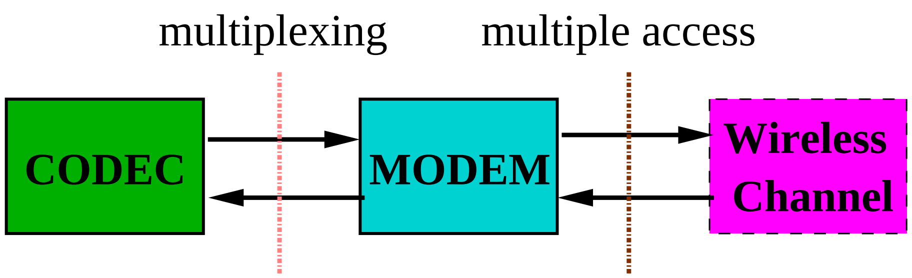
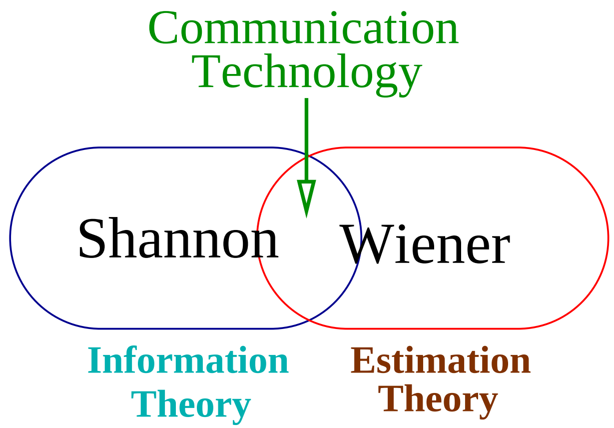
Motivations of Signal Processing

- Previous four lectures introduce basic concepts of **channel coding** and discuss two most widely used channel coding methods, convolutional codes and BCH codes
  - Vital have a deep understand of these essential concepts and practical coding/decoding methods
  - Appreciation of **SISO** **iterative** approach, a generic principle in many state-of-the-arts
- We now come back to Modem, deal with **frequency selective channels**
  - **Adaptive** signal processing, in particular, equalisation, are not just for combating ISI, also for combating multiple access interference



- Communication technology is where Shannon meets Wiener
  - You all know how important **information theory**, largely due to Shannon, is to communications
  - **Estimation theory**, started with Wiener, is also crucial to communications
  - Communication is truly where information theory meets estimation theory



- Information theory deals with performance bounds and provides foundations and principles, while estimation theory offers means of realising and implementing designs and systems
  - Adaptive signal processing is **practice** of estimation theory

