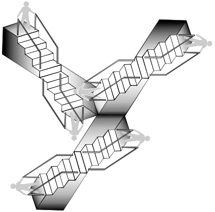
SAMPLING WITH AUTOMATIC GAIN CONTROL

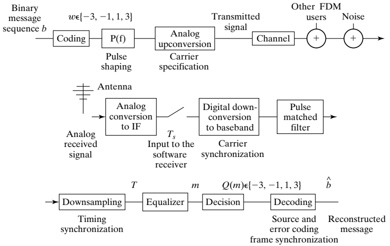
- ⋆ Impulse Sampler
- ⋆ Interpolation
- ⋆ Iterative Optimization
- ⋆ Automatic Gain Control
- ⋆ Tracking Example: Time-Varying Fade



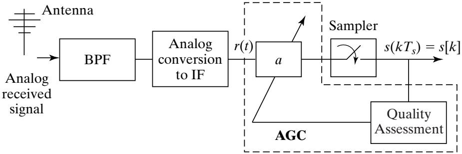
idealized system

Sampling with AGC

Continuing our expansion out from the channel at the center of our telecommunication system beyond the analog up and down converters



we now focus on the sampler and its surrounding automatic gain control (AGC) in the receiver front end

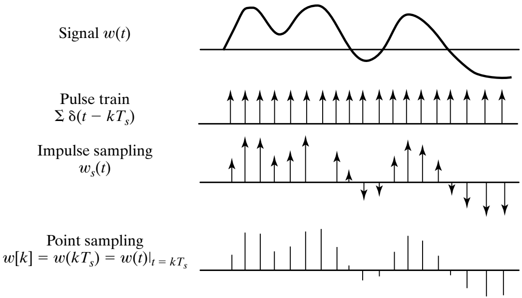


Impulse Sampler

With $w(t)$ the input to an impulse sampler, the output $w_s(t)$ is

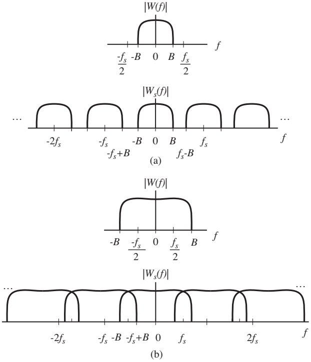
$$w_s(t) = w(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$$

Analog $w(t)$ is multiplied point-by-point by a pulse train



Impulse Sampler (cont'd)

- Using (A.28) with $f_s = 1/T_s \Rightarrow W_s(f) = f_s \sum_{n=-\infty}^{\infty} W(f - nf_s)$
- Relative to $W(f)$, $W_s(f)$ has been scaled by f_s and contains replicas at every f_s.
- Largest frequency in $W(f)$ less than $f_s/2$ (top plot) and slightly larger than $f_2/2$ (bottom)

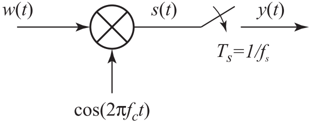


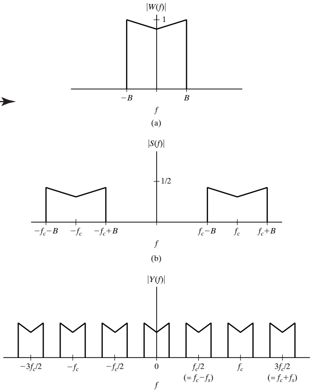
Impulse Sampler (cont'd)

- *Nyquist Sampling Theorem*:
  If the signal $w(t)$ is bandlimited to B, ($W(f) = 0$ for all $|f| > B$) and if the sampling rate is faster than $f_s = 2B$, then $w(t)$ can be reconstructed exactly for all t from its samples $w(kT_s)$.
- *Sub-Nyquist Sampling*:
  - What if the signal to be sampled is a passband signal, but the signal to be reconstructed is this passband signal downconverted to a baseband signal with a much lower maximum frequency?
  - Can sub-Nyquist sampling of the passband signal be employed without aliasing of the baseband signal?
  - The following examples provide a positive answer.

