EC 614: Adaptive Signal Processing Techniques

Course Instructor: Dr. Debashis Ghosh

Department of Electronics & Computer Engg.

Indian Institute of Technology Roorkee

Syllabus

- Definitions, assumptions and requirements of adaptive signal processing applicable to different application examples; Linear filter structures; Adaptive beamforming (4 lectures).

- Optimum linear combiner and Wiener-Hopf equations, orthogonality principle, minimum mean square error and error performance surface; Multiple linear regression model; Linearly constrained minimum-variance filter (6 lectures).

Syllabus (contd.)

- Forward and backward prediction error filters; Levinson—Durbin algorithm; Properties of prediction-error filters; Autoregressive modeling of a stationary stochastic process; All-pole, all-pass lattice filter (8 lectures).
- Steepest-descent algorithm and its stability; Principles of stochastic gradient descent, LMS algorithm and its variants (4 lectures).
- Least Squares method, its efficient implementation: Minimum sum of error squares, normal equations and linear least-squares filters, Singular value decomposition, cyclic Jacobi, Householder methods (8 lectures).

Syllabus (contd.)

- RLS adaptive filtering algorithms; Exponentially weighted RLS algorithm; Kalman filter and its variants; Square-root adaptive filters, adaptive beamforming (8 lectures).

- Implementation examples: Adaptive modeling and system identification, inverse adaptive modeling, equalization and deconvolution, adaptive control systems, adaptive interference cancellation (4 lectures).

Suggested Texts / References

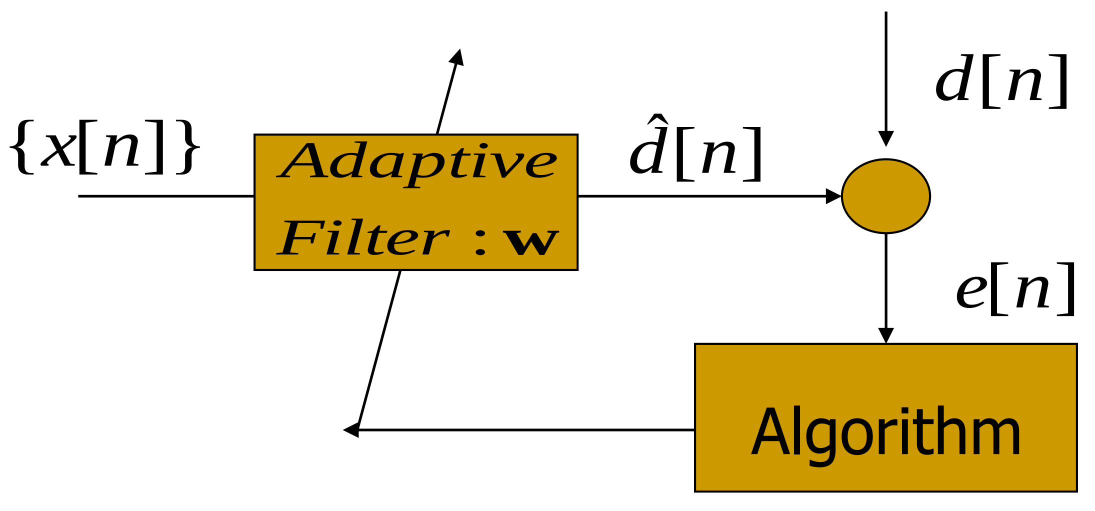
- S. Haykin, “Adaptive Filter Theory”, Pearson Education.
- B. Widrow and S.D. Stearns, “Adaptive Signal Processing”, Pearson Education.
- D.G. Manolakis, V.K. Ingle and M.S. Kogon, “Statistical and Adaptive Signal Processing”, Artech House.
- H. Sayed Ali, “Fundamentals of Adaptive Filtering”, Wiley-Interscience, IEEE Press.
- H. Sayeed Ali, “Adaptive Filters”, 1st Edn., John Wiley & Sons.
- P.S.R. Diniz, “Adaptive Filtering: Algorithms and Practical Implementation”, Springer.

Basics

- **Problem:** Equalise through an FIR filter the distorting effect of a communication channel that may be changing with time.
- If the channel were fixed then a possible solution could be based on the Wiener filter approach
- We need to know in such case the correlation matrix of the transmitted signal and the cross correlation vector between the input and desired response.
- When the filter is operating in an unknown environment these required quantities need to be found from the accumulated data.
- The problem is particularly acute when not only the environment is changing but also the data involved are non-stationary

Basics

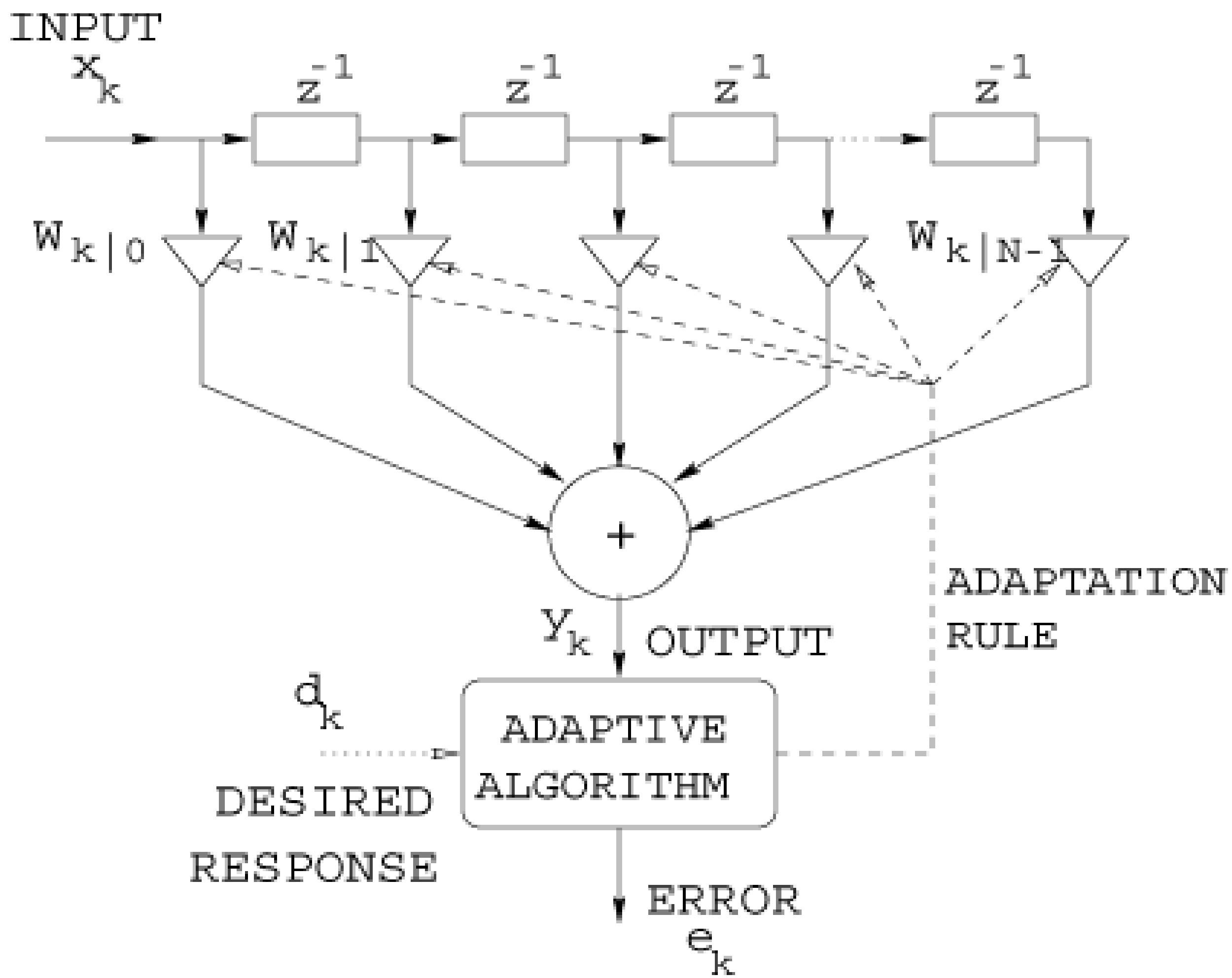
- In such cases we need temporally to follow the behaviour of the signals, and adapt the correlation parameters as the environment is changing.
- This would essentially produce a **temporally adaptive filter**.



Applications of Adaptive DSP

- Digital communications
- Channel equalization
- Adaptive noise cancellation
- Adaptive echo cancellation
- System identification
- Smart antenna systems
- Blind system equalization
- And many, many others

Application: Channel Equalization



Example: Weiner Filter

- FIR filter tap weights can be found using the Weiner-Hopf equation: $\mathbf{R}\mathbf{w}_{\text{opt}} = \mathbf{p}$
  - where correlation matrix $\mathbf{R} = E\,[\,\mathbf{x}(n)\,\mathbf{x}^{H}(n)\,]$ and cross-correlation vector $\mathbf{p} = E\,[\,\mathbf{x}(n)\,d^{*}(n)\,]$
- Based on the minimization of the mean-square error $J(n) = E\,[\,e(n)\,e^{*}(n)\,]$ where $e(n) = d(n) - \mathbf{w}^{H}.\mathbf{x}(n)$
- For changing signal statistics and/or channel condition, the filter weights are required to be updated continuously over time.