Channel Equalisation: Introduction

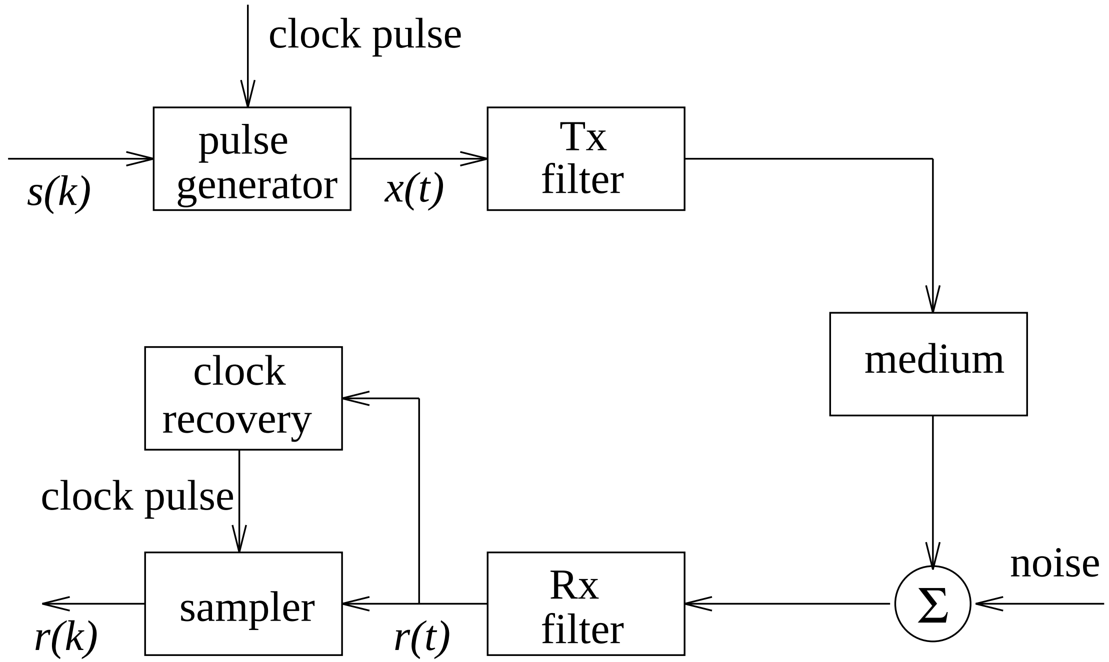
- Due to a restricted bandwidth and/or multipath, a wideband channel introduces ISI, and an **equaliser** is required at receiver to **overcome ISI distortion**
  - Since channel $G_c(f)$ is non-ideal, combined channel and transmit/receive filter $G_{\mathrm{tot}}(f) = G_{\mathrm{R}}(f)G_c(f)G_{\mathrm{T}}(f)$ is no longer a Nyquist filter
- The equaliser $H(f)$ should make $G_{\mathrm{tot}}(f)H(f)$ a Nyquist system again. In RF passband or frequency-domain equalisation, this is very difficult to achieve
  - However, for digital communication, equalisation can be achieved at baseband with sampled receive signal, and this is much easier
- As the channel also introduces AWGN, the equaliser also need to take into account this noise and does not **enhance the noise** in its operation
  - To remove ISI completely, the baseband equaliser $H(z)$ should be an inverse of the channel $G_{\mathrm{tot}}(z)$ but this may amplify the noise too much
  - So equaliser design **trades off** eliminating ISI and enhancing noise

Digital Baseband Channel Model

- Assume correct carrier recovery and synchronisation as well as complex-valued channel and modulation scheme (with real-valued as special case)

  – Discrete-time channel model

$$r(k) = \sum_{i=0}^{n_c} c_i s(k-i) + n(k)$$



  – N-QAM symbols

$$s(k) \in \{s_{i,l} = u_i + \mathrm{j} u_l,\ 1 \le i,l \le \sqrt{N}\}$$

  – AWGN $n(k)$: $E[|n(k)|^2] = 2\sigma_n^2$

- Note ISI: symbols transmitted at different symbol instances are mixed

  – Also, **channel** acts like an "**encoder**" with memory length $n_c + 1$ and non-binary complex-valued weights → compare it with CC encoder

Equaliser Classification

- Training-based and blind equalisers:
  - **Training**: during link initialisation/set up, a prefixed sequence $\{s(k)\}$ known to receiver is sent and receiver generates this sequence locally which together with received signal $\{r(k)\}$ are used to either identify channel $\{c_i\}$ and/or adjust equaliser's parameters
  - If channel is time-varying, a periodic training is needed, and transmitted symbols are organised into frames with middle part of each frame allocated to training sequence

