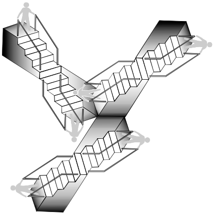
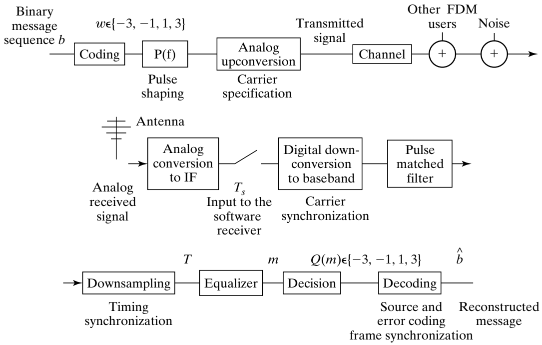
BITS TO SYMOLS TO SIGNALS AND BACK AGAIN

- ⋆ Bits to Symbols
- ⋆ Symbols to Signals
- ⋆ Correlation
- ⋆ Signals to Symbols
- ⋆ Symbols to Bits



idealized system

Bits to Symbols to Signals and Back Again



- Our interest here is in the coding and pulse-shaping of the transmitter and the corresponding decoding and pulse-matched filtering with downsampling of the receiver.
- The processing we call upon is correlation.

Bits to Symbols

- *Text to bits*: ASCII, 8 bits per character, e.g. T is 01010100 (from `dec2base('T',2,8)`)
- *Further encoding*: The bit stream can be further manipulated, e.g. to provide extra resilience to broadband channel noise, into a different bit stream to be converted to symbols to signals. Its reversal must be incorporated in the decoder for the receiver conversion from bits to text.
- *Bit pairs to multilevel symbols*: For 4-PAM alphabet of $\{\pm 1, \pm 3\}$

  $$\begin{array}{l} 11 \rightarrow +3 \\ 10 \rightarrow +1 \\ 01 \rightarrow -1 \\ 00 \rightarrow -3 \end{array}$$

  e.g. T is $\{-1, -1 - 1 - 3\}$ (from `letters2pam('T')`)

Symbols to Signals

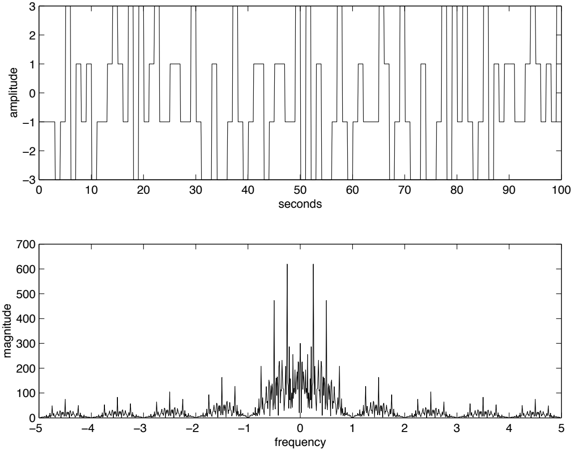
- The symbol sequence scales successive pulses in a periodic pulse train to form the analog signal to be transmitted.
- "Good" pulses
    - offer no interference to their neighboring pulses (at least at the sampling instants)
    - make efficient use of spectrum
    - are resilient to noise
    - admit a low complexity time-domain implementation
- Realistic pulses compromise among these features.

Symbols to Signals (cont'd)

- Inter-symbol interference can be removed by using a pulse time-limited to less than the symbol transmission period.
- However, a time-limited pulse has a magnitude spectrum that is unlimited in frequency.
- To conserve bandwidth, we would like to pick a pulse shape resulting in rapidly decaying tails in the magnitude spectrum of the associated message signal.

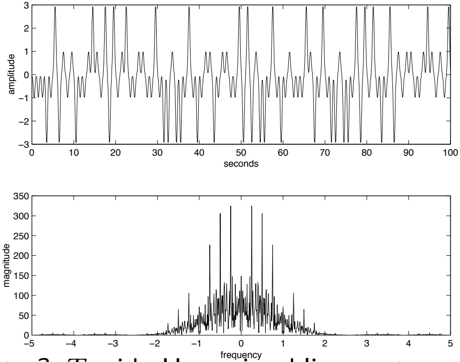
Text to Signals Example

- Text: "Transmit this text"
- Coding: 8-bit ASCII to 4-PAM
- Rectangular pulse message and spectrum (from `pulseshape`)



Text ... Example (cont'd)

- 10-sample even-symmetric, Hamming blip:
  $\{0.08, 0.19, 0.46, 0.77, 0.97, 0.97, 0.77, ..., 0.08\}$
- Hamming blip message and spectrum



- *So, which is better?* T-wide Hamming blip spectrum has lower power bandwidth (but higher null-to-null) bandwidth than the spectrum of a T-wide rectangular pulse.

