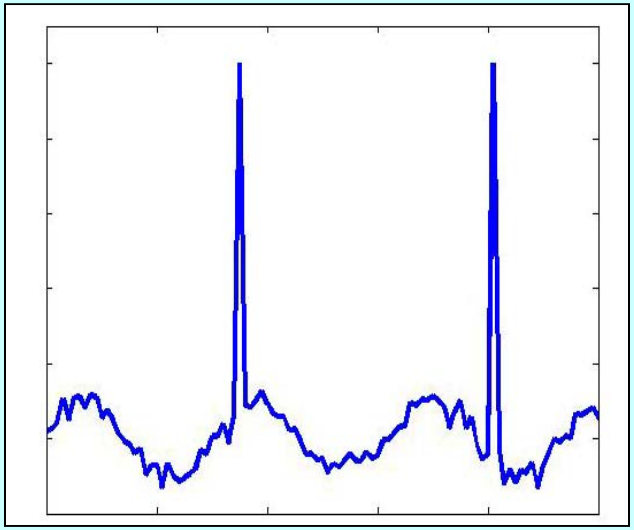
Introduction to Discrete-Time Systems

- Linear filters will not handle outliers very well – they are interpreted as *impulses*
- First defense against outliers: find out their origin and eliminate them at the beginning!
- Detection: Exceeding a known, fixed bound, or an impossible deviation from previous values. *Example: vehicle speed >> the possible value given thrust level and prior tests.*
- Second defense: set data to NaN (or equivalent), so it won't be used in any calculations.
- Third defense: try to fill in.

Example:

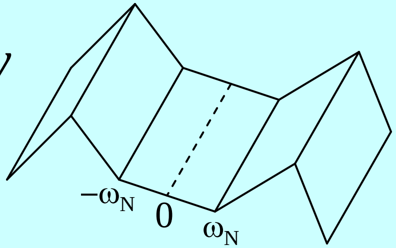
```
if abs(x(k) – x(k-1)) > MX,
        x(k) = x(k-1) ;
end;
```

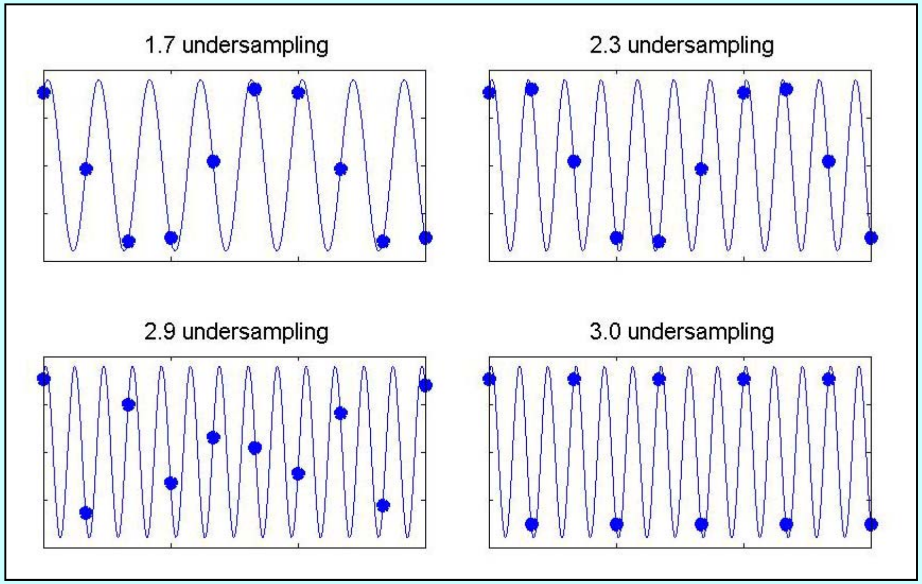
Can get lost if multiple outliers occur!