Example: Weiner Filter

- One approach may be using steepest-descent method.
- We will study later that the weight updation may be done as

$$\delta\mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n)$$

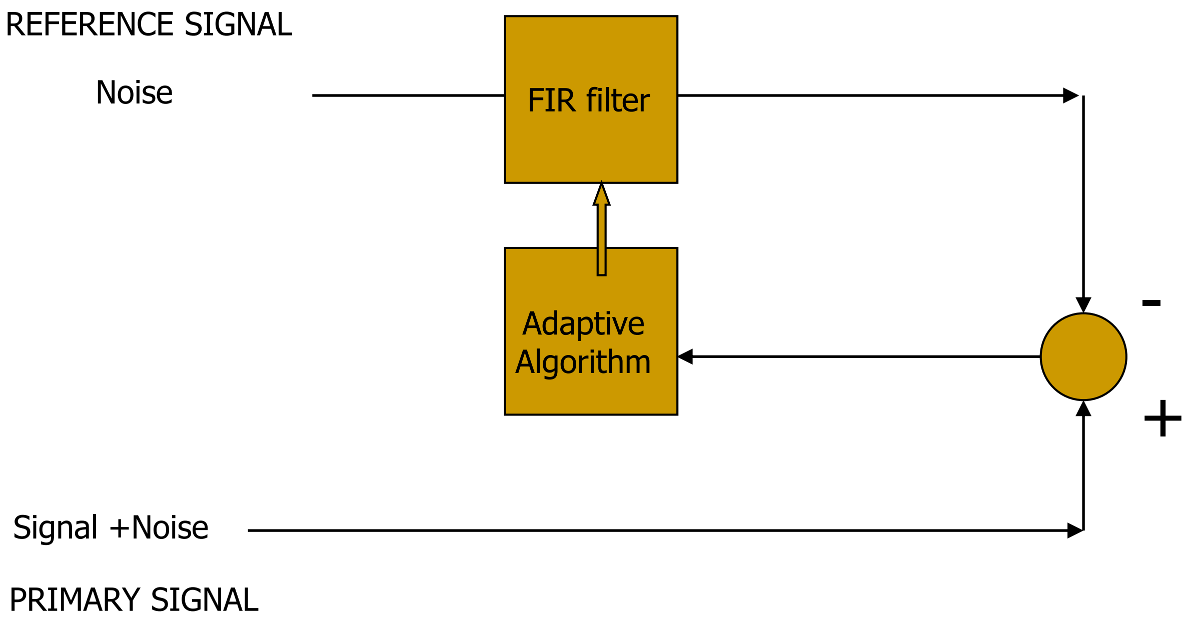
$$= \mu\,[\,\mathbf{p} - \mathbf{R}\mathbf{w}(n)\,]$$

$$= \mu\,E\,[\,\mathbf{x}(n)d^*(n) - \mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}(n)\,]$$

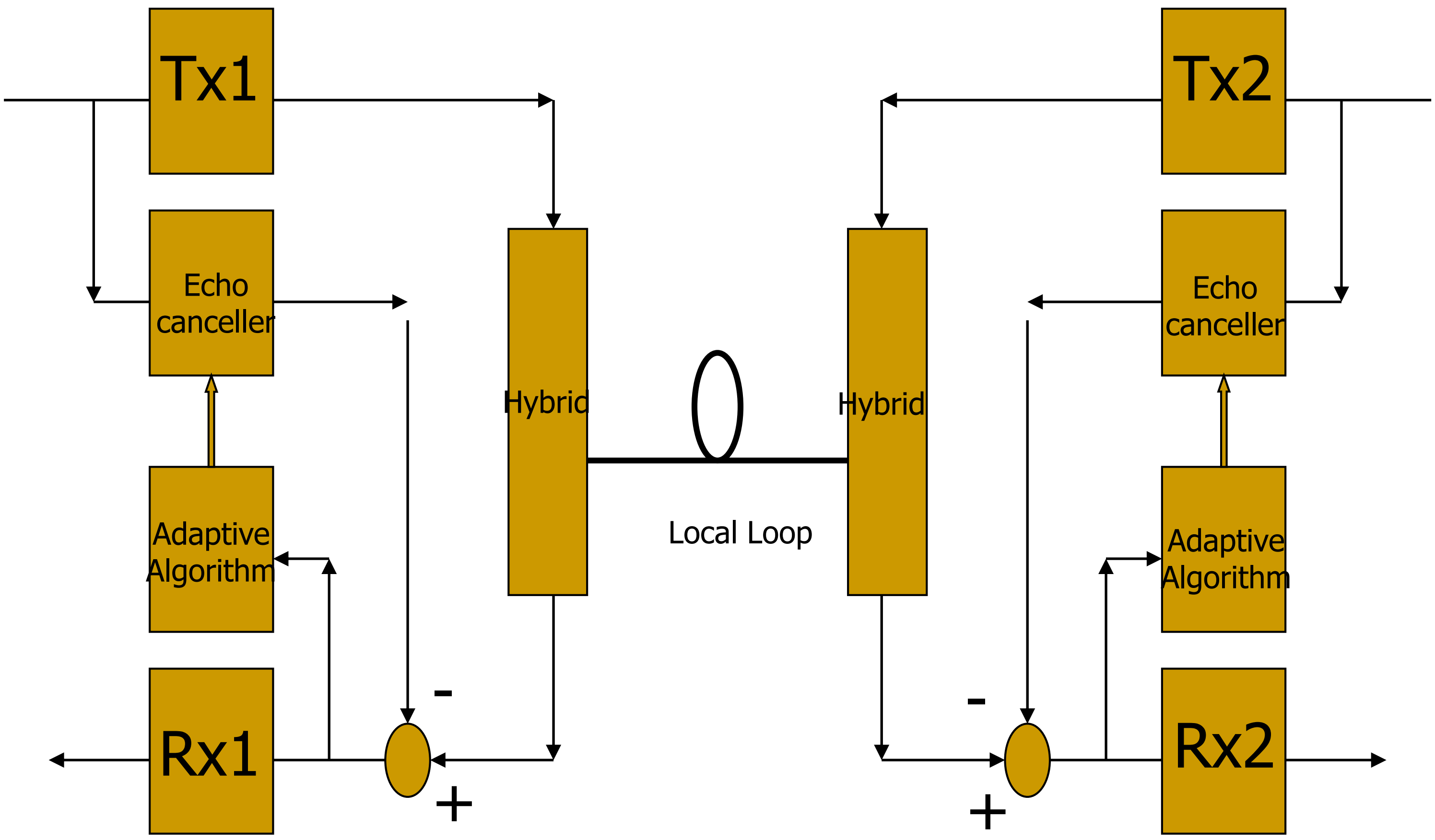
$$= \mu\,E\,[\,\mathbf{x}(n)\,\{\,d^*(n) - \mathbf{x}^H(n)\mathbf{w}(n)\,\}\,]$$

