Continuous-Time Signals and Systems

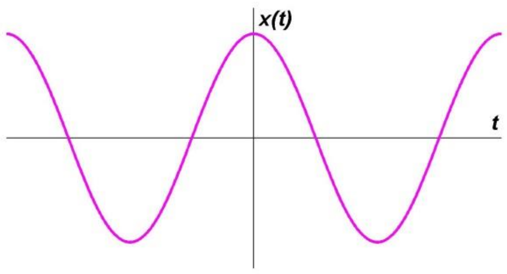
Sinusoidal Amplitude Modulation & Convolution Review

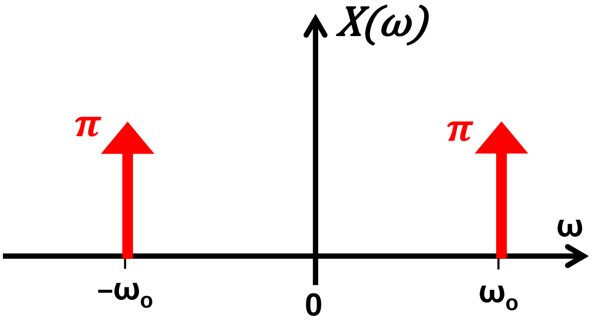
Edward (Ted) A. Clancy

Department of Electrical and Computer Engineering
Department of Biomedical Engineering
Worcester Polytechnic Institute
Worcester, MA U.S.A.

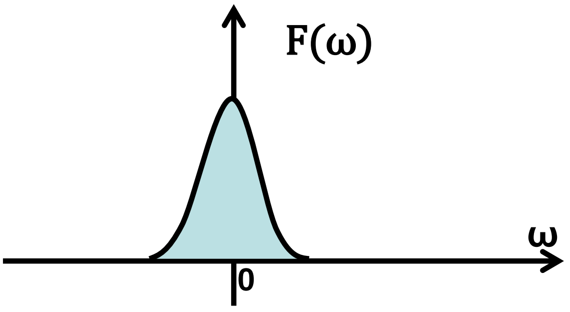
Recall

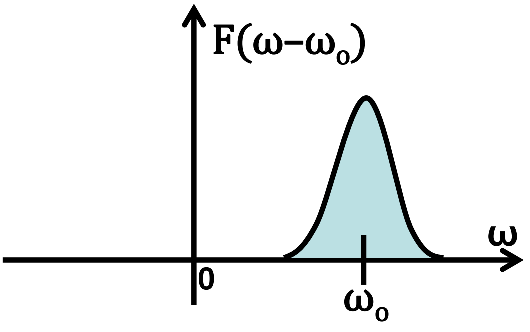
1) $x(t) = \cos(\omega_o\, t) \Leftrightarrow X(\omega) = \pi\, \delta(\omega - \omega_o) + \pi\, \delta(\omega + \omega_o)$





2) $F(\omega) * \delta(\omega - \omega_o) = F(\omega - \omega_o)$





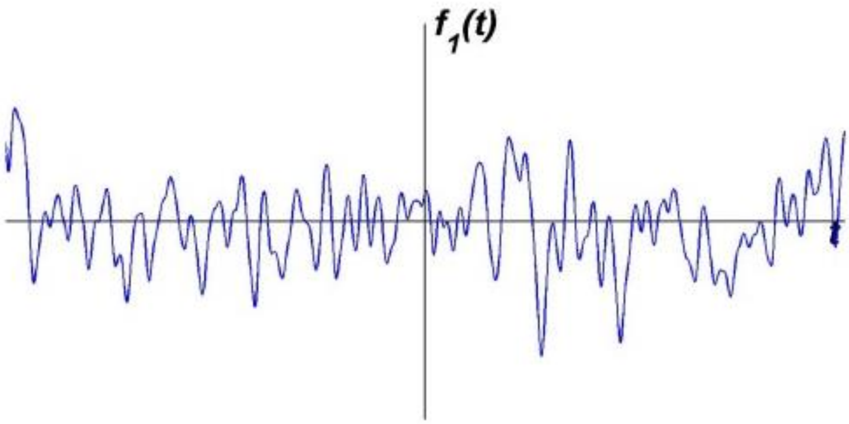
3) Multiplication in time

$\Downarrow$

Convolution in frequency

Baseband Signal