Time Resolution in Sampled Systems

Frequency folding





- The Sampling Theorom shows that the highest frequency that can be detected by sampling at frequency $\omega_s = 2\pi/\Delta t$ is the Nyquist rate: $\omega_N = \omega_s / 2$.
- Higher frequencies than this are "aliased" to the range below the Nyquist rate, through "frequency folding." *Includes sensor noise!*
- The required rate for "visual" analysis of the signal, and phase and magnitude calculation is much higher, say ten samples per cycle.

Filtering of Signals in Discrete Time

$x \rightarrow \boxed{\text{Filter}} \rightarrow x_f$

Use good judgement!

- filtering brings out trends, can reduce noise, but
- filtering obscures some properties of the signal

Causal filtering: $x_f(t)$ depends only on past measurements – appropriate for real-time implementation

Example: $x_f(t) = (1 - \varepsilon)\, x_f(t-\Delta t) + \varepsilon\, x(t-\Delta t)$

First-order lag, backward Euler

Acausal filtering: $x_f(t)$ depends on past *and future* measurements – appropriate for post-processing

Example: $x_f(t) = [\, x(t+\Delta t) + x(t) + x(t-\Delta t)\,] / 3$

Moving window, centered, uniform weighting

A first-order filter transfer function in the freq. domain:

$$x_f(j\omega) / x(j\omega) = \lambda / (j\omega + \lambda)$$

At <u>low ω</u>, this is approximately 1 (= λ/λ)

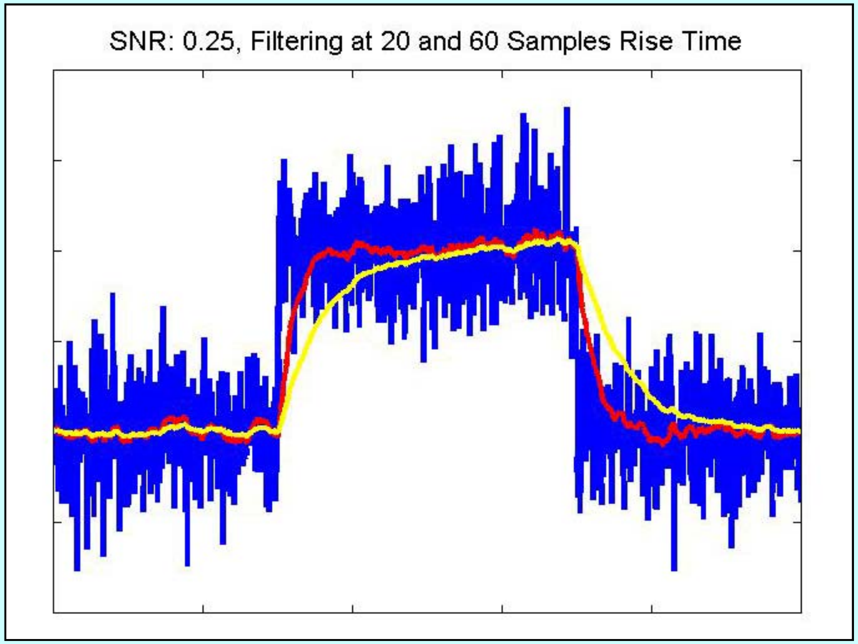
At <u>high ω</u>, this goes to 0 magnitude, with 90 degrees phase lag (λ / jω = -jλ / ω)

