8.10 Signal Processing Examples of LS

We'll briefly look at two examples from the book…

Book Examples

1. Digital Filter Design
2. AR Parameter Estimation for the ARMA Model
3. Adaptive Noise Cancellation
4. Phase-Locked Loop (used in phase-coherent demodulation)

The two examples we will cover highlight the flexibility of the LS viewpoint!!!

Then (in separate note files) we'll look in detail at two emitter location examples not in the book

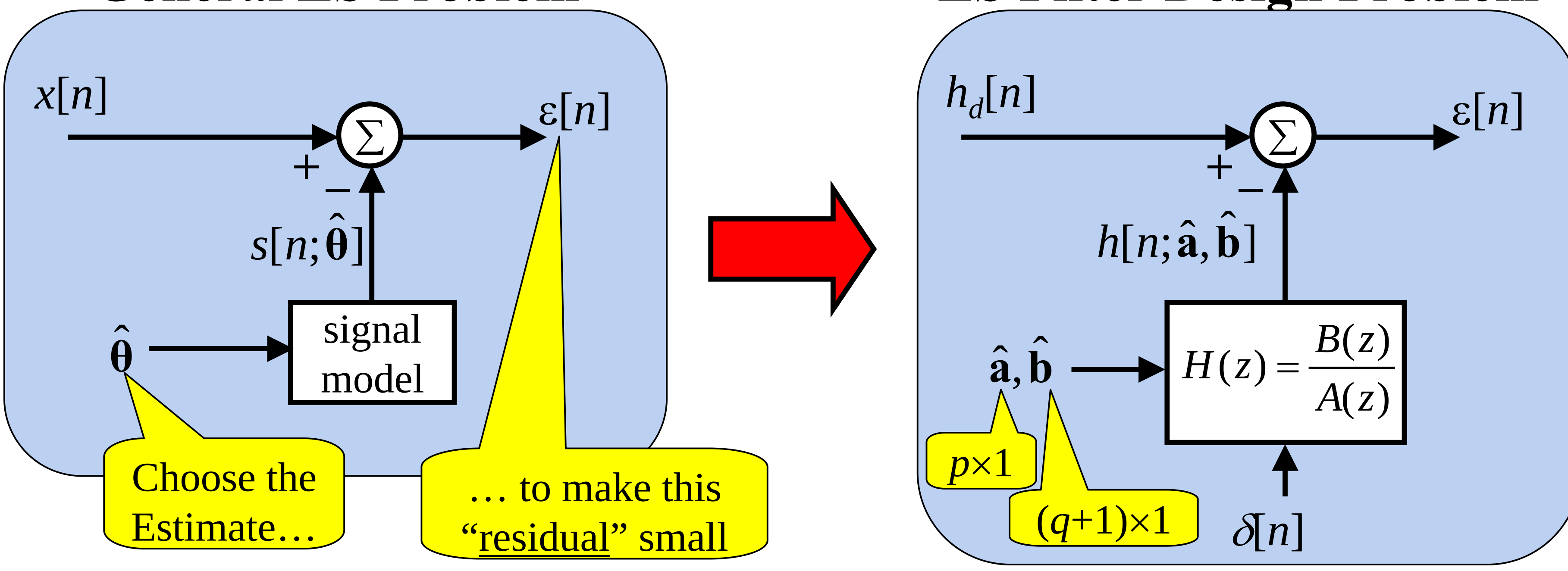
Ex. 8.11 Filter Design by Prony's LS Method

The problem:

- You have some desired impulse response $h_d[n]$
- Find a <u>rational</u> TF with impulse response $h[n] \approx h_d[n]$

View: $h_d[n]$ as the observed "data"!!!
Rational TF model's coefficients as the parameters