Impulse Sampler (cont'd)

- *Example*:
  - Consider $f_s = f_c/2$

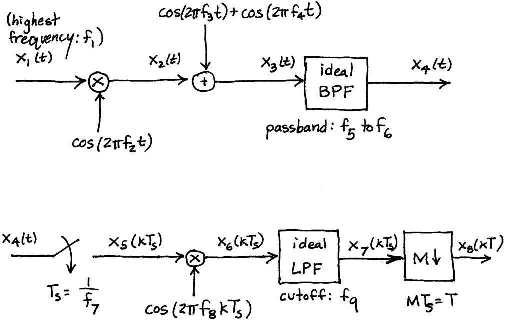




  - Works for $f_s = f_c/n$
  - What if f_s not exactly f_c/n?

Impulse Sampler (cont'd)

- *Another Example*: For a PAM system the sampler, downconverter, and downsampler (to symbol period T) should produce an output x_8 with a spectrum matching that of a sampled version (with sample period T) of the baseband source x_1.



Impulse Sampler (cont'd)

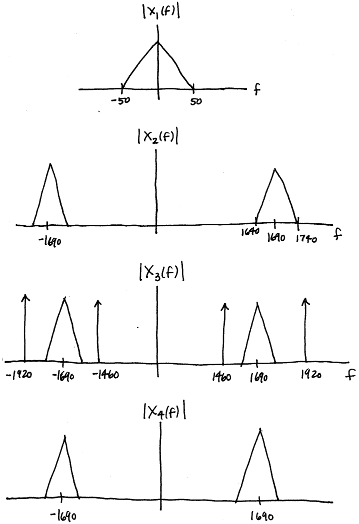
Another Example (cont'd)

- For the following specifications in kHz
  - $f_1 = 50$
  - $f_2 = 1690$
  - $f_3 = 1920$
  - $f_4 = 1460$
  - $f_5 = 1620$
  - $f_6 = 1760$
  - $f_7 = 800$
  - $f_8 = 90$
  - $f_9 = 60$

  given $|X_1(f)|$ as even-symmetric, triangular shaped, and centered at zero frequency, we can draw $|X_i(f)|$ for $i = 1, 2, ..., 8$ to show that $|X_8(f)|$ matches $|X_1(f)|$ (up to a scalar gain factor) with $M = 2$.

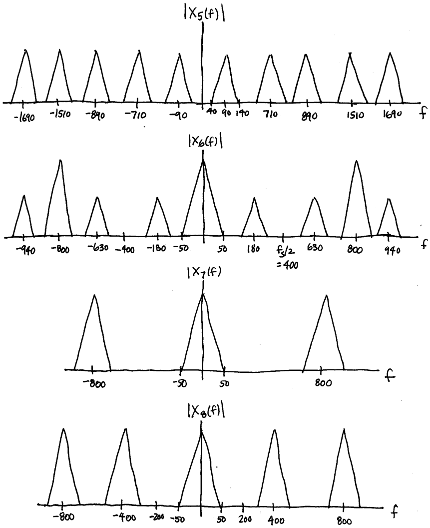
Impulse Sampler (cont'd)

Another Example (cont'd)



Impulse Sampler (cont'd)

Another Example (cont'd)



Interpolation

- *Objective*: Use signal samples from times kT_s to reconstruct the analog signal value at a time instant not among this set of sample times.
- *Sinc interpolator*:

$$w(t)|_{t=\tau} = w(\tau) = \int_{t=-\infty}^{\infty} w_s(t)\mathrm{sinc}(\tau - t)dt$$