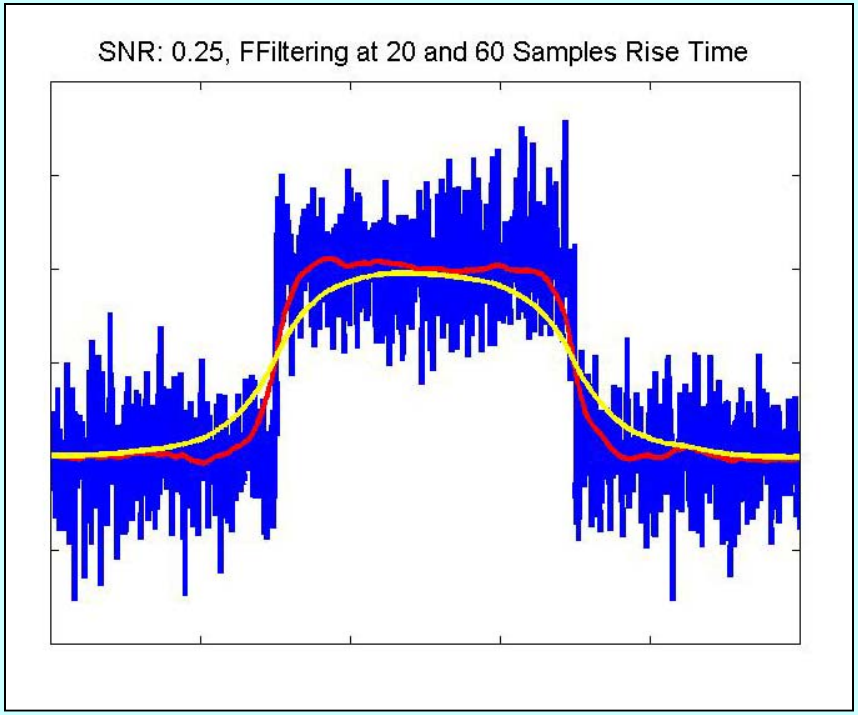
Time domain equivalent:

$$dx_f / dt = \lambda\ (x - x_f)$$

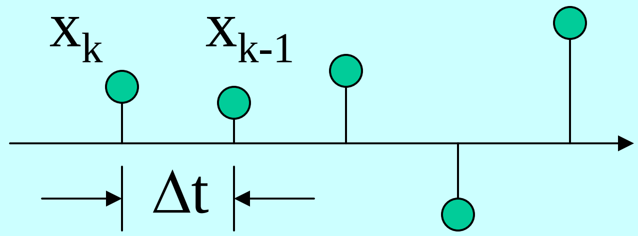
In discrete time, try this quick algorithm:

$$x_f(k) = (1-\lambda\Delta t)\ x_f(k-1) + \lambda\Delta t\ \ x(k-1)$$





Thinking about discrete-time signals: time step Δt between x_k and x_{k-1}



Example: $dx/dt = u$

Fourier Transform: $j\omega\, X(j\omega) = U(j\omega)$

$X(j\omega) / U(j\omega) = 1 / j\omega$

Let q be the delay operator: <u>q-transform:</u> $X(q) = \sum_0^\infty x_k\, q^{-k}$

This means: $X(q) = x_0 + x_1 q^{-1} + x_2 q^{-2} + \ldots$

Make a discrete-time approximation for the example:

$dx/dt = u \rightarrow \quad (x_k - x_{k-1}) / \Delta t = u_{k-1}$

$X(q)\,(1 - q) / \Delta t = q\, U(q)$

$X(q) / U(q) = q\, \Delta t / (1 - q)$

So $j\omega = (1 - q) / q\, \Delta t$ for this case *backward Euler*

$q = 1 / (1 + j\omega\, \Delta t)$

Try a different method:

$$dx/dt = u \rightarrow \quad (x_k - x_{k-1}) / \Delta t = u_k$$
$$X(q)\,(1 - q) / \Delta t = U(q)$$
$$X(q) / U(q) = \Delta t / (1 - q)$$

So $j\omega = (1 - q) / \Delta t$ for this case *forward Euler*

$q = 1 - j\omega\, \Delta t$

Let's do a more careful job by using both old and new u's:

$$dx/dt = u \rightarrow \quad (x_k - x_{k-1}) / \Delta t = (u_k + u_{k-1}) / 2$$
$$X(q)\,(1 - q) / \Delta t = U(q)\,(1 + q) / 2$$
$$X(q) / U(q) = (1 + q)\,\Delta t / 2\,(1 - q)$$