$$= \mu\,E\,[\,\mathbf{x}(n)\,e^*(n)\,]$$

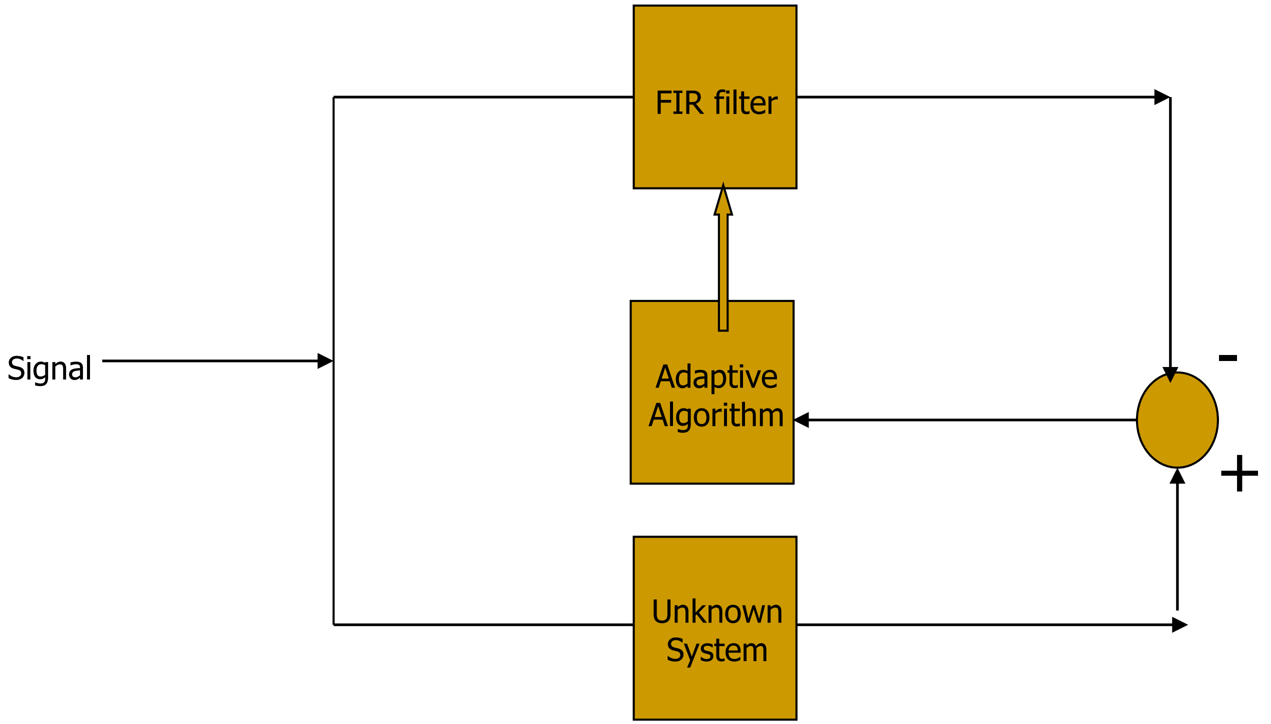
Application: Noise Cancellation



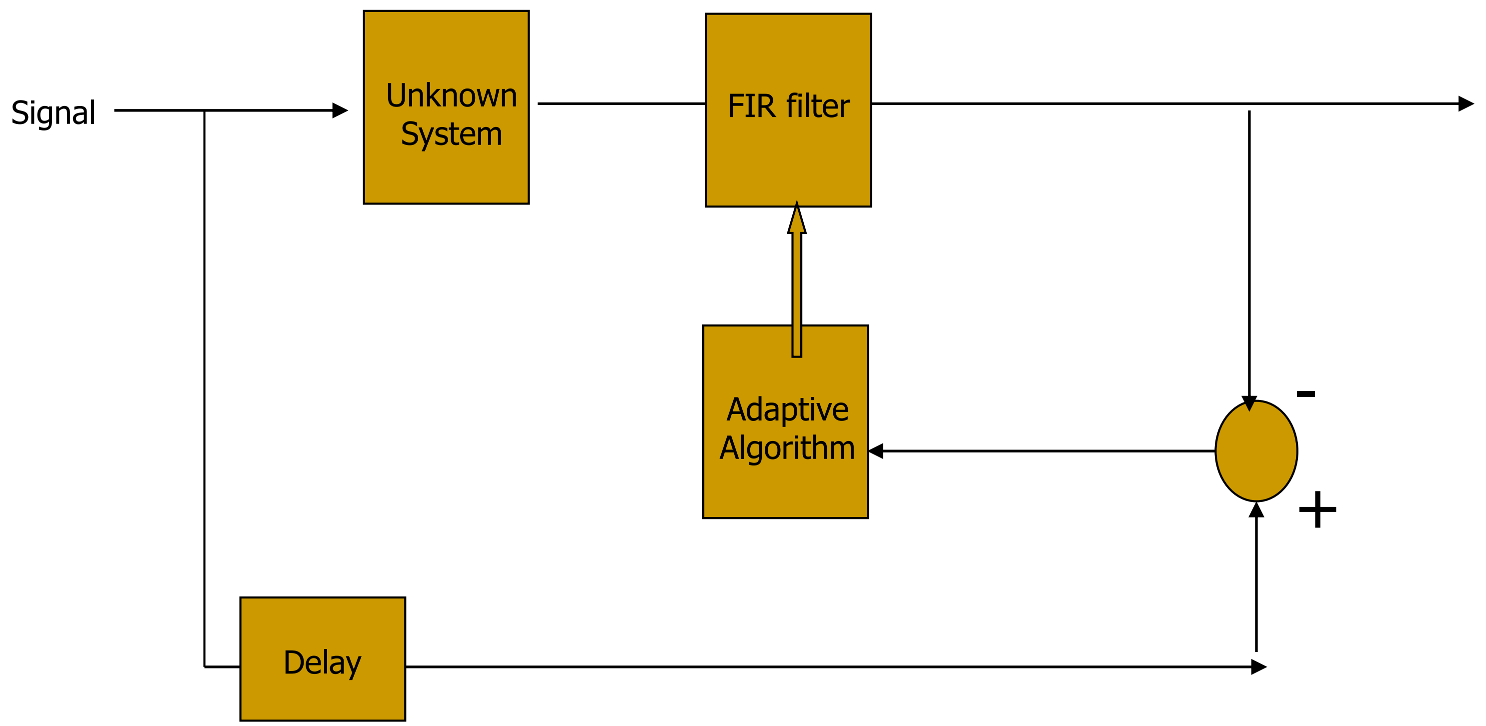
Application: Echo Cancellation



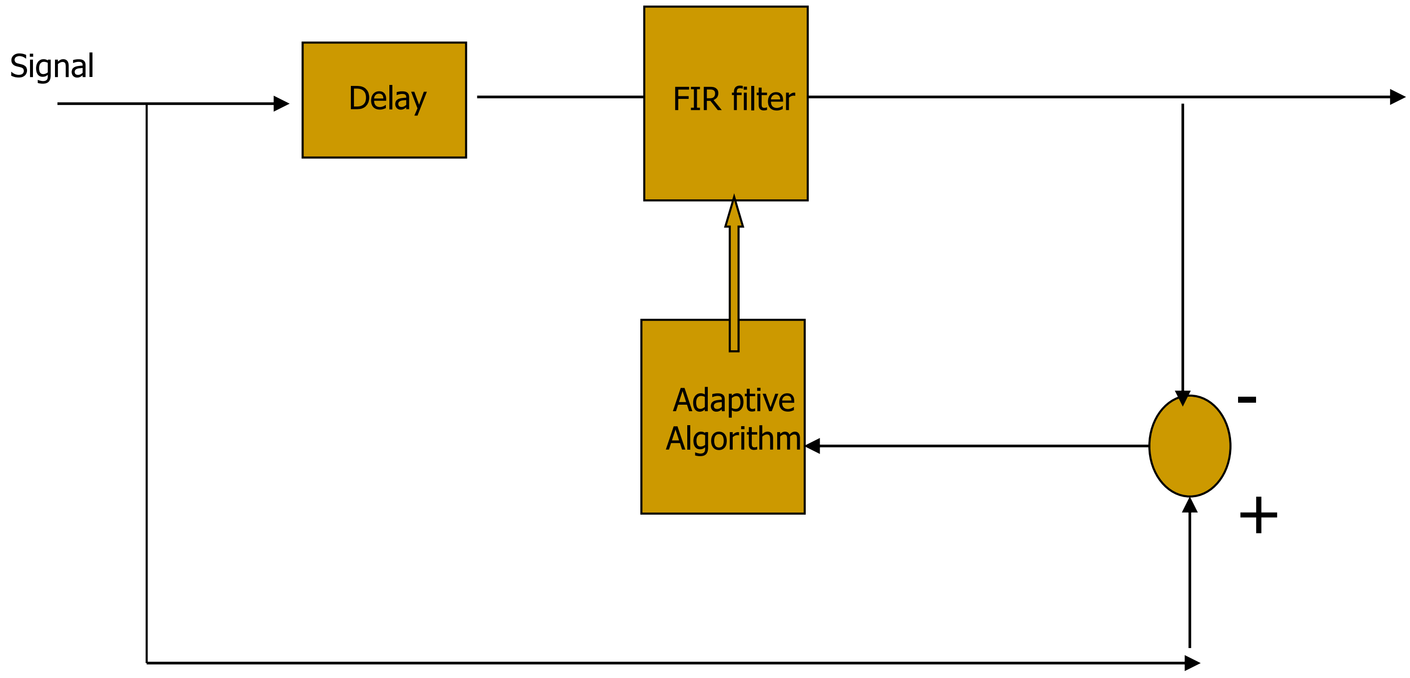
Application: System Identification



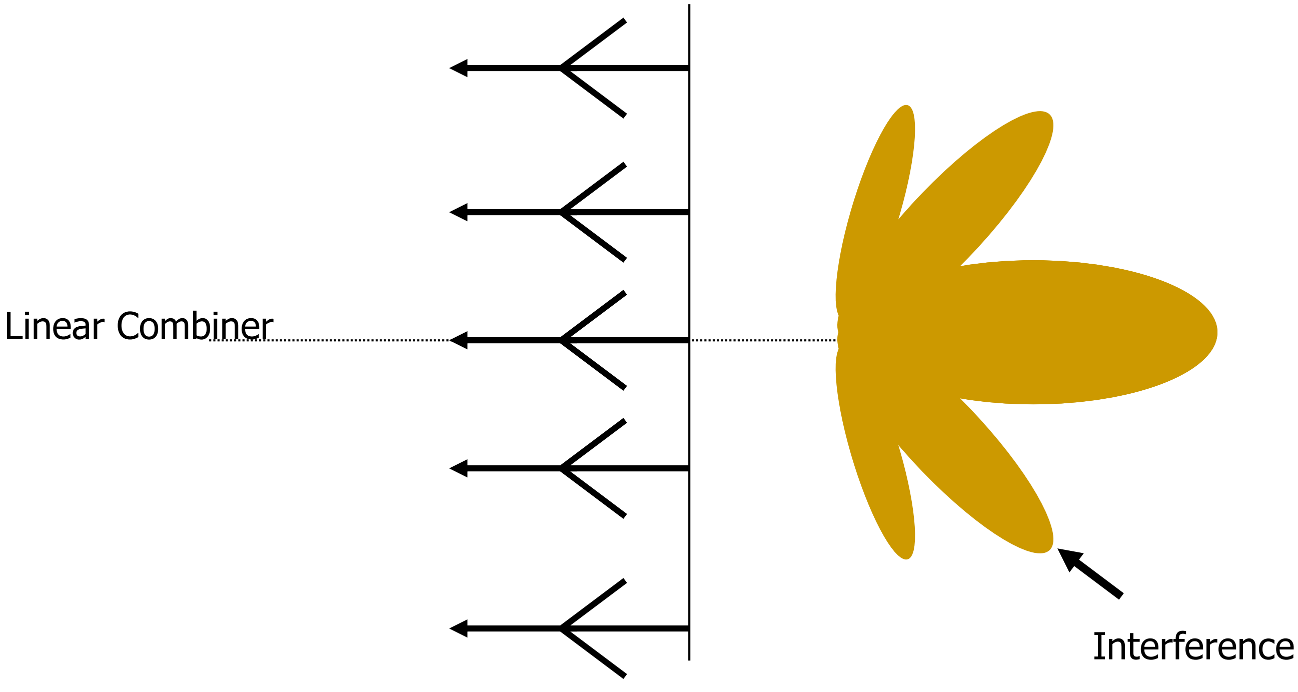
Application: Blind Equalization



Application: Adaptive Predictors



Application: Smart antenna arrays



Algorithm for adaptation

- Basic principles:
  - Form an objective function (performance criterion)
  - Find gradient of objective function with respect to FIR filter weights
  - There are several different approaches that can be used at this point, e.g. steepest-descent method.
  - Form a differential / difference equation from the gradient.

Adaptive Filters

- An adaptive filter is in reality a ***nonlinear device***, in the sense that it does not obey the principle of superposition.
- Adaptive filters are commonly classified as:
  - **Linear**

    An adaptive filter is said to be *linear* if the estimate of quantity of interest is computed adaptively (at the output of the filter) as a *linear combination* of the available set of observations applied to the filter input.
  - **Nonlinear**

    Neural Networks

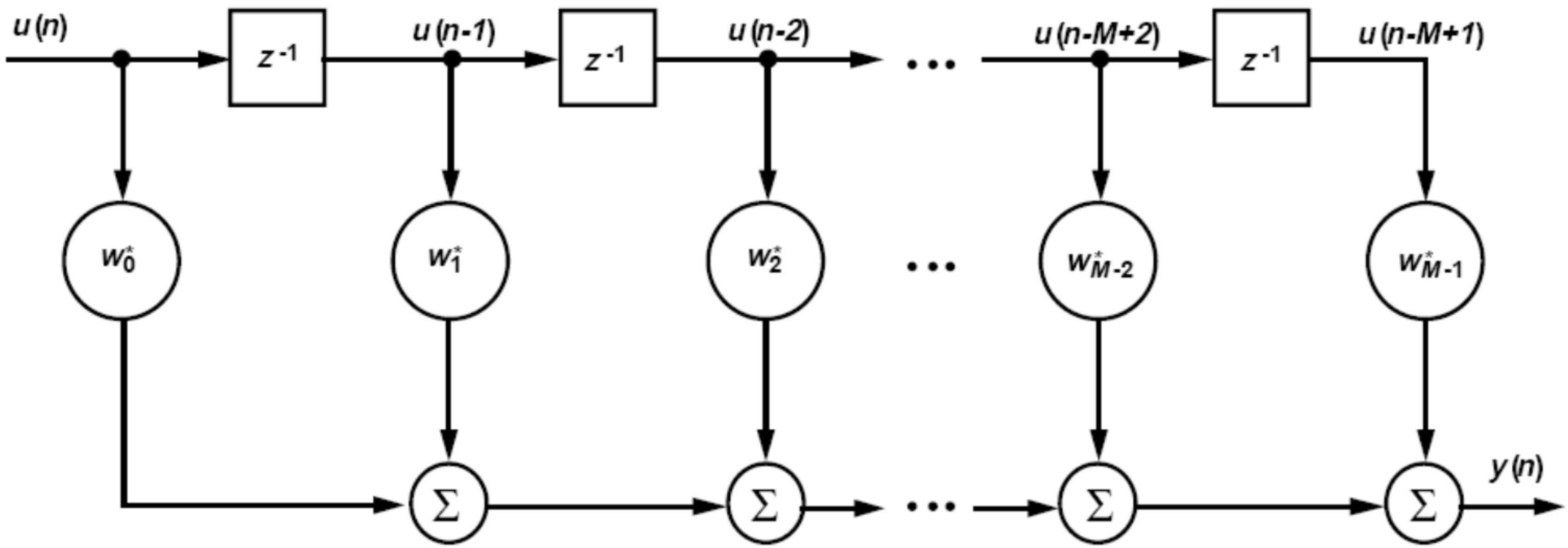
Linear Filter Structures

- The operation of a linear adaptive filtering algorithm involves two basic processes:
  - a ***filtering process*** designed to produce an output in response to a sequence of input data
  - an ***adaptive process***, the purpose of which is to provide mechanism for the *adaptive control* of an *adjustable* set of parameters used in the filtering process.
- These two processes work interactively with each other.