Because $w_s(t)$ is nonzero only when $t = kT_s$,

$$w(\tau) = \sum_{k=-\infty}^{\infty} w_s(kT_s)\mathrm{sinc}(\tau - kT_s)$$

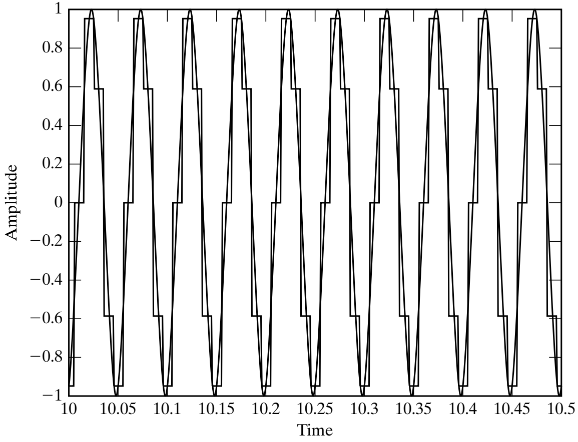
- *Prescription for perfection*: As long as $f_s > 2B$ (where B is the highest frequency present in $w(t)$) this (doubly infinite) sinc interpolator is exact.
- *Filtering interpretation*: Creation of $w(\tau)$ can be interpreted as a convolution of w_s with a sinc-shaped impulse response.

Interpolation (cont'd)

- *Ideal LPF Interpolator*: Convolution in time domain is multiplication in frequency domain. Spectrum of sinc is a rectangle, i.e. an ideal LPF. Thus, an ideal lowpass filter with appropriate cutoff frequency is a perfect interpolator for a Nyquist-sampled signal.
- *Perfection inhibiting practicalities*: In practice, it is necessary to truncate the doubly infinite convolutional sum. Furthermore, $w(t)$ can always be expected to have traces of frequencies above B. Therefore, in practice, we must settle for an approximation.
- *Non-ideal LPF interpolator*: Fortunately, any suitable LPF (with nonzero, flat magnitude and linear phase up to frequency B and fully rejecting before reaching next higher frequency chunk in spectrum of w_s) will provide accurate interpolation.

Interpolation (cont'd)

Example: Using `sininterp` to reconstruct a sinusoid sampled five times per period (as indicated by the choppy staircase zero-order-hold reconstruction of the samples)



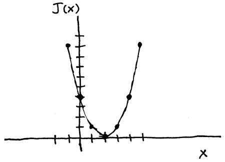
Iterative Optimization

- *Task*: Find value of x at which polynomial

$$J(x) = x^2 - 4x + 4$$

  is minimized.

- *Plot cost function*:



- *Zeroing derivative*: Setting the derivative

$$\frac{\partial J(x)}{\partial x} = 2x - 4$$

  to zero by selecting $x = 2$ locates a stationary point, i.e. either a maximum or minimum.

Iterative Optimization (cont'd)

- *Minimum or maximum*: The stationary point $x = 2$ is a minimum because the derivative of the derivative evaluated at $x = 2$ is positive, i.e.

$$\frac{\partial^2 J(x)}{\partial x^2} = \frac{\partial(2x-4)}{\partial x} = 2$$

  A negative second derivative would indicate a local maximum.
- *Iterative gradient descent minimizing strategy*: Because the derivative points toward larger values, we descend in the opposite direction with μ positive (and small)

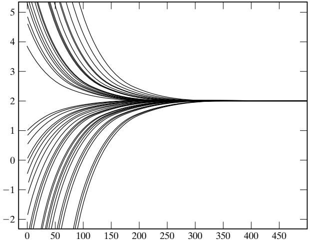
$$x[k+1] = x[k] - \mu \frac{\partial J(x)}{\partial x}\bigg|_{x=x[k]}$$

  In this case,

$$x[k+1] = x[k] - \mu(2x[k]-4)$$

Iterative Optimization (cont'd)

- *Simulated test*: From 50 different starting points with $\mu = 0.01$ (using `polyconverge`), we converge to the desired setting



- *When maximizing*: If seeking maximum, would change sign on correction term so

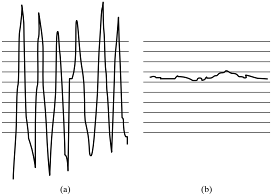
$$x[k+1] = x[k] + \mu \frac{\partial J(x)}{\partial x}\bigg|_{x=x[k]}$$