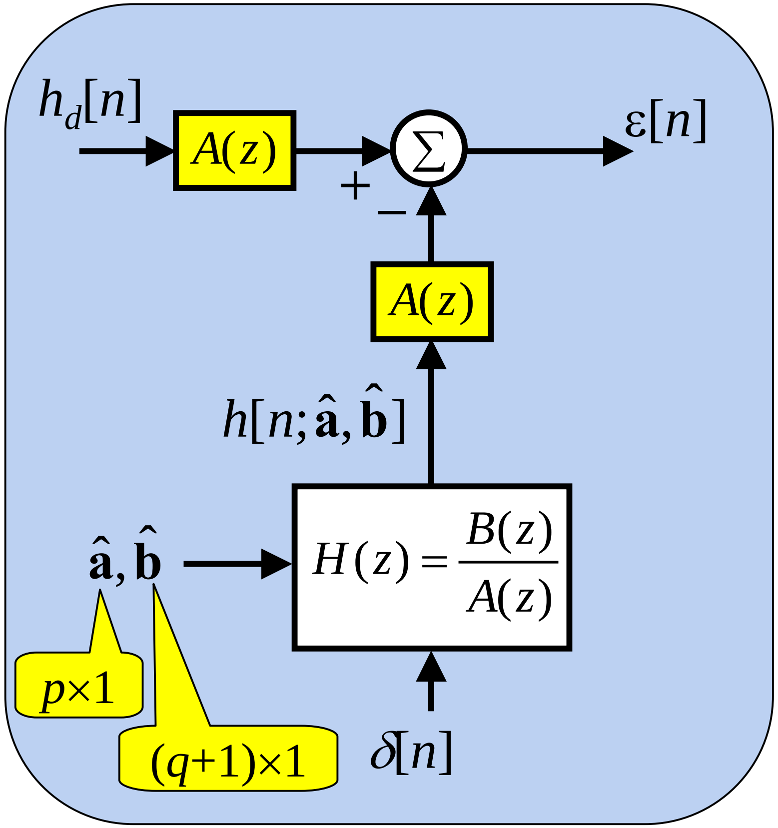
General LS Problem **LS Filter Design Problem**



Prony's Modification to Get Linear Model

The previous formulation results in a model that is nonlinear in the TF coefficient vectors **a, b**

Prony's idea was to change the model slightly…



This model is only approximately equivalent to the original!!

Solution (see book for details):

$$\hat{\mathbf{a}} = [\mathbf{H}_q^T \mathbf{H}_q]^{-1} \mathbf{H}_q^T \mathbf{h}_{q,N-1}$$

$$\hat{\mathbf{b}} = \mathbf{h}_{0,q} + \mathbf{H}_0 \hat{\mathbf{a}}$$

$\mathbf{H}_q$, $\mathbf{H}_0$, $\mathbf{h}_{q,N\text{-}1}$, and $\mathbf{h}_{0,q}$ all contain elements from $h_d[n]$… the subscripts indicate the range of these elements

<u>Key Ideas in Prony LS Example</u>

1. Shows power and flexibility of LS approach
   - There is <u>no noise</u> here!!! $\Rightarrow$ MVU, ML, etc. are not applicable
   - But, LS works nicely!