Correlation

- Similarity test for pulse baud/sample time and frame location can exploit correlation function peakiness.
- Crosscorrelation

$$R_{wv}(j) = \lim_{T\to\infty} \frac{1}{T} \sum_{k=-T/2}^{T/2} w[k]v[k+j]$$

- Called autocorrelation when $w = v$ and labelled $R_w[j]$.
- With finite length data records, MATLAB `xcorr` removes limit and $1/T$ scale factor and sums over nonzero values of w and v.
- Crosscorrelation is not the same as convolution of w and v which is

$$y[j] = w[j] * v[j] = \sum_{k=-\infty}^{\infty} w[k]v[j-k]$$

Correlation (cont'd)

If either w or v is even symmetric then crosscorrelation (without the limit and the $1/T$ normalizer) and convolution are the same.

- To show this, assume $w[i] = w[-i]$
- Then
$$\sum_{k=-\infty}^{\infty} w[k]v[j-k] = \sum_{k=-\infty}^{\infty} w[-k]v[j-k]$$
- Redefine dummy variable $-k$ as $\bar{k}$ and
$$\begin{aligned} \sum_{k=-\infty}^{\infty} w[k]v[j-k] &= \sum_{\bar{k}=\infty}^{-\infty} w[\bar{k}]v[j+\bar{k}] \\ &= R_{wv}[j] \end{aligned}$$
- So, given w even-symmetry, correlation can be thought of as a linear filter (and vice versa).

Correlation (cont'd)

Crosscorrelation can also be used to identify the delay in a scaled, delayed, noisy measurement of a known signal.

- Assume $r[k] = gw[k-\Delta] + v[k]$ and $R_{wv}[j] = 0$ for all j.
- Crosscorrelation of w and r

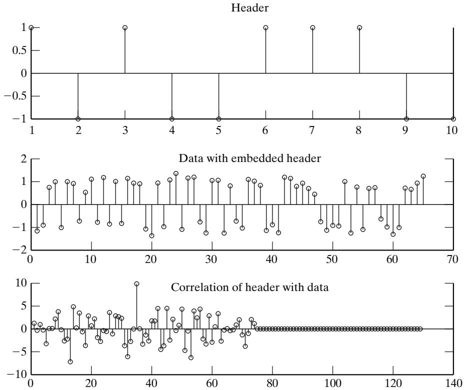
$$
\begin{aligned}
R_{wr}[j] &= \lim_{T\to\infty} \frac{1}{T} \sum_{k=-T/2}^{T/2} gw[k]w[k-\Delta+j] \\
&\quad + \lim_{T\to\infty} \frac{1}{T} \sum_{k=-T/2}^{T/2} w[k]v[k+j] \\
&= gR_w[j-\Delta]
\end{aligned}
$$

  i.e. a time-shifted autocorrelation.
- If R_w is deliberately single-peaked then Δ is readily ascertained from R_{wr}.

Correlation (cont'd)

Example:

- Signal: binary data for 30 symbols, a 10 symbol binary marker, and 25 more data symbols
- Top ∼ marker; Middle ∼ signal; Bottom ∼ signal and marker crosscorrelation (from `correx`)



Correlation (cont'd)

- With two vectors w and v `xcorr(w,v)` zero pads the shorter of its two arguments with trailing zeros before computing the result $\sum_k w[k]v[k+j]$ for each j.
- `xcorr(w,v)` starts by setting the last entry of the second vector over the first of the first vector, multiplies adjacent pairs, and sums the products, slides top vector to right by one position, and repeats.
- Because w is 55 entries shorter than v, a string of 55 zeros appears as the tail of the output of `xcorr`.
- Header is self-aligned at the 35th sample time, as reflected in largest peak in correlation output.