Linear Filter Structures

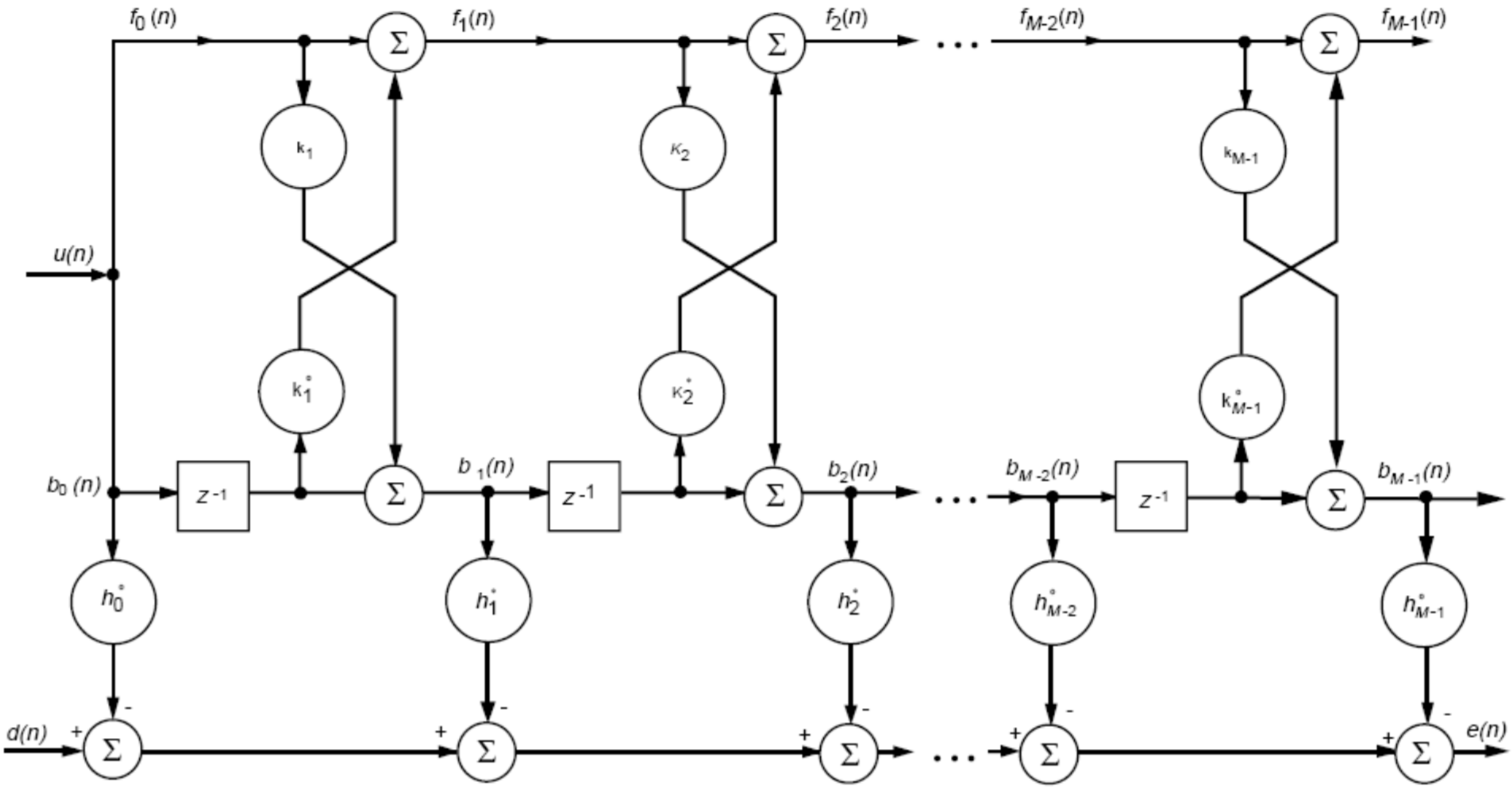
- The impulse response of a linear filter determines the memory order of the filter:
  - Infinite impulse response (IIR) filter
  - Finite impulse response (FIR) filter
- There are three types of filter structures with *finite memory :*
  - transversal filter,
  - lattice predictor,
  - and systolic array.

Transversal or Tapped Delay Line Filter



$$y(n) = \sum_{k=0}^{m-1} w_k^* u(n-k)$$

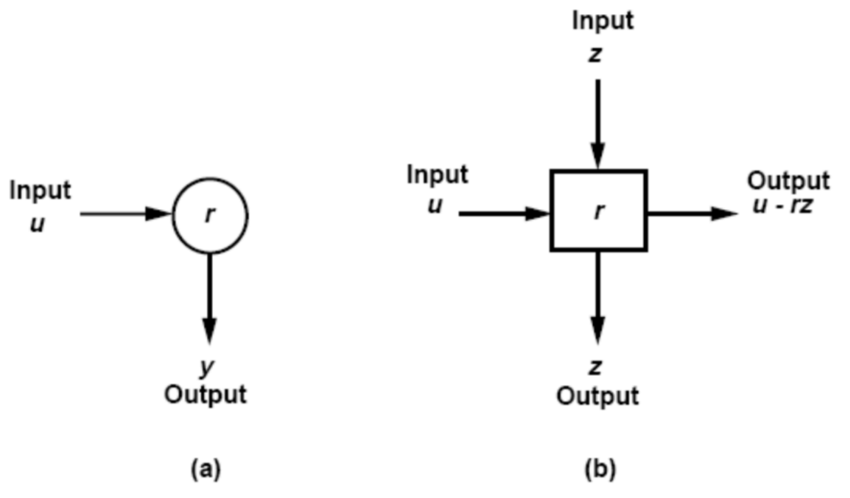
Lattice Predictor



It has the advantage of simplifying the computation

Systolic Array

Represents a parallel computing network ideally suited for mapping a number of linear algebra computations, e.g. matrix multiplication, triangularization, back substitution, etc.

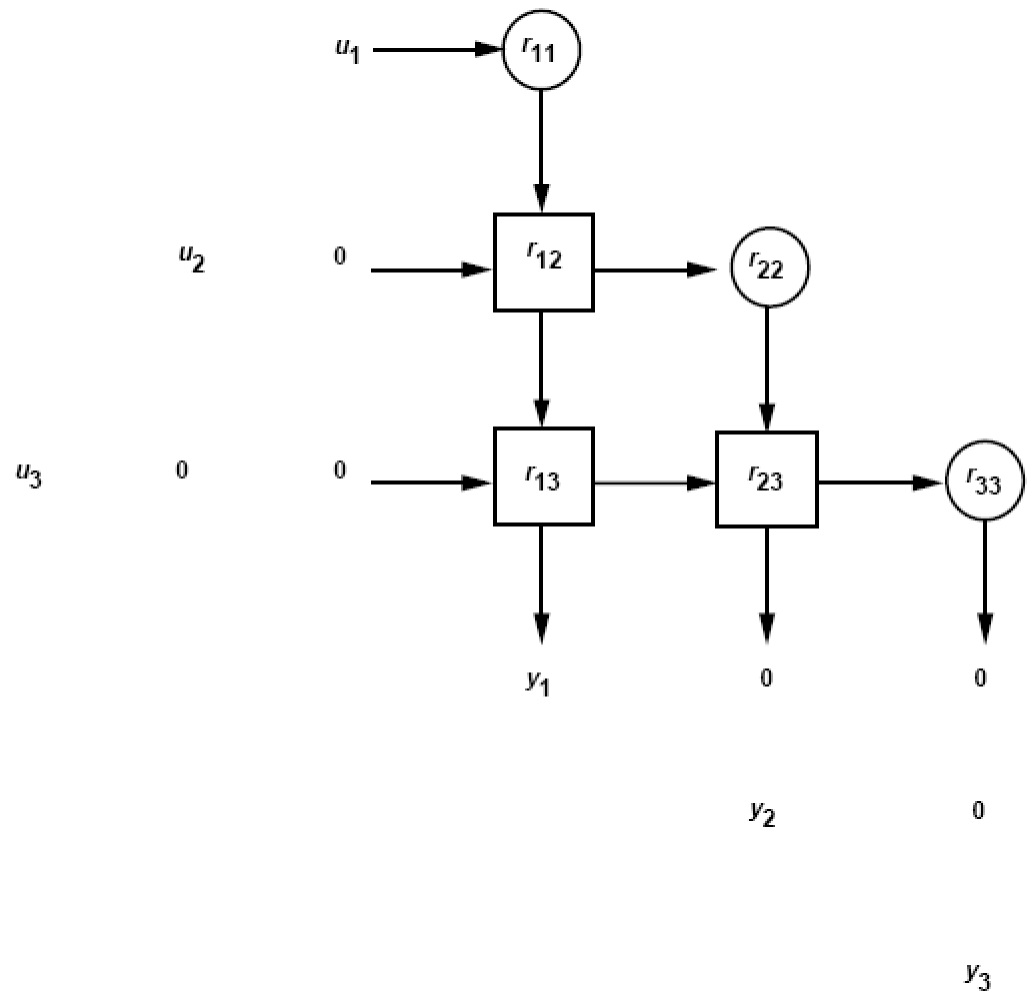


Two basic processing elements:

(a) Boundary cell and (b) Internal cell

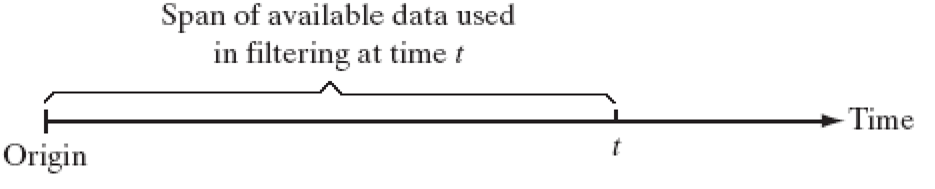
Triangular Systolic Array

The use of systolic arrays has made it possible to achieve very high throughput, which is required for many advanced signal processing algorithms to operate in *real time*

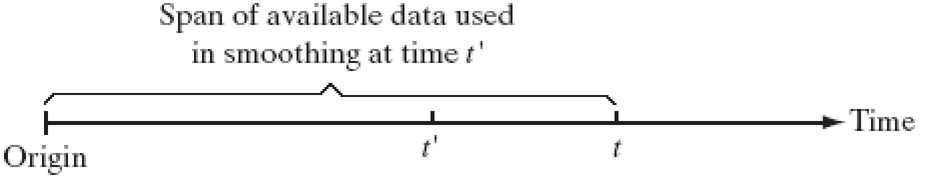


The three basic forms of estimation

- The function of receiver is to operate on the received signal and deliver a reliable estimate of the original message.
- Filters may be used for the purpose of estimation using one of the following three information-processing operations.
  - Filtering

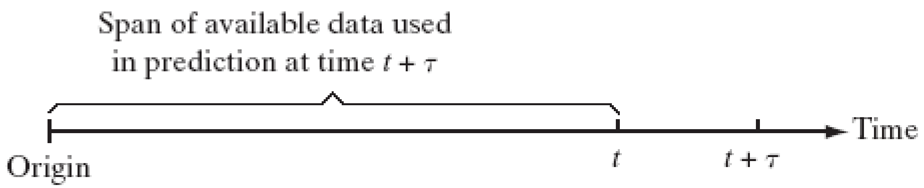


  - Smoothing



The three basic forms of estimation

- Prediction



- Given an optimality criterion we can design optimal filters
  - Requires a priori information about the environment
  - Example: Under certain conditions the so called Wiener filter is optimal in the mean-squared sense
- Adaptive filters are self-designing using a recursive algorithm
  - Useful if complete knowledge of environment is not available a priori.

Linear Filters

- For stationary inputs, the resulting solution is commonly known as the ***Wiener filter***, which is said to be optimum in the *mean-square* sense.
- A plot of the mean-square value of the error signal vs. the adjustable parameters of a linear filter is referred to as the *error-performance surface.*
- The minimum point of this surface represents the Wiener solution.

Linear Filters

- The Wiener filter is inadequate for dealing with situations in which *non-stationarity* of the signal and/or noise is intrinsic to the problem.
- A highly successful solution to this more difficult problem is found in the ***Kalman filter***, a powerful device with a wide variety of engineering applications.

Linear Adaptive Filtering Algorithms

- **Stochastic Gradient Approach**
  - *Least-Mean-Square* (*LMS*) *algorithm*
  - *Gradient Adaptive Lattice* (*GAL*) *algorithm*

$$\begin{pmatrix} \text{updated value} \\ \text{of tap-weight} \\ \text{vector} \end{pmatrix} = \begin{pmatrix} \text{old value} \\ \text{of tap-weight} \\ \text{vector} \end{pmatrix} + \begin{pmatrix} \text{learning-} \\ \text{rate} \\ \text{parameter} \end{pmatrix} \begin{pmatrix} \text{tap-} \\ \text{input} \\ \text{vector} \end{pmatrix} \begin{pmatrix} \text{error} \\ \text{signal} \end{pmatrix}$$

Linear Adaptive Filtering Algorithms

- **Least-Squares Estimation**
  - *Recursive least-squares* (*RLS*) *estimation*
    - *Standard RLS algorithm*
    - *Square-root RLS algorithms*
    - *Fast RLS algorithms*

$$\begin{pmatrix}\text{updated value}\\ \text{of the}\\ \text{state}\end{pmatrix} = \begin{pmatrix}\text{old value}\\ \text{of the}\\ \text{state}\end{pmatrix} + \begin{pmatrix}\text{Kalman}\\ \text{gain}\end{pmatrix}\begin{pmatrix}\text{innovation}\\ \text{vector}\end{pmatrix}$$

Stochastic Gradient Approach

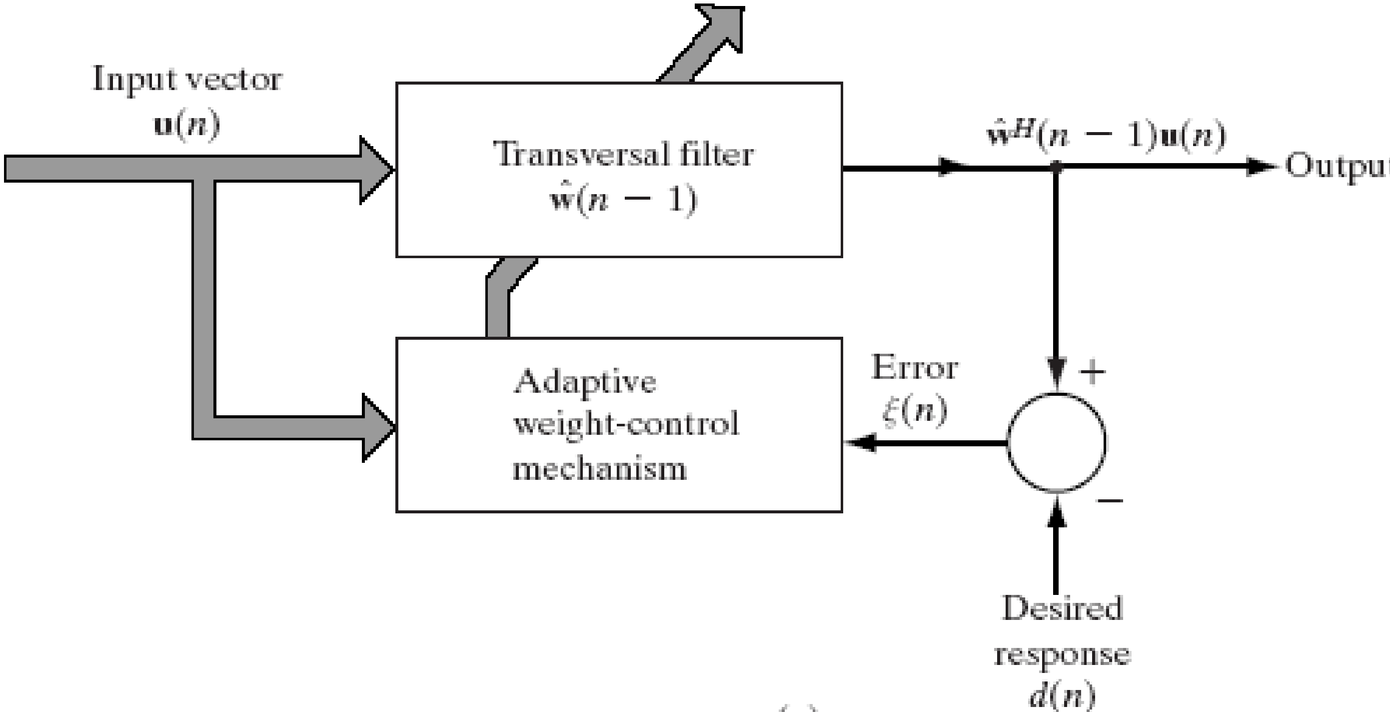
- Most commonly used type of Adaptive Filters
- Define cost function as mean-squared error
  - Difference between filter output and desired response
- Based on the method of steepest descent
  - Move towards the minimum on the error surface to get to minimum
  - Requires the gradient of the error surface to be known

Least-Mean-Square (LMS) Algorithm

- Most popular adaptation algorithm is LMS
  - Derived from steepest descent
  - Doesn't require gradient to be known: it is estimated at every iteration
- Consists of two basic processes
  - Filtering process
    - Calculate the output of FIR filter by convolving input and taps
    - Calculate estimation error by comparing the output to desired signal

Least-Mean-Square (LMS) Algorithm

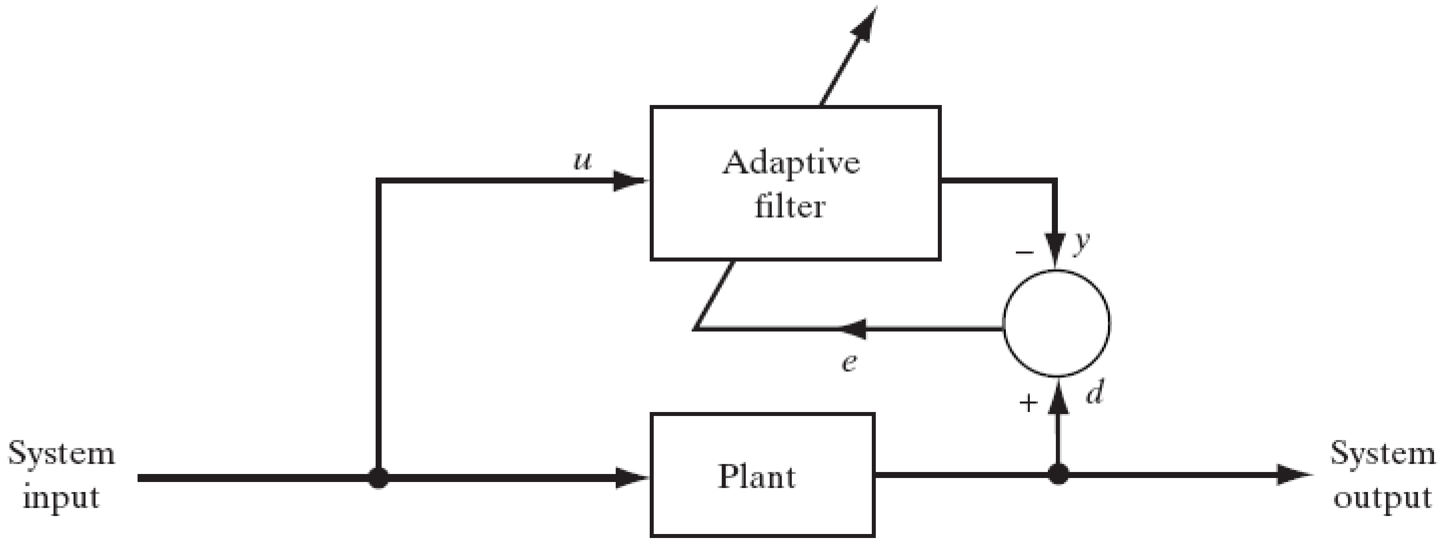
- Adaptation process
  - Adjust tap weights based on the estimation error



