T \cdot \bar{\hat{y}}$$

$$\bar{c} = \underbrace{(A^T A)^{-1} \cdot A^T}_{\text{pseudo inverse}} \cdot \bar{\hat{y}}$$

Min. mean squared error solution

10 points

(x_0, y_0)

$\vdots$

(x_{10}, y_{10})

$mx + c = y$ $\quad (A^T A)^{-1} A^T \cdot y$

$$\underbrace{\begin{bmatrix} x_0 & 1 \\ x_1 & 1 \\ & \\ x_{10} & 1 \end{bmatrix}}_{[A]} \begin{bmatrix} m \\ c \end{bmatrix} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \\ y_N \end{bmatrix}}_{y}$$