Iterative Optimization (cont'd)

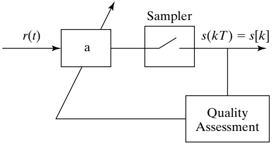
- *Convergence consequence*: If x converges to the tight vicinity of a particular value, then the update term must be zero (at least on average). For a gradient descent, this implies that the gradient is zero, as expected. In our example, zeroing the update term $2x[k] - 4$ leads trivially to the desired answer of $x = 2$.
- *Cost function modality*: With only one stationary point, our cost function $J(x) = x^2 - 4x + 4$ is unimodal. Other cost functions could be multimodal, which would mean that a gradient descent would be trapped in a local minimum. The particular local minimum would depend on the starting value of x.

Automatic Gain Control (AGC)

- An AGC maintains the dynamic range of a (zero-average) signal by attenuating when it is too large (as in (a)) and by amplifying when too small (as in (b)).



- AGC adjusts gain parameter a so average energy at output remains (roughly) fixed, despite fluctuations in average received energy.



AGC (cont'd)

Gain Tuning:

- We are to choose a for a received waveform $r(t)$ segment that produces sampler outputs $s[k]$ with the intent of having the average s^2 value over that dataset match a preselected constant S^2.
- Because $s[k] = ar(kT_s)$, we can choose

$$a^2 = \frac{S^2}{\left(\frac{1}{N}\sum_{i=1}^{N} r^2[k+i]\right)} = \frac{S^2}{\text{avg}\{r^2[k]\}}$$

(preferring $a > 0$) to make (as desired)

$$\{\frac{1}{N}\sum_{i=1}^{N} s^2[k+i]\} = S^2$$

- Unfortunately, we need the samples of r, which are not available on the DSP side of the receiver, to solve this formula for a.
- Our search for a gain tuner continues.

AGC (cont'd)

Heuristic Algorithm Development:

As an alternative, consider the following strategy:

- select an initial positive a.
- As a sample s arrives, compare its square to S^2.
- If s^2 at that particular sample instant is greater than S^2, we will reduce a positive a to a smaller positive value. If a is negative, we would decrease its magnitude, i.e. increase it toward zero.
- Plus, the correction term should be larger the further s^2 is from S^2.
- Similarly, if $s^2 < S^2$, we will increase a positive a by an amount proportional to $S^2 - s^2$. If a is negative, a should be decreased (i.e. made more negative), so its magnitude increases.

AGC (cont'd)

An algorithm that performs this strategy is

$$a[k+1] = a[k] + \mu\{\text{sign}(a[k])\}(S^2 - s^2[k])$$

where μ is a suitably small positive stepsize. (The sign($a[k]$) term can be removed if $a[k]$ starts and stays positive.)

- Can this algorithm be implemented from data available on the DSP side of the sampler?
  Ans: Yes, s (and not r) is needed
- Will this algorithm converge to the desired a of $\pm S/\sqrt{\frac{1}{N}\sum_{i=1}^{N} r^2[k]}$?
  Ans: It depends what you mean by "converge".

AGC (cont'd)

- The candidate algorithm

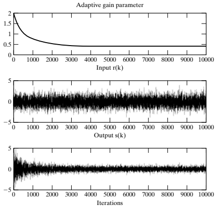
$$a[k+1] = a[k] + \mu\{\text{sign}(a[k])\}(S^2 - s^2[k])$$

  cannot be expected to converge to a fixed value.
- Because r ranges widely, only on average does a^2r^2 (or s^2) actually equal S^2.
- The resulting (typically) nonzero instantaneous error in $S^2 - s^2$ and a nonvanishing stepsize μ will result in a change in a even if it is already at the right value for the average behavior of s^2.
- A sufficiently small μ should keep this asymptotic rattling within a tolerable level.