(a)

Four classes of applications

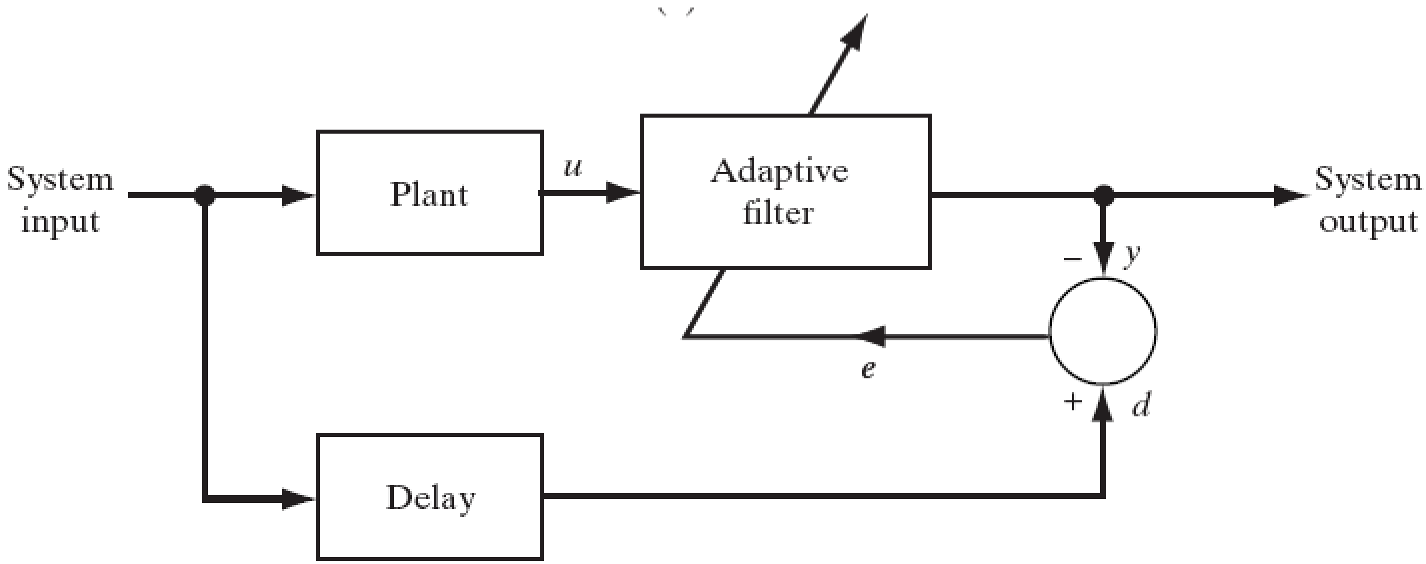
- Used to provide a linear model of an unknown plant



- Applications:
  - System identification

Four classes of applications

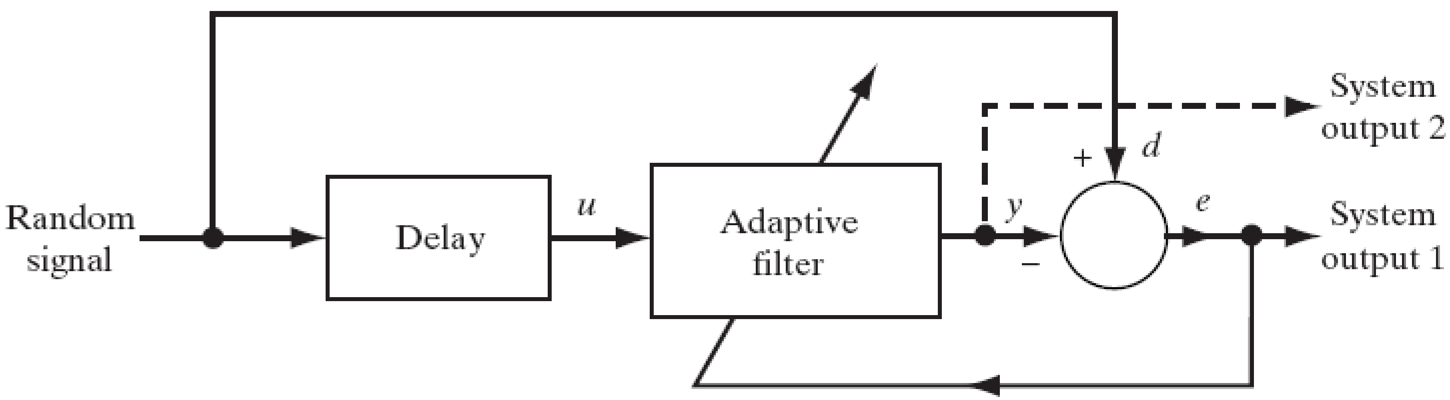
- Used to provide an inverse model of an unknown plant



- Applications:
  - Equalization

Four classes of applications

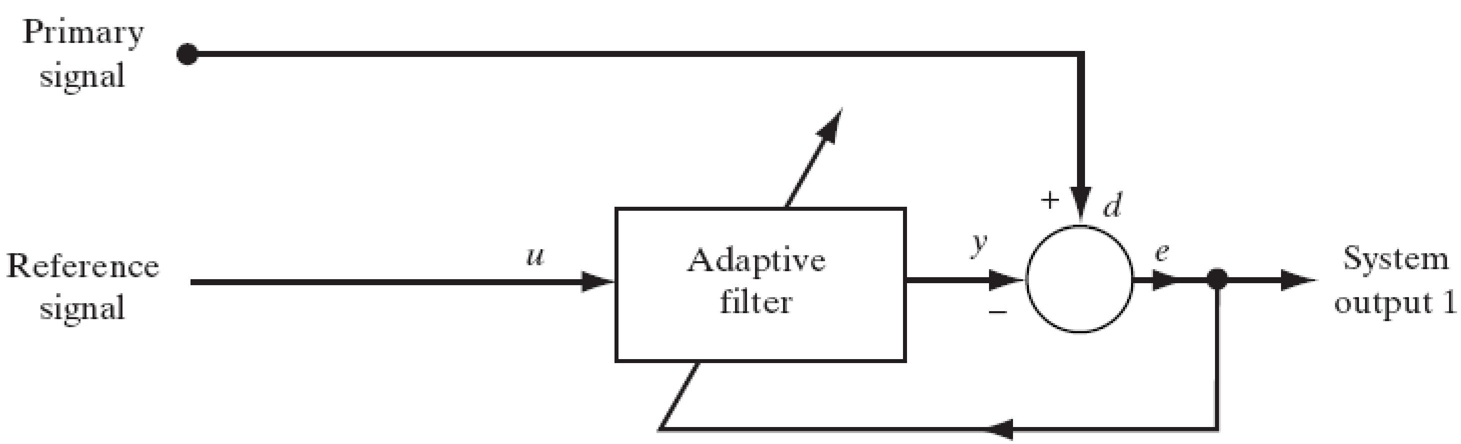
- Used to provide a prediction of the present value of a random signal



- Applications:
  - Linear predictive coding

Four classes of applications

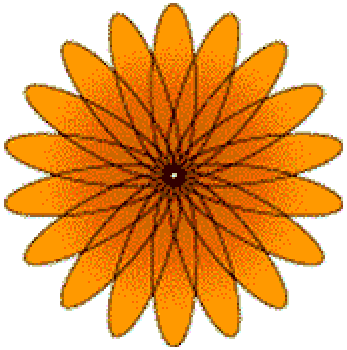
- Used to cancel unknown interference from a primary signal



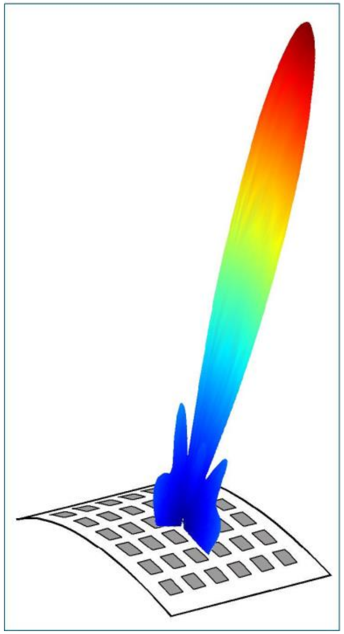
- Applications:
  - Echo cancellation

Beamforming

- **Smart Antenna systems**:
  - Switched beam: finite number of fixed predefined patterns



  - Adaptive array: Infinite number of (real time) adjustable patterns



Beamforming

- **Adaptive filtering**: An array of independent sensors placed at different locations in space essentially samples the received signal spatially.
- Applications: Radar, Sonar, Speech enhancement, etc.
- **Smart Antenna systems**: Using a variety of new signal-processing algorithms, the adaptive system takes advantage of its ability to effectively locate and track various types of signals to dynamically minimize interference and maximize intended signal reception.