2. Shows a slick trick to convert nonlinear problem to linear one
   - Be aware that finding such tricks is an art!!!

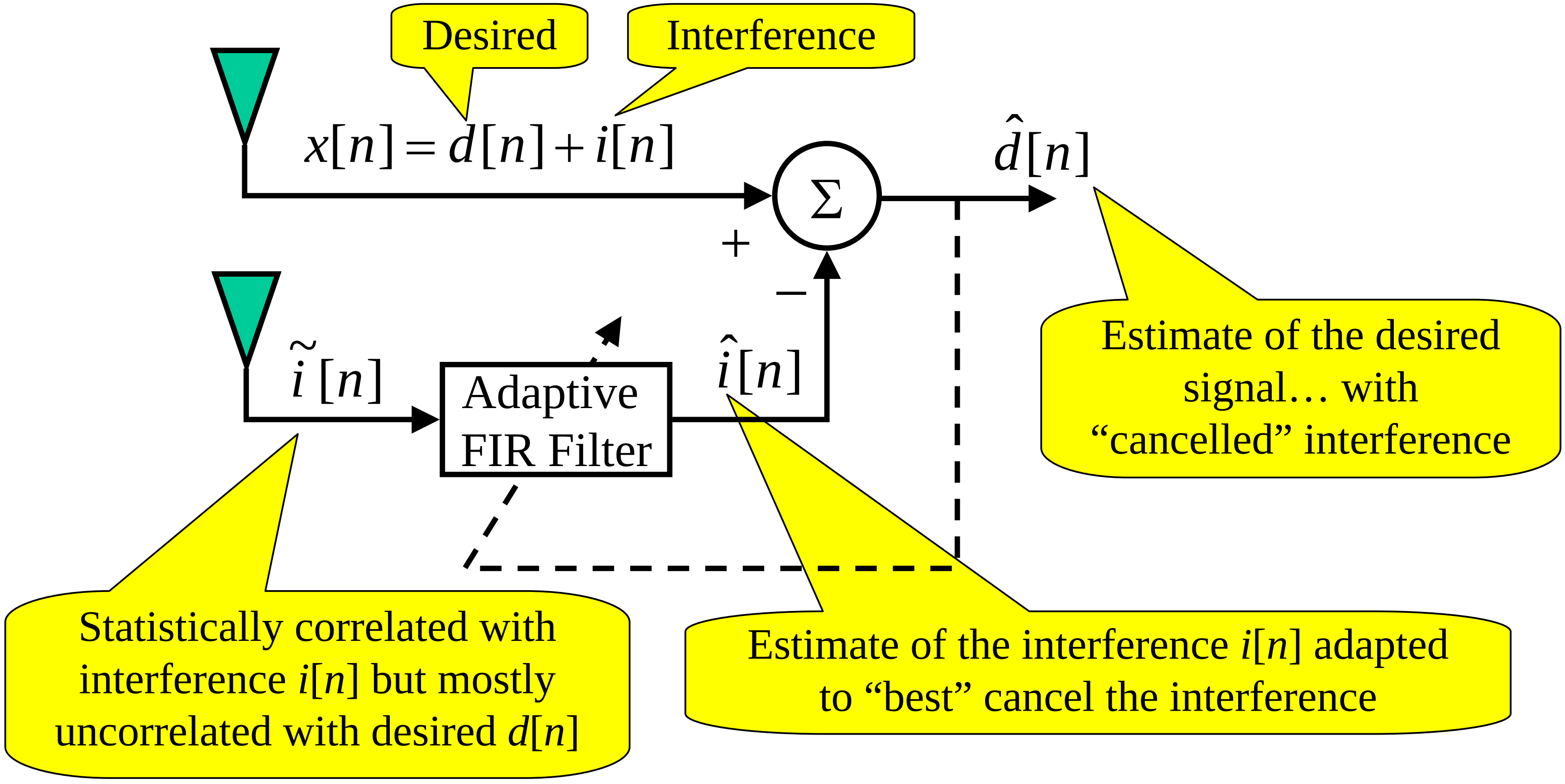
3. Results for LS "Prony" method have links to modeling methods for Random Processes (i.e. AR, MA, ARMA)

Is this a practical filter design method?