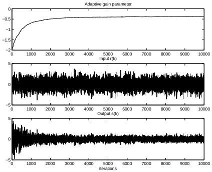
AGC (cont'd)

Testing:

- Using `agcgrad` with $\text{avg}\{r^2\} \approx 1$ and $S^2 = 0.15$, the desired $a \approx \sqrt{0.15} \approx 0.38$.
- Start at $a[0] = 2$ with $\mu = 0.001$

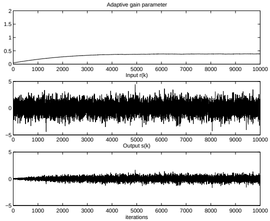


- Start of $a[0] = -2$ with $\mu = 0.001$

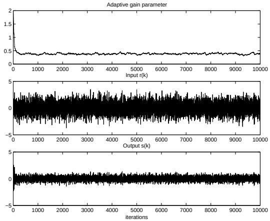


AGC (cont'd)

- Start at $a[0] = 0.05$ with $\mu = 0.001$



- Start at $a[0] = 2$ with $\mu = 0.02$



AGC (cont'd)

Observations:

- Asymptotically, this algorithm hovers in a small region about the desired answer.
- The asymptotic hovering region's size can be decreased by reducing the stepsize μ, which also reduces the algorithm convergence rate.
- When the average value of the hovering parameter has effectively reached a fixed value, the average of $a[k+1]$ will equal the average of $a[k]$ such that from our algorithm

  $$a[k+1] = a[k] + \mu \text{sign}(a[k])(S^2 - s^2[k])$$

  the average of the correction term $\mu \text{sign}(a[k])(S^2 - s^2[k])$ must be zero.
- With $\mu > 0$ and the asymptotic hovering $a[k]$ not changing sign, zeroing the average correction term zeros the average of $S^2 - s^2$. But, indeed that is what we seek.

AGC (cont'd)

Gradient Descent Algorithm Development:

- As a more generalizable approach to adaptor algorithm development consider specifying a cost function and using an iterative optimizer based on gradient descent.
- Try $J_{LS}(a) = (1/4)\text{avg}\{\left[s^2[k] - S^2\right]^2\}$ with the definition of "avg" as

$$\text{avg}\{x[k]\} = (1/N) \sum_{i=k}^{k-N+1} x[i]$$

- With $s[k] = ar[k]$

$$J_{LS}(a) = (1/4)\text{avg}\{\left[a^2 r^2[k] - S^2\right]^2\}$$

- A gradient descent algorithm

$$a[k+1] = a[k] - \mu \frac{\partial J_{LS}(a)}{\partial a}\bigg|_{a=a[k]}$$

AGC (cont'd)

- For small stepsize μ, from Appendix G, differentiation and averaging are approximately interchangeable

$$\begin{aligned}\frac{\partial J_{LS}(a)}{\partial a} &= \left(\frac{1}{4}\right)\frac{\partial}{\partial a}[\text{avg}\{a^2r^2(kT)-S^2)^2\}] \\ &\approx \left(\frac{1}{4}\right)\text{avg}\{\frac{\partial}{\partial a}[a^2r^2(kT)-S^2)^2]\}\end{aligned}$$

- So

$$\frac{\partial J_{LS}(a)}{\partial a} \approx \text{avg}\{ar^2(kT)(a^2r^2(kT)-S^2)\}$$

- Replace ar with s and ar^2 with s/a and a with $a[k]$

$$a[k+1]=a[k]-\mu\text{avg}\{(s^2[k]-S^2)\frac{s^2[k]}{a[k]}$$

- This is different from the heuristically developed algorithm.