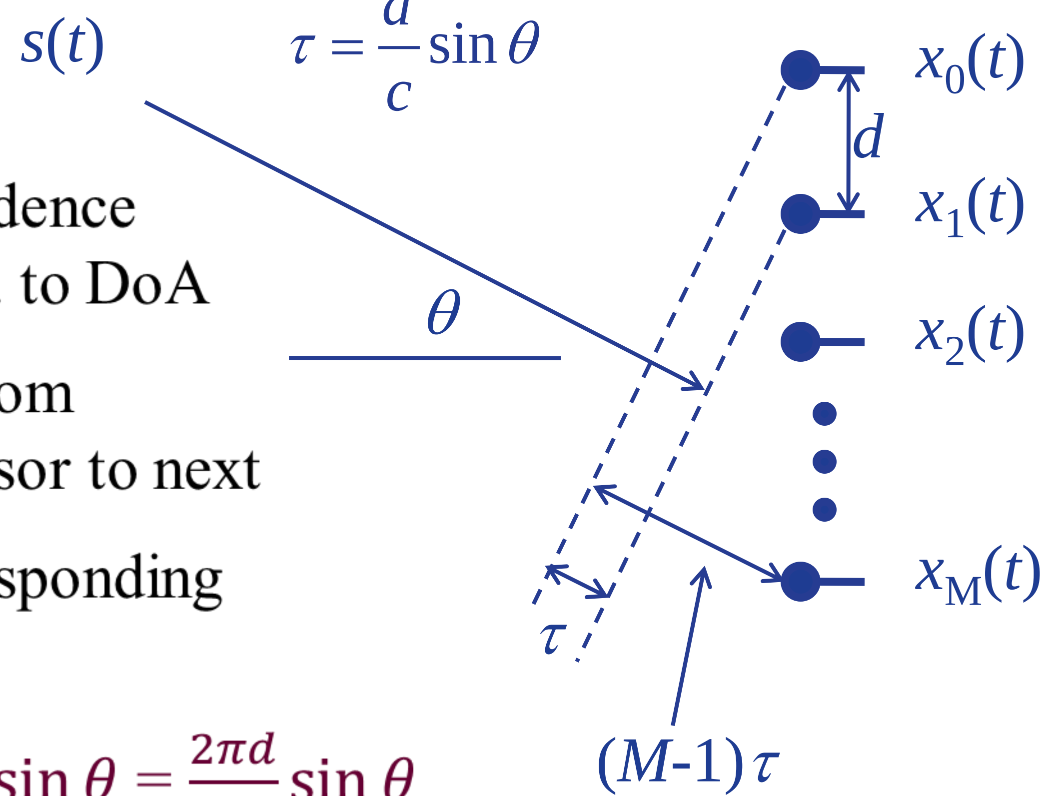
Beamforming

- **Beamforming:** spatial filtering
  - The sensor outputs are individually weighted and then summed to produce overall beamformer output.
- The term derived from the fact that early antennas were designed to form a pencil like beam so as to receive signal radiating from a specific direction and to attenuate signals originating from other directions that are of no interest.
- Beamforming applies to both transmission as well as reception.

Beamforming: Uniform line array

$$x_m(t) = s(t - m\tau)$$

$$\tau = \frac{d}{c}\sin\theta$$



θ = angle of incidence corresp. to DoA

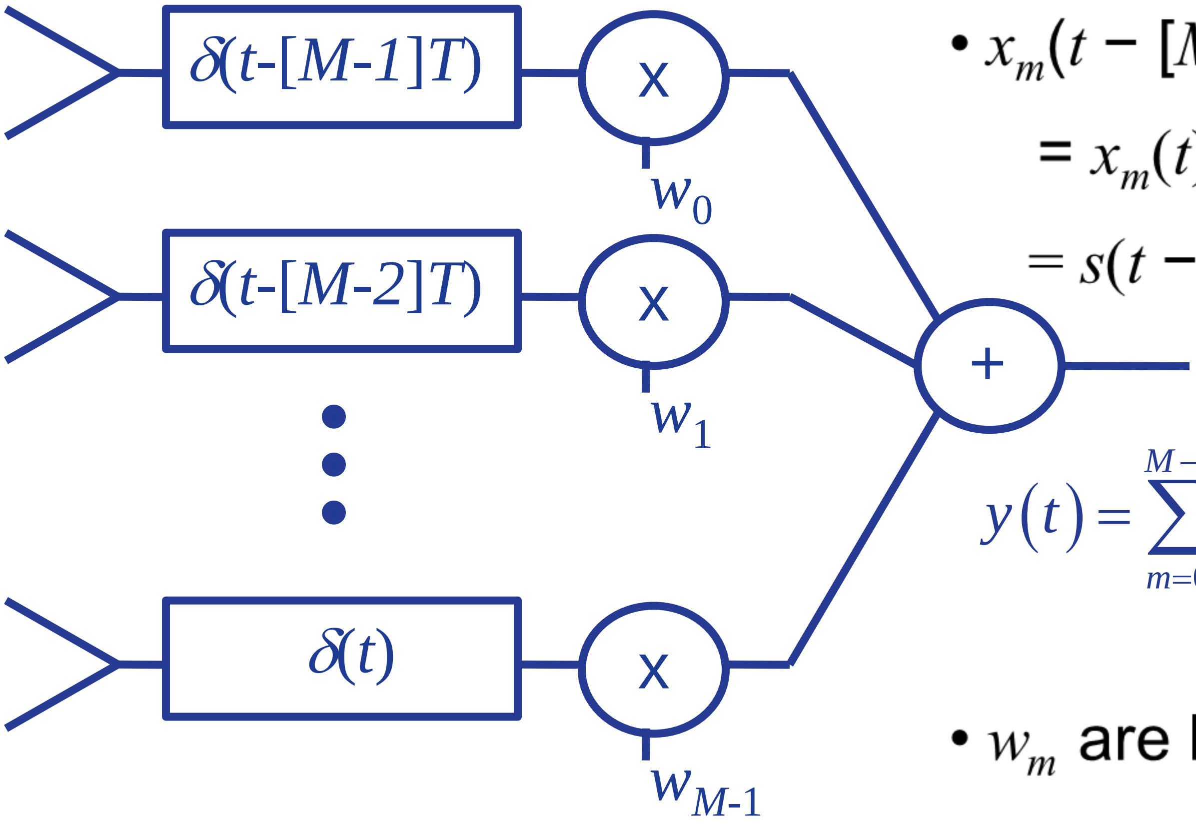
τ = time delay from one sensor to next

Therefore, corresponding phase delay

$$\xi = \frac{2\pi f d}{c}\sin\theta = \frac{2\pi d}{\lambda}\sin\theta$$

Beamforming: Delay-sum

Timed array



- When $T = \tau$, the channels are all time aligned
- $x_m(t - [M - m - 1]T)$
  $= x_m(t) * \delta(t - [M - m - 1]T)$
  $= s(t - [M - 1]T), \forall m$

$$y(t) = \sum_{m=0}^{M-1} w_m x_m\left(t - [M - m - 1]T\right)$$

- w_m are beamformer weights

Beamforming: Delay-sum

- $Y(\omega) = \sum_{m=0}^{M-1} w_m X_m(\omega) e^{-j\omega[M-m-1]\tau}$

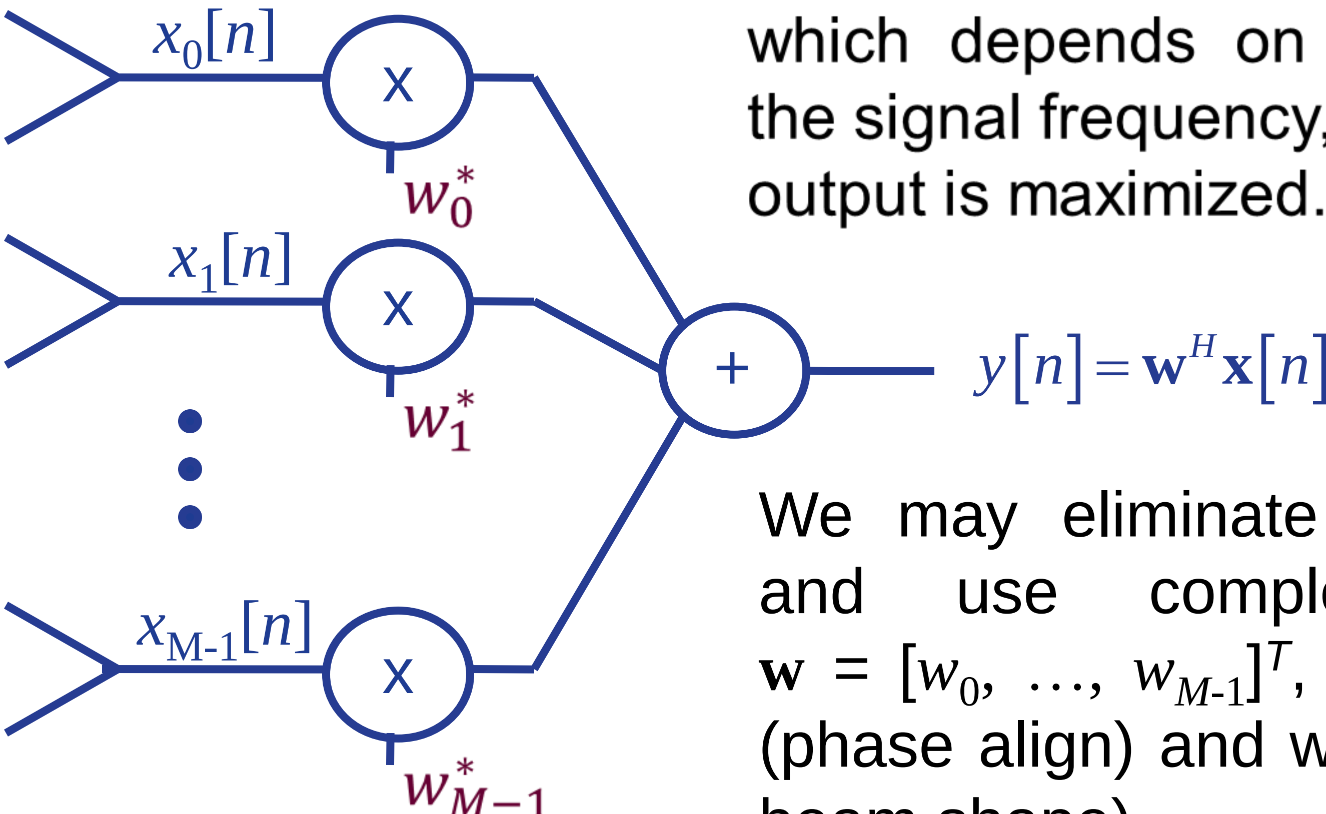
$$= \sum_{m=0}^{M-1} w_m S(\omega) e^{-j\omega m\tau} e^{-j\omega[M-m-1]\tau}$$

$$= S(\omega) \sum_{m=0}^{M-1} w_m e^{-j\omega[M-1]\tau} = S(\omega) \sum_{m=0}^{M-1} \alpha_m$$

- So, required frequency response of every channel in the beamformer:

$$W_m(\omega) = w_m\, e^{-j\omega[M-1]\tau} e^{j\omega m\tau} = \alpha_m\, e^{jm\xi}$$