It's not the best: Remez-Based Method is Used Most

Ex. 8.13 Adaptive Noise Cancellation

Done a bit different from the book

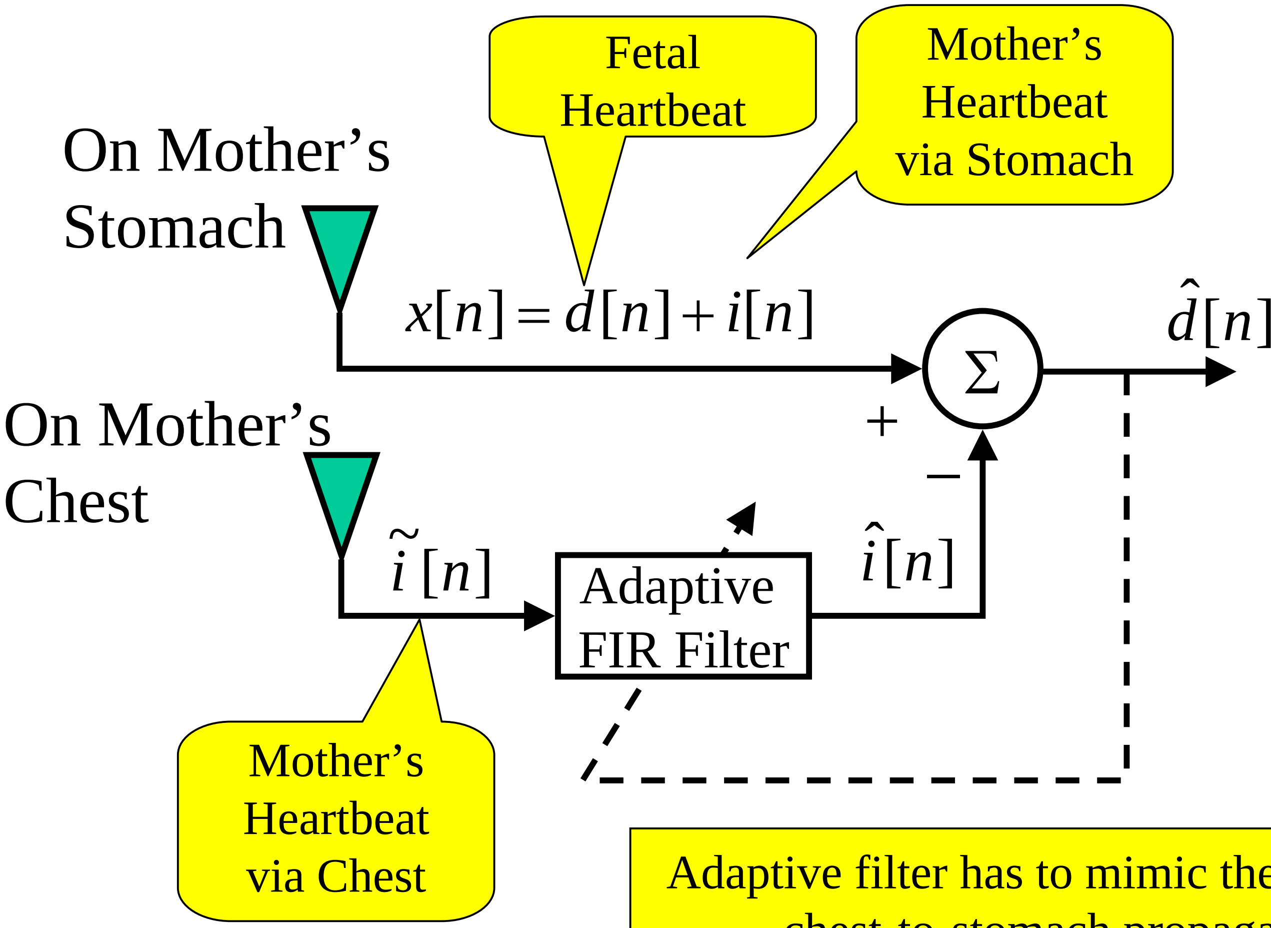


$$\hat{i}[n] = \sum_{l=0}^{p} h_n[l]\,\tilde{i}[k-l]$$

Time-Varying Filter!!
Coefficients change at each sample index

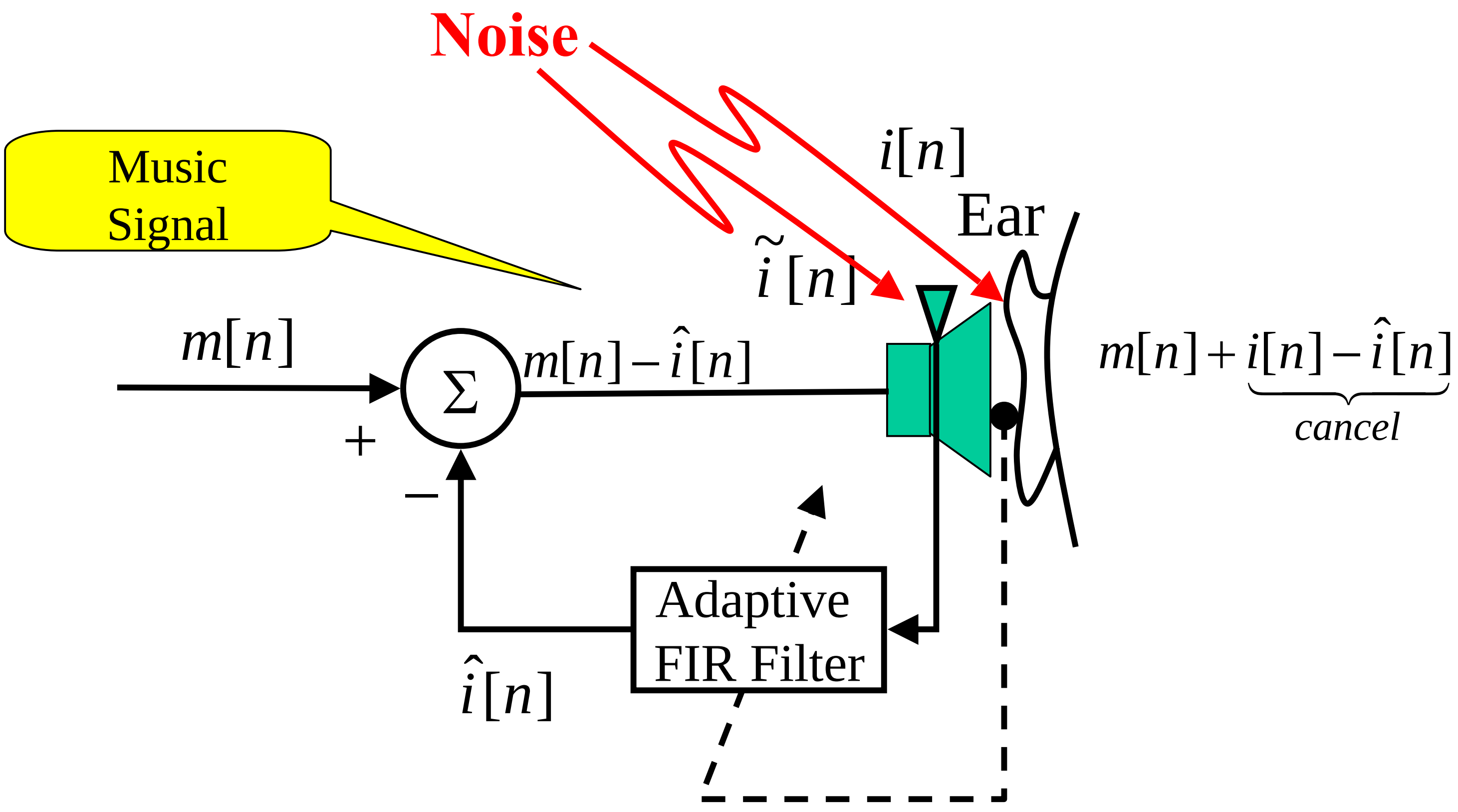
Noise Cancellation Typical Applications

1. Fetal Heartbeat Monitoring

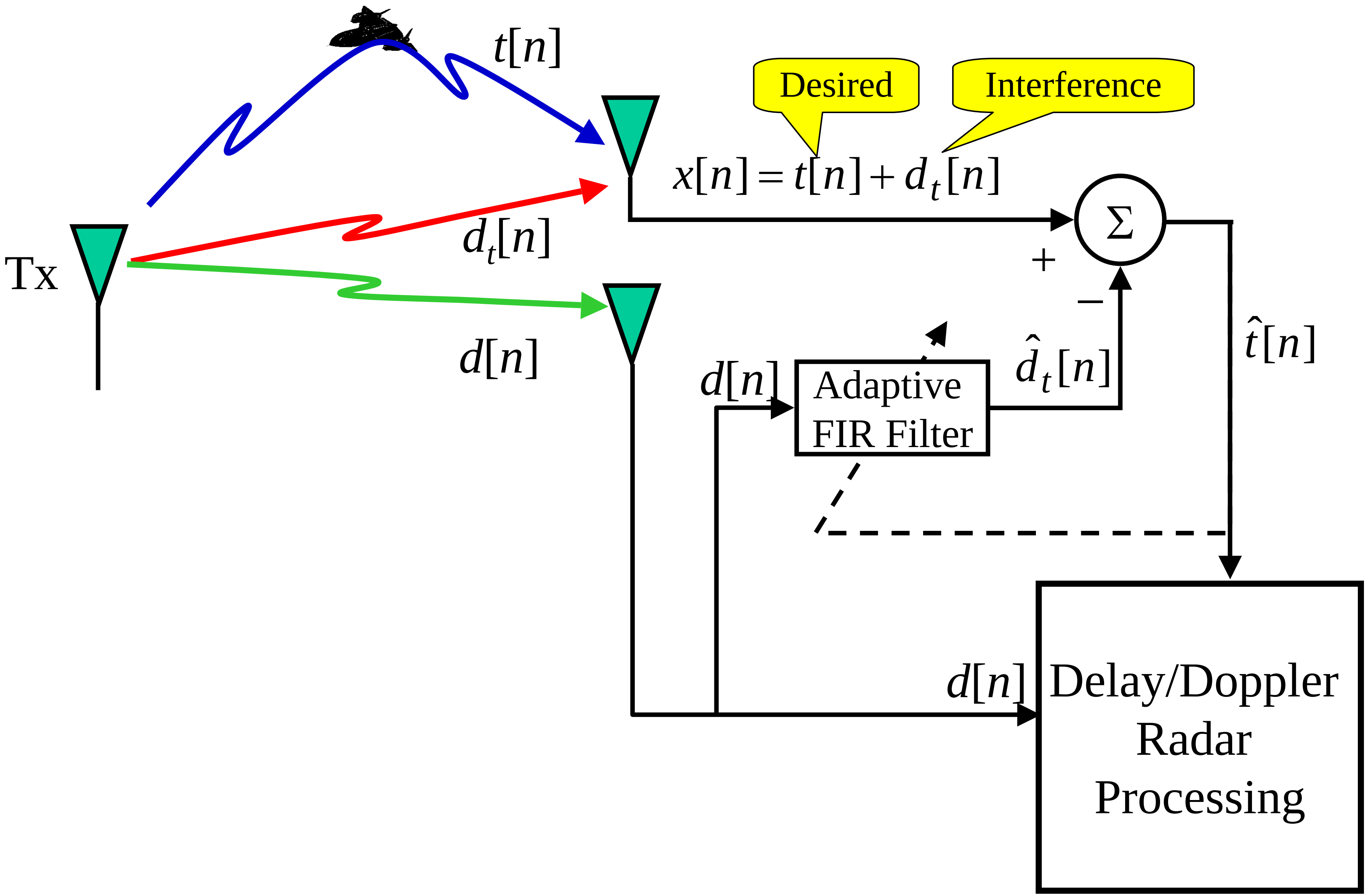


Adaptive filter has to mimic the TF of the chest-to-stomach propagation

2. Noise Canceling Headphones



3. Bistatic Radar System



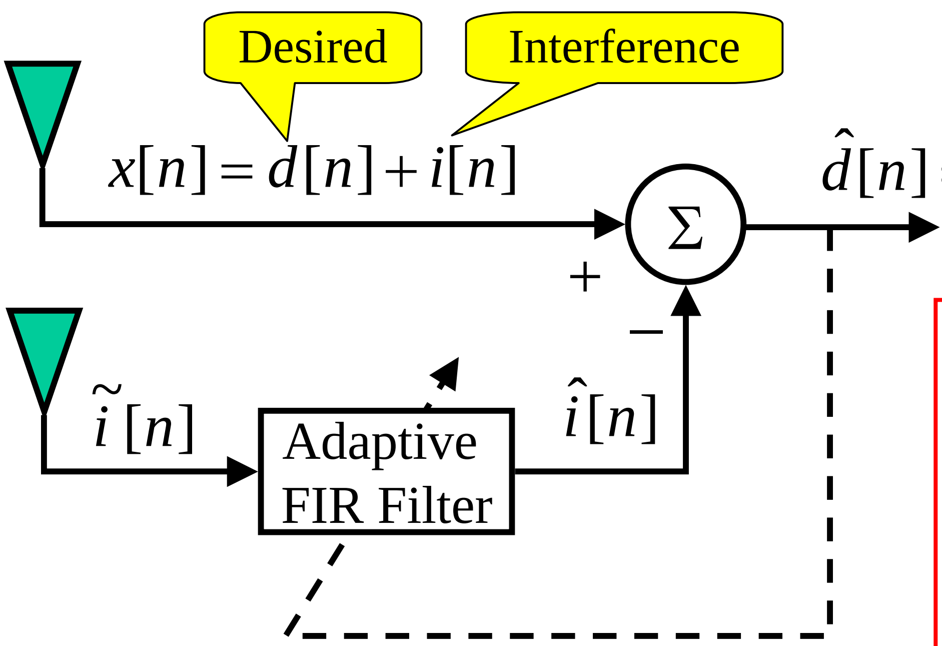