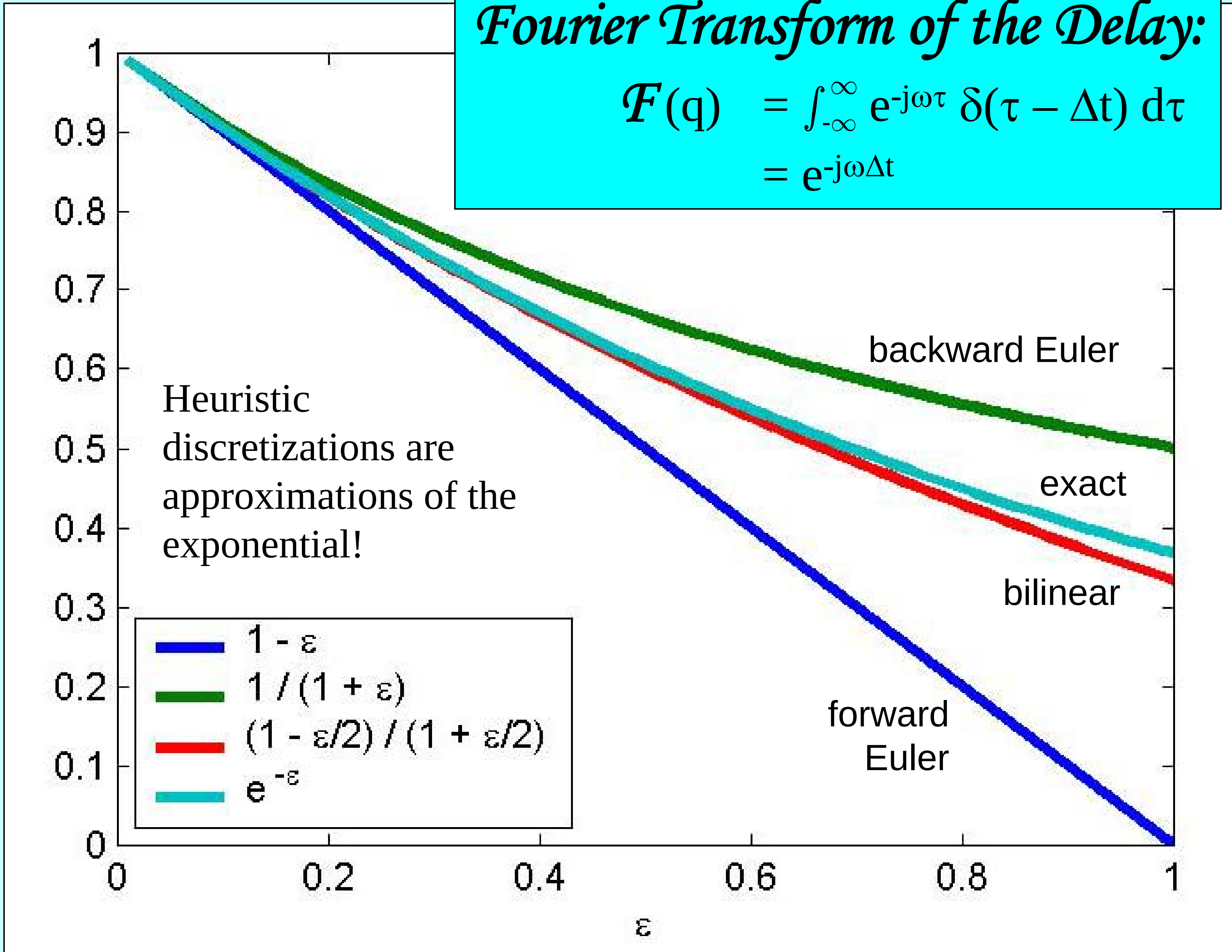
So $j\omega = 2\,(1 - q) / (1 + q)\,\Delta t$ *bilinear approximation*

$q = (1 - j\omega\, \Delta t / 2) / (1 + j\omega\, \Delta t / 2)$

Fourier Transform of the Delay:

$$\mathcal{F}(q) = \int_{-\infty}^{\infty} e^{-j\omega\tau}\,\delta(\tau - \Delta t)\,d\tau$$

$$= e^{-j\omega\Delta t}$$



Translating an analog system into discrete-time, e.g.,
Digital Filtering and Control:

Example:

dx/dt = -x + u   *(low-pass filter with cutoff frequency 1rad/s)*

$X(j\omega) / U(j\omega) = 1 / (j\omega + 1)$

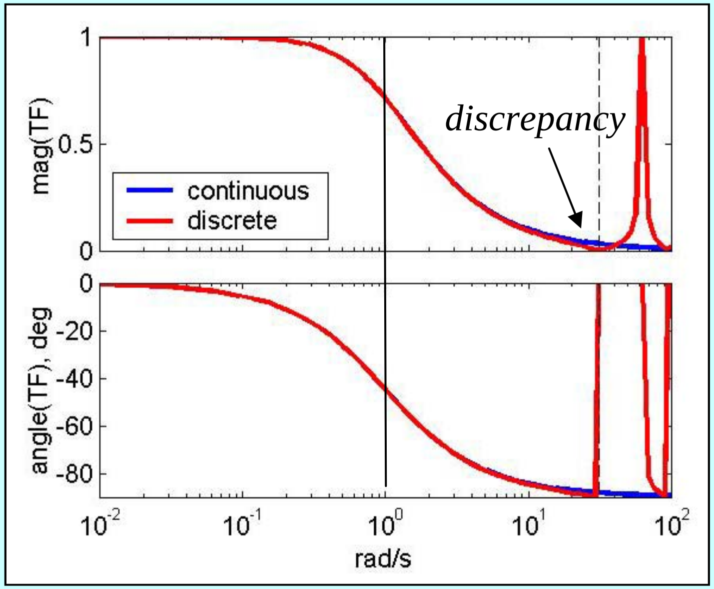
Substitute the bilinear approximation:

$X(q) / U(q) \quad = (1 + q) / [(1 + 2 / \Delta t) + q (1 - 2 / \Delta t)]$

$x_k = [-(1 - 2 / \Delta t) x_{k-1} + u_k + u_{k-1}] / (1 + 2 / \Delta t)$

Pseudo-code: It's ready to go!

Fact: TF's $X(j\omega) / U(j\omega)$ and $X(q) / U(q)$ have <u>almost the same</u> magnitude and phase plots vs ω when $q = e^{-j\omega\Delta t}$ (up to the Nyquist rate)



Consider vector case: $d\underline{x}/dt = A\underline{x}$, $\underline{x}(0) = \underline{x}_0$

Solution must satisfy $d\underline{x}(0)/dt = A\underline{x}_0$ and $\underline{x}(0) = \underline{x}_0$

Solution is $\underline{x}(t) = e^{At}\,\underline{x}_0$

where the e^{At} is a matrix exponential, obeying many of the rules of the scalar exponential:

$d(e^{At})/dt = Ae^{At}$ $e^{At} = I + At + A^2t/2! + \ldots$

$e^0 = I$ (identity) etc…

Consider the system $d\underline{x}/dt = A\underline{x} + Bu$

$e^{At}B$ is the impulse response of the system such that

$$\underline{x}(t) = e^{At}\,\underline{x}_0 + \int_0^t e^{A(t-\tau)}\,B\,u(\tau)\,d\tau \qquad \text{OR}$$

$$\underline{x}_k = e^{A\Delta t}\,\underline{x}_{k-1} + \int_{tk-1}^{tk} e^{A(tk-\tau)}\,B\,u(\tau)\,d\tau$$

Let u be constant on t_{k-1} to t_k (*zero-order hold*):

$$\underline{x}_k = e^{A\Delta t}\,\underline{x}_{k-1} + \int_{tk-1}^{tk} e^{A(tk-t)}\,d\tau\,B\,u_{k-1}$$

$$\underline{x}_k = \Phi\,\underline{x}_{k-1} + \Gamma\,u_{k-1}$$

1:1 mapping between continuous and discrete-time systems

2.017J Design of Electromechanical Robotic Systems
Fall 2009