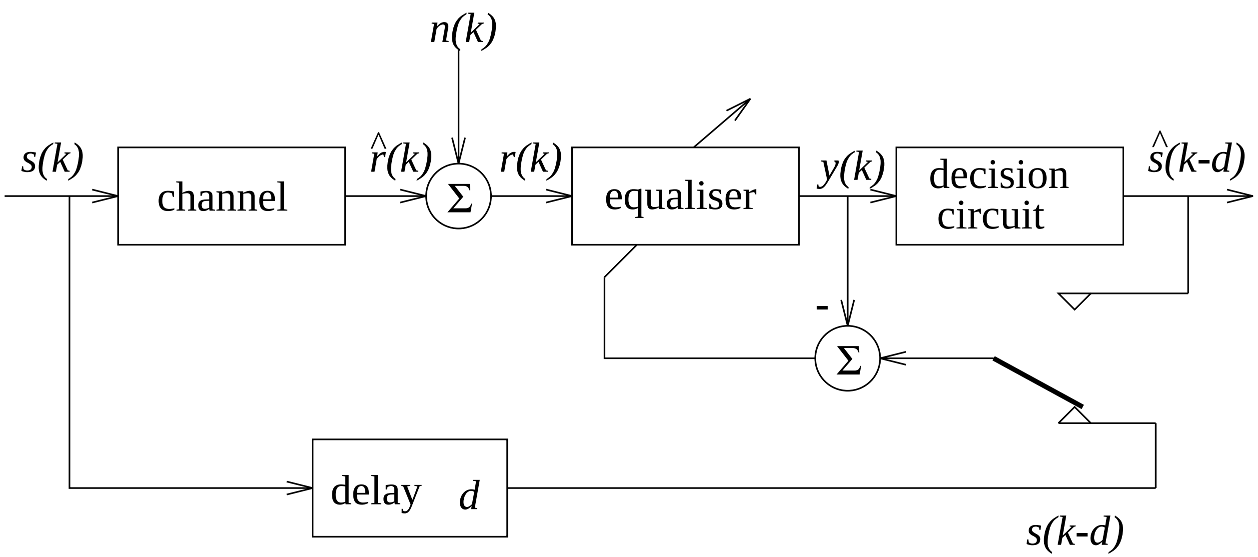
data	training	data

Frame structure

  - **Blind**: training costs extra bandwidth, also for multi-point communications, e.g. digital TV, training is impossible
  - Equaliser has to figure out the channel and/or adjust its parameters based on the received $\{r(k)\}$ only and some known statistics of transmitted symbols
- Sequence-decision and symbol-decision equalisers:
  - **Sequence estimation**: estimate entire transmitted sequence, generally optimal but for long channel and high-order N, complexity is often too much
  - Maximum likelihood sequence estimation with Viterbi algorithm is optimal and widely used (Mobile handset may have two Viterbi algorithms, one for equaliser and one for channel coding)
  - **Symbol estimation**: at each k estimate a symbol transmitted at $k - d$, such as linear equaliser and decision feedback equaliser

Adaptive Equalisation Structure

- The general framework with two operation modes
  - **Training mode**: During training, equaliser has access to the transmitted (training) symbols $s(k)$ and can use them as the desired response to adapt the equaliser's coefficients

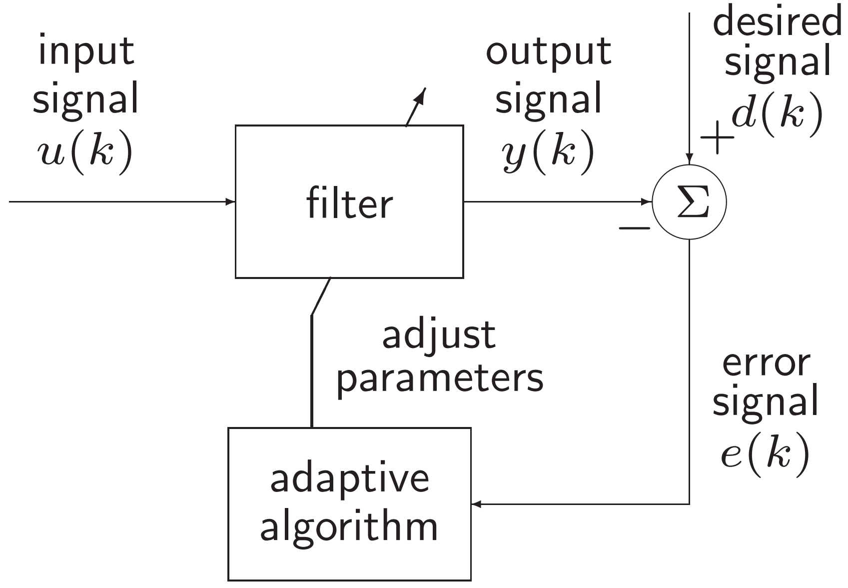


  - **Decision-directed mode**: During data communication phase, equaliser's decisions $\hat{s}(k-d)$ are assumed to be correct and are used to substitute for $s(k-d)$ as the desired response to continuously track a time-varying channel

- At sample k, the equaliser detects the transmitted symbol $s(k-d)$, not the current symbol $s(k)$. This **decision delay** d is necessary for a **nonminimum phase** channel
  - Equaliser $H_E(z)$ attempts to inverse the channel $H_C(z)$. If $H_C(z)$ is nonminimum phase, its causal inverse is unstable
  - The best can be done is to truncate the anticausal inverse of $H_C(z)$ and to delay the resulting transfer function to obtain a causal $H_E(z)$ such that $H_C(z)H_E(z) \approx z^{-d}$
  - d related to number of zeros outside unit circle, and for minimum phase, $d = 0$

General Structure of Adaptive Filter

- Adaptive equaliser is an example of general **adaptive filter**, whose structure is
  - **Communication** is enabling technology for our information society
  - **Adaptive signal processing** is enabling technology for communication
  - We therefore pay a visit to adaptive filter which is based on estimation theory



- An **adaptive algorithm** adjusts the filter parameters involving **error signal** $e(k) = d(k) - y(k)$ so that
  - **Filter output** $y(k)$ matches **desired output** $d(k)$ as close as possible in some statistic sense

- Some important issues: rate of convergence, misadjustment, tracking, robustness, computational requirements, and structure

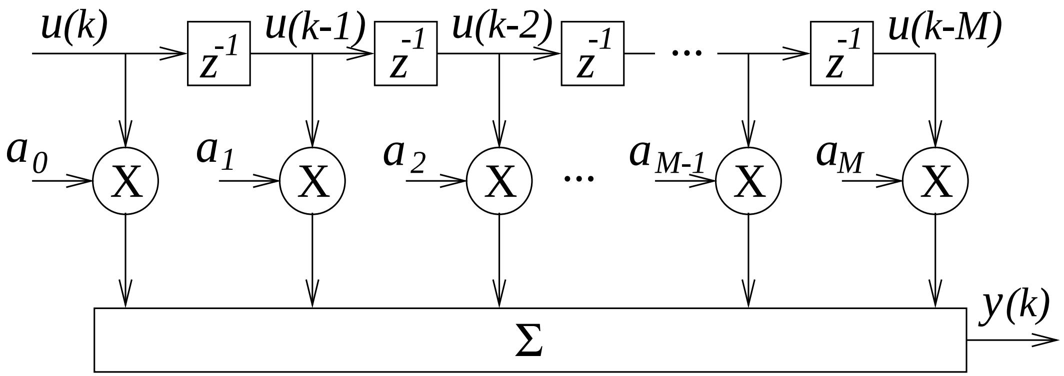
Tap-Delay-Line Filter

- The simplest linear filter structure is the tap-delay-line or transversal filter with transfer function:

$$H(z) = \sum_{i=0}^{M} a_i z^{-i}$$

and filter output given by:

$$y(k) = \sum_{i=0}^{M} a_i u(k-i)$$



- This is an **FIR filter**, $H(z)$ has no poles and is inherently stable, and the mean square error $E[|e(k)|^2]$ has a single global minimum for $\mathbf{a} = [a_0 \ a_1 \cdots a_M]^T$

- Minimum phase: all zeros of $H(z)$ are inside unit circle $|z| = 1$ of z-plane; and nonminimum phase: otherwise

- A drawback is that an FIR filter may require large number of coefficients (large order M) in some applications

Recurrent Filter

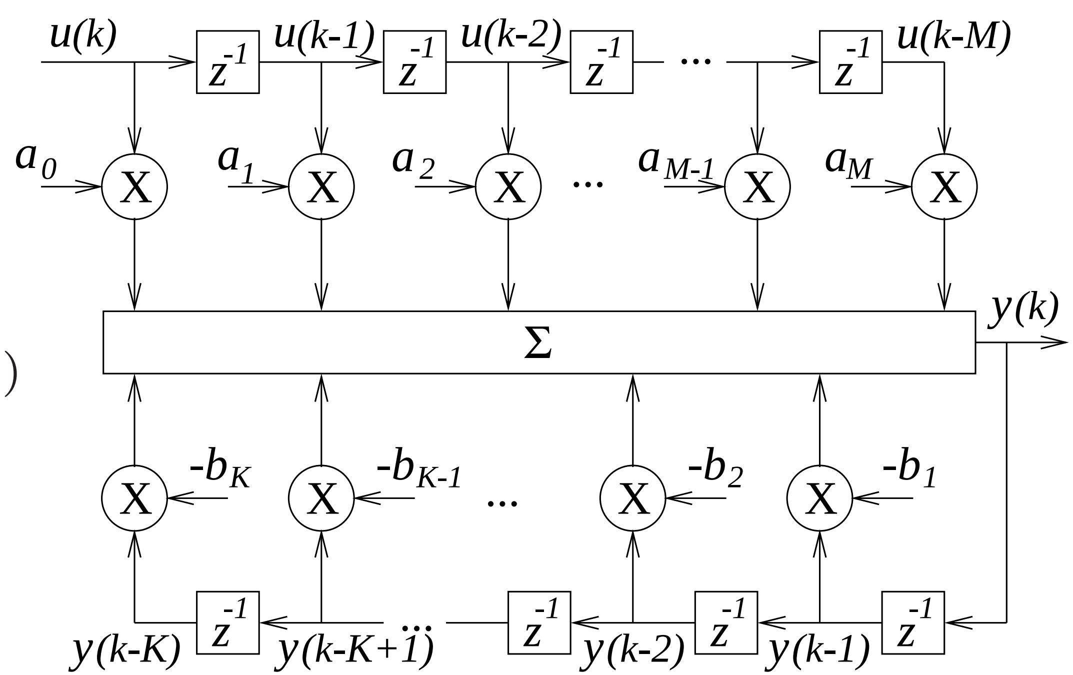
- A much more complicated linear filter is the **recurrent** or **ARMA** filter with transfer function:

$$H(z) = \frac{\sum_{i=0}^{M} a_i z^{-i}}{1 + \sum_{i=1}^{K} b_i z^{-i}}$$

and filter output given by:

$$y(k) + \sum_{i=1}^{K} b_i y(k-i) = \sum_{i=0}^{M} a_i u(k-i)$$

This is an example of **IIR filter**



- More efficient in terms of number of coefficients required for many problems

- To be stable, all **poles** of $H(z)$ must be inside $|z| = 1$. Also the mean square error may have many local/global minimum solutions for $\mathbf{w} = [a_0\ a_1 \cdots a_M\ b_1\ \cdots b_K]^T$

Optimisation

- Filter design is an **optimisation** problem: adjust the filter coefficient vector $\mathbf{w}$ to minimise some **cost function**

- Typical cost function in filter design optimisation is the **mean square error**:

$$J(\mathbf{w}) = \mathrm{E}[|e(k)|^2]$$

where the error signal $e(k) = d(k) - y(k)$ is the difference between the desired filter response and actual filter response and $E[\cdot]$ denotes ensemble average

- **Gradient** of the cost function with respect to the parameter vector plays a central role in optimisation

Let $\mathbf{w} = [w_1 \cdots w_{N_w}]$. The gradient of the MSE with respect to $\mathbf{w}$ is defined by

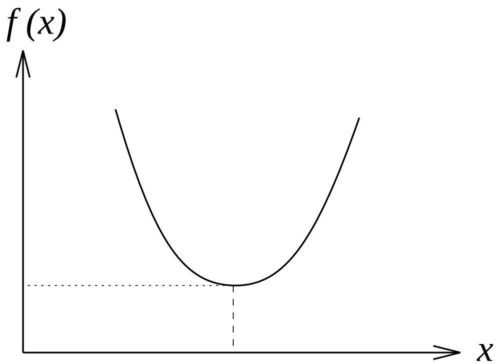
$$\nabla J(\mathbf{w}) = \left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right]^T \quad \text{with} \quad \text{derivative} \quad \left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right] = \left[\frac{\partial J}{\partial w_1} \cdots \frac{\partial J}{\partial w_{N_w}}\right]$$

Minimum of Cost Function

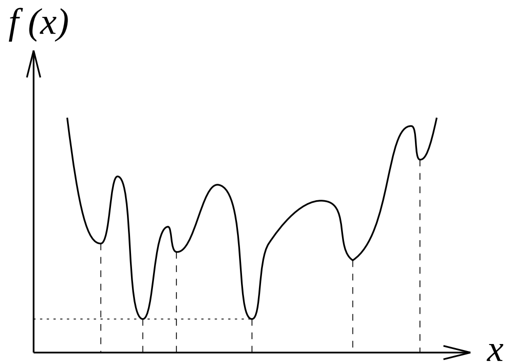
- For cost function of scalar variable $f(x)$, conditions for x to be a minimum are:

$$\frac{\partial f(x)}{\partial x} = 0 \quad (\text{necessary})$$

$$\frac{\partial^2 f(x)}{\partial x^2} > 0 \quad (\text{sufficient})$$



single global minimum



many local/global minima

- The MSE $J(\mathbf{w})$ can be viewed as an error-performance surface on the $\mathbf{w}$ space, and conditions for $\mathbf{w}$ to be a minimum of $J(\mathbf{w})$ are:

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0 \quad (\text{necessary}) \qquad \frac{\partial^2 J(\mathbf{w})}{\partial \mathbf{w}^2} \text{ is positive definite} \quad (\text{sufficient})$$

- For FIR, $J(\mathbf{w})$ has a single global minimum and for IIR, $J(\mathbf{w})$ may have many local/global minima
- $J(\mathbf{w})$ is probabilistic, and a time-average cost function over N samples is often used instead

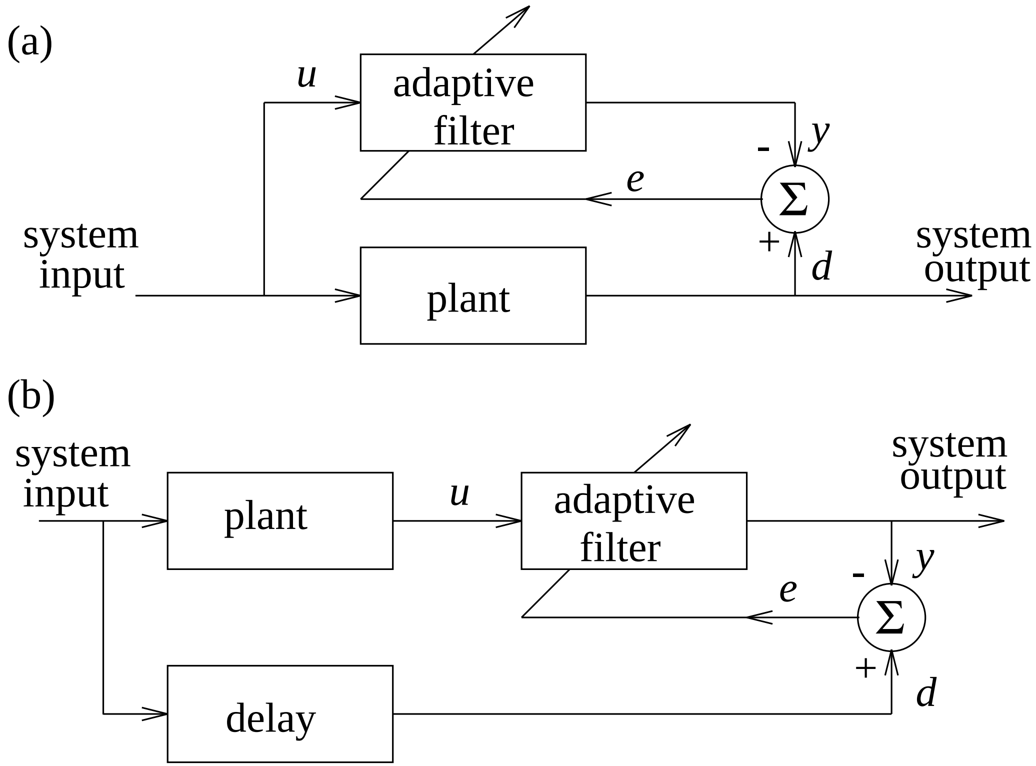
$$J_N(\mathbf{w}) = \sum_{k=1}^{N} |e(k)|^2$$

Classical Applications

(A) **Identification**: Adaptive filter provides a linear model to an unknown noisy plant

The plant and the adaptive filter are driven by the same input, and the noisy plant output supplies the desired output for the adaptive filter

An example is identifying an FIR channel model for MLSE using Viterbi algorithm



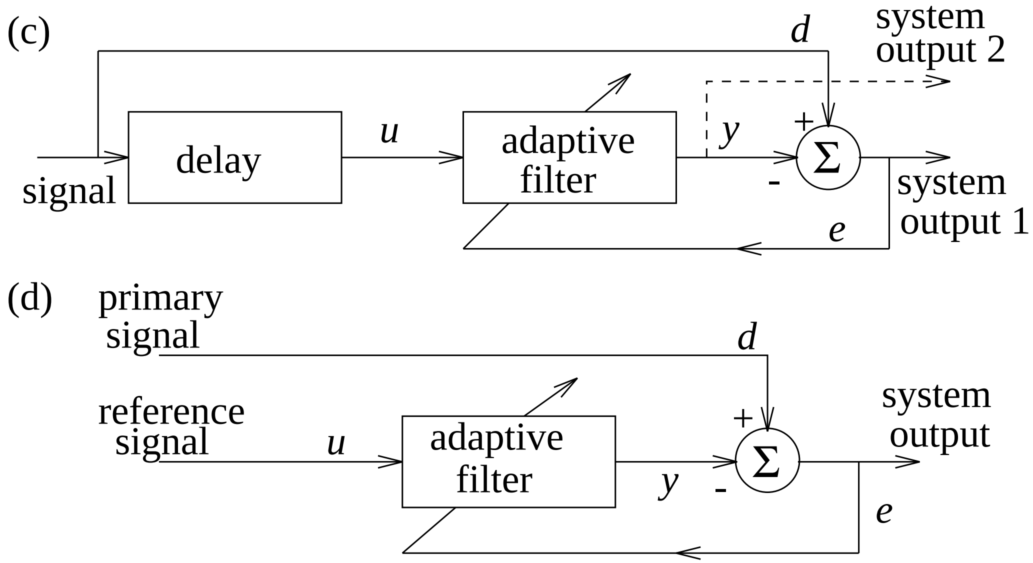
(B) **Inverse modelling**: Adaptive filter provides an inverse model to an unknown noisy plant

The adaptive filter is driven by the noisy plant output, and a delayed version of the plant input constitutes the desired output

Examples include predictive deconvolution and adaptive equalisation

Notice the blind deconvolution is a generalised case of inverse modelling, where adaptive filter does not have access to the plant input and therefore cannot use it as the desired output

Applications (continue)



(C) **Prediction**: Adaptive filter provides prediction of the current value of a signal

The current signal value is the desired response, and past signal values are filter input

The adaptive filter output or error may serve as the system output. In the former case, the system operates as a predictor, and in the latter case, it operates as a prediction-error filter

Examples include linear prediction coding and signal detection

(D) **Interference cancelling**: Adaptive filter cancel unknown interference contained in the information-bearing signal (known as the primary signal)

The primary signal serves as the desired response for the adaptive filter, and a reference (auxiliary) signal is employed as the input to the adaptive filter. The reference signal must contain the unknown interference and should be uncorrelated with the information-bearing signal

Examples include adaptive noise cancelling, echo cancellation and adaptive beamforming

Summary

- Understand communications technology is about “Shannon meets Wiener’
- Adaptive signal processing is an enabling technology for communications
  - Appreciation of general structure of adaptive filter and relevant issues;
  - Appreciation of “simplicity’ of FIR filter and “complexity’ of IIR filter
  - Concepts of cost function and optimisation
  - Appreciation of practical applications of adaptive filter
- **Communication** and **computing** are ‘**entangled**’
  - **Turning**’s machine intelligence or AI
  - **Wiener**’s cybernetics and society
  - **Schrödiner**’s theory of life depends on negative entropy
  - Machine learning/neural networks $\longrightarrow$ **deep learning**