<u>LS and Adaptive Noise Cancellation</u>

<u>Goal</u>: Adjust the filter coefficients to cancel the interference

There are many signal processing approaches to this problem…

We'll look at this from a LS point of view:

Adjust the filter coefficients to minimize $J = \sum_{n} \hat{d}^{2}[n]$



$\hat{d}[n] = d[n] + (i[n] - \hat{i}[n])$

Because $i[n]$ is uncorrelated with $d[n]$ minimizing J is essentially the same as making this term zero

Because the interference likely changes is character with time… we want to adapt!

Use <u>Sequential</u> LS with "Fading Memory"

Sequential LS with Forgetting Factor

We want to weight recent measurements more heavily than past measurements… that is we want to "forget" past values.

So we can use weighted LS… and if we choose our weighting factor as an exponential function then it is easy to implement!

$$J[n] = \sum_{k=0}^{n} \lambda^{n-k} \left[x[k] - \hat{i}[k] \right]^2$$

$$= \sum_{k=0}^{n} \lambda^{n-k} \left[x[k] - \sum_{l=0}^{p-1} h_n[l] \tilde{i}[k-l] \right]^2$$

Small λ quickly "down weights" the past errors

λ = forgetting factor if 0 < λ < 1

See book for solution details

See Fig. 8.17 for simulation results