Beamforming: Narrowband phased array

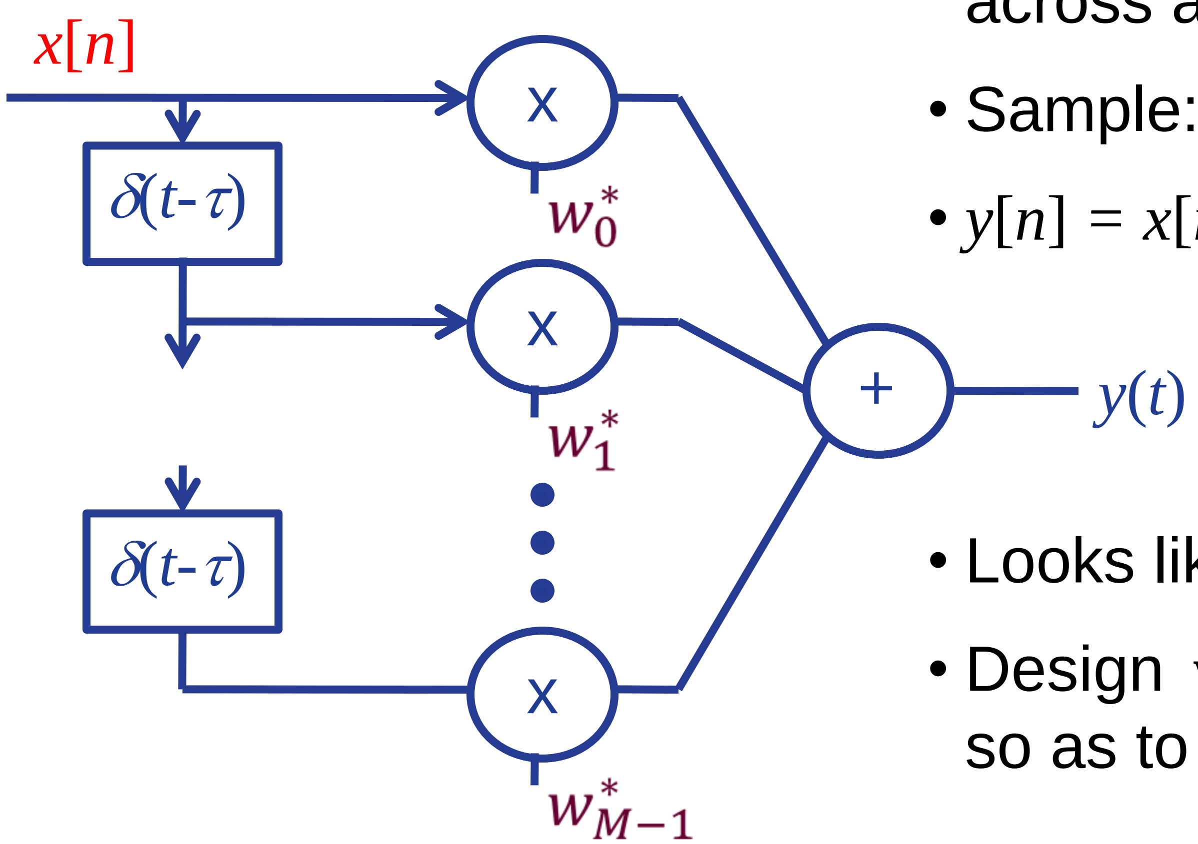


We are now interested in finding optimal $\alpha_m e^{jm\xi}$ for every channel, which depends on the DoA and the signal frequency, such that the output is maximized.

We may eliminate time delays and use complex weights, $\mathbf{w} = [w_0, \ldots, w_{M\text{-}1}]^T$, to both steer (phase align) and weight (control beam shape)

Beamforming: Similar to FIR filter

Tapped delay line



- Represent signal delay across array as a delay line
- Sample: $x[n] = x_0(nT) = s(nT)$
- $y[n] = x[n]*w[n]$
- Looks like an FIR filter.
- Design $\mathbf{w}$ with FIR methods so as to maximize the output

Beamforming: Narrowband

- Narrowband assumption: If BW of $s(t) << c\,/\,(M\text{-}1)d$ Hz.
- This means the phase difference between upper and lower band edges for propagation across the entire array is small.
- Most communication signals fit this model.
- Check that d must be less than half-wavelength (λ/2); this ensures one-to-one correspondence between θ (lying in the range $[-\pi/2, +\pi/2]$) and ξ (lying in the range $[-\pi, +\pi]$).
- If signal is not narrowband, bandpass filter it and build a new beamformer for each subband.

Beamforming

- Consider *s*(*t*) be complex sinusoid $s(t) = e^{j\omega t}$.
- Then, $x_m[n] = x_m(nT) = s(nT - m\tau)$
  $$= e^{j\omega nT} e^{-j\omega m\tau} = e^{-j\omega m\tau} s(nT) = e^{-jm\xi} s[n]$$
- $\mathbf{x}[n] = [\, x_0[n] \quad x_1[n] \quad \cdots \quad x_{M-1}[n] \,]^T$
  $$= [1 \quad e^{-j\xi} \quad \dots \quad e^{-j(M-1)\xi} \,]^T s[n]$$
- $y[n] = \mathbf{w}^H \mathbf{x}[n]$
  $$= [w_0^* \quad w_1^* \quad \cdots \quad w_{M-1}^*][1 \quad e^{-j\xi} \quad \dots \quad e^{-j(M-1)\xi} \,]^T s[n]$$

Beamforming

- The vector $\mathbf{a}(\theta) = [1 \quad e^{-j\xi} \quad \cdots \quad e^{-j(M-1)\xi}\,]^T$ is called steering vector, depends on the angle of arrival of each incident signal.
- $y[n] = \mathbf{w}^H \mathbf{x}[n] = \mathbf{w}^H \mathbf{a}(\theta) s[n]$
- If the signal of interest (SOI) is incident in the direction of θ then we wish that
  - *the desired signal is received without any modification, i.e.* $\mathbf{w}^H \mathbf{a}(\theta) = 1$
  - *the interference signals are rejected, i.e.* $\mathbf{w}^H \mathbf{a}(\phi) = 0$

Beamforming

- An optimal weight vector satisfying our requirement is of the form

$$\mathbf{w} = \left[\alpha_0, \alpha_1 e^{-j\zeta}, \cdots, \alpha_{M-1} e^{-j(M-1)\zeta}\right]^T,$$

$$\zeta = \frac{2\pi f_0 d}{c} \sin\theta_0$$

α_m = amplitude weight for sensor m,

$m\xi$ = phase weight for sensor m,

f_0 = bandpass center frequency, θ_0 = direction of max response

- Amplitude components control the sidelobe level and main beam width.

Beamforming

- Phase components control the angle of the main beam and nulls.
- Phase weights for narrowband arrays are applied by a phase shifter
- Gain in direction θ is $\sum_{m=0}^{M-1} \alpha_m$. Less in other directions due to incoherent addition.
- The values of amplitude weights are chosen such that $\sum_{m=0}^{M-1} \alpha_m=1$.

Beamforming

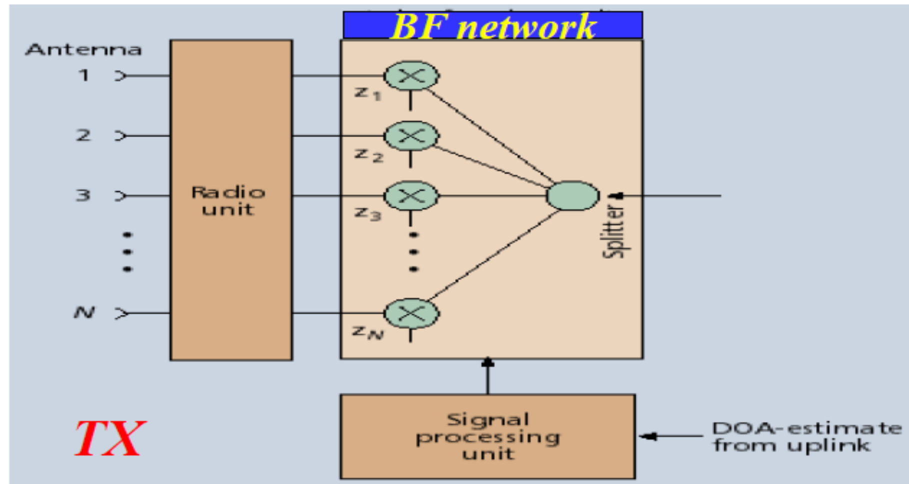
- Thus, we define beamforming as
  - Beam forming is a method used to create the radiation pattern of an array antenna by adding constructively the weights of the signals in the direction of SOI and nulling the pattern in the direction of SNOI (interference)
- This array can be antennas in the smart antennas context, or any other types of sensors (radars, medical sensors, …, etc), can be an array of microphones in the speech signal processing context.

Beamforming

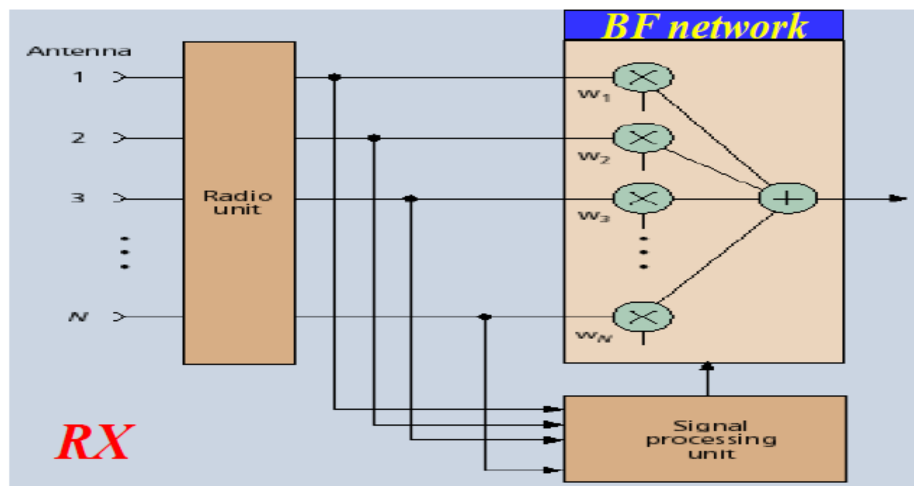
- Beamforming can be used at both the transmitting and receiving ends in order to achieve spatial selectivity
- That is, an appropriate feeding allows antenna arrays to steer their beam and nulls towards certain directions, this is often referred to as spatial filtering.

Beamforming

In transmission mode , the majority of signal energy transmitted from a group of sensor array can be directed in a chosen angular direction



In reception mode, you can calibrate your group of sensors when receiving signals such that you predominantly receive from a chosen angular direction



Factors for choice of filter / algorithm

- Choice of Adaptive filters:
  - Computational cost
  - Performance
  - Robustness
- Choice of recursive algorithm:
  - Rate of convergence
  - Misadjustment
  - Tracking

Factors for choice of filter / algorithm

- Robustness
- Computational requirements
- Structure
- Numerical properties