AGC (cont'd)

- Consider another cost function

$$J_N(a) = \mathrm{avg}\{|a|((s^2[k]/3) - S^2)\}$$

- For small stepsize μ, from Appendix G, differentiation and averaging are approximately interchangeable

$$\begin{aligned}\frac{\partial J_N(a)}{\partial a} &= \frac{\partial}{\partial a}[\mathrm{avg}\{|a|\left(\frac{a^2r^2(kT)}{3} - S^2\right)\}] \\ &\approx \mathrm{avg}\{\frac{\partial}{\partial a}[|a|\left(\frac{a^2r^2(kT)}{3} - S^2\right)]\}\end{aligned}$$

- With $\partial|a|/\partial a = \mathrm{sign}(a)$ and (A.60)

$$\frac{\partial J_N(a)}{\partial a} \approx \mathrm{avg}\{|a|(1/3)2ar^2(kT) + \mathrm{sign}(a)(1/3)a^2r^2(kT)\} - \mathrm{sign}(a)S^2$$

- With $\mathrm{sign}(a)|a| = a$

$$\frac{\partial J_N(a)}{\partial a} \approx \mathrm{avg}\{\mathrm{sign}(a)\left(a^2r^2(kT) - S^2\right)\}$$

AGC (cont'd)

- With $a^2 r^2 = s^2$

$$\frac{\partial J_N(a)}{\partial a} \approx \text{avg}\{\text{sign}(a)\left(s^2[k] - S^2\right)\}$$

So, the stationary points of zero gradient are in the right places with $\text{avg}\{s^2\} = S^2$.

- With $\partial(\text{sign}(a))/\partial a = 0$ everywhere but $a = 0$, the second derivative is approximately

$$\text{avg}\{\frac{\partial}{\partial a}\left[\text{sign}(a)\left(a^2 r^2(kT) - S^2\right)\right]\}$$

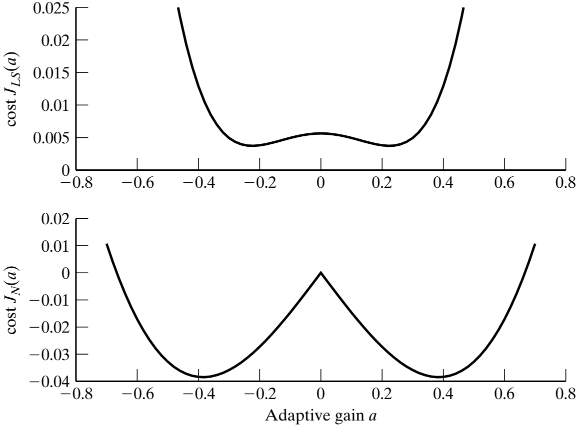
$$= \text{avg}\{2a \ \text{sign}(a) r^2(kT)\}$$

$$= \text{avg}\{2|a| r^2(kT)\} > 0$$

So, stationary points at other than $a = 0$ are minima.

AGC (cont'd)

- With constant $\mathrm{avg}\{r^2\}$ and S, J_N has double dip "egg carton" style cross section as does J_{LS}.
- For specific data set (with $N = 1000$) in `agcerrorsurf`



AGC (cont'd)

- Computation of the gradient requires that a remain constant over the N samples over which $\text{avg}\{s^2\}$ is composed.
- Consider squeezing the averaging window to a single sample so $N = 1$ and

$$a[k+1] = a[k] - \mu \text{sign}(a[k]) \left(s[k]^2 - S^2\right)$$

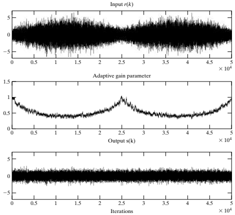
- This is the algorithm developed heuristically and tested previously.
- This algorithm also emerges from first reducing the averaging window to $N = 1$ in the cost function and then taking the gradient and forming a gradient descent iteration.
- This technique of shrinking the averaging window so averaging is explicitly removed works because LPF action of adaptation acts similarly to averaging before updating.

Tracking Example: Time-Varying Fade

- To demonstrate desired tracking capability, use `agcvsfading` to test

$$a[k+1] = a[k] - \mu \text{sign}(a[k]) \left(s[k]^2 - S^2\right)$$

with $\mu = 0.01$, $S^2 = 0.5$, $a[1] = 1$, and a large, slow, oscillating channel gain (initially 0.75)



- Fade must be changing sufficiently slowly and the input must never die for the AGC with small stepsize to track adequately.

NEXT... DFT and digital filter design tidbits for the variety of linear filters in a recevier.