Signals to Symbols

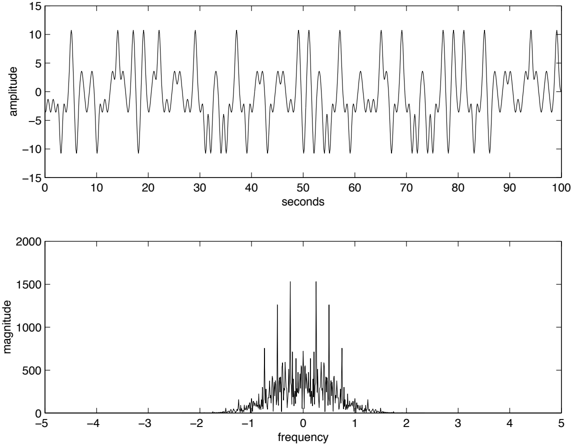
- At the receiver, could use a LPF with bandwidth similar to that of pulse shape to suppress out-of-band interferers.
- Could use a pulse-matched correlator to enhance peakiness of pulses for timing assistance.
- For an even symmetric pulse, correlation and convolution are the same operation.
- Correlation with symmetric pulse shape often called “receive filter”.

Text to Signals to Symbols to Text Example

- *Received signal generation*: Recall previous text to transmitted 4-PAM example with Hamming blip pulse ($T = 1$, $T_s = 0.1$) using `pulseshape`.
- *Receive filter*: Use Hamming blip pulse shape as in `recfilt`.
- *Baud timing*: First maximum in receive filter output magnitude (at index `N*M`) corresponds to first symbol.

Text ... Example (cont'd)

Using `plotspec(y(N*M:end),.1)`, the receive filter output and spectrum can be plotted.

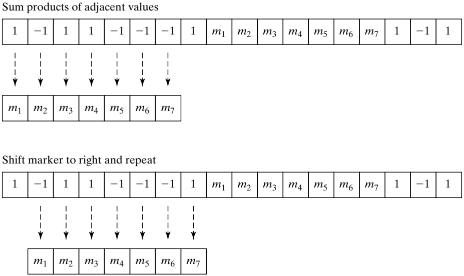


Text ... Example (cont'd)

- *Normalization for quantizer*: When first local maximum in magnitude is normalized by inner product of 10-element pulse vector of 3.583 a -1 is recovered, as expected as first symbol of "T" quadruple.
- *Downsampling*: Subsequent sampling of the receive filter output every symbol period after the first local magnitude maximum recovers the transmitted symbol sequence without error.
- *Decoding*: Simply reverse ASCII code with quadruples located relative to first magnitude peak. This frame synchronization is typically not so easy as here where we locked on to the first peak as the first symbol of the first quadruple of the message to be decoded following a zero preceding (or marker) signal.

Symbols to Bits

- *Frame synchronization*: Successful decoding requires properly grouping the quadruples of symbols to be transformed into a text character.
- *Inserted marker*: Special subsequence inserted at the front of a sequence of quadruples (to be found by correlation) and subsequent quadruples are framed from this reference point.
- *Correlation*:



Symbols to Bits (cont'd)

- *Marker autocorrelation peakiness*: Some markers are peakier than others:
  - transmitted sequence example: ..., $+1$, -1, $+1$, $+1$, -1, -1, -1, $+1$, marker, $+1$, -1, $+1$, ...
  - Marker A: 1, 1, 1, 1, 1, 1, 1
  - Marker B: 1, 1, 1, -1, -1, 1, -1
  - Crosscorrelation with marker starting at 7th transmitted sequence entry shown
    - ⊙ for marker A: -1, -1, 1, 1, 1, 3, 6, 7, 7, 7, 5, 5.
    - ⊙ for marker B: 1, 1, 3, -1, -5, -1, -1, 1, 7, -1, 1, -3
  - Marker B is a "pseudonoise" (PN) sequence so named for its single-peaky autocorrelation typically associated with a white "noise" signal.

Scrambling

- Each block of message data M (in binary 0/1 form) need not be spectrally flat, though the transmission system may be designed under that assumption.
- To encourage spectral flatness of the transmitted signal, a pre-arranged scrambling sequence S can be added modulo-2, bit-by-bit to the message string M and the sum transmitted.
- After demodulation to a symbol sequence and bit string recovery, the receiver locates the start of the binary message block (e.g. using correlation to a marker) and adds S again bit-by-bit with modulo-2 arithmetic where $1 + 1 = 0$ and $0 + 0 = 0$.
- Thus, if aligned correctly the added scrambling sequence disappears leaving M.

NEXT... We compose a working digital radio for idealized operating conditions, but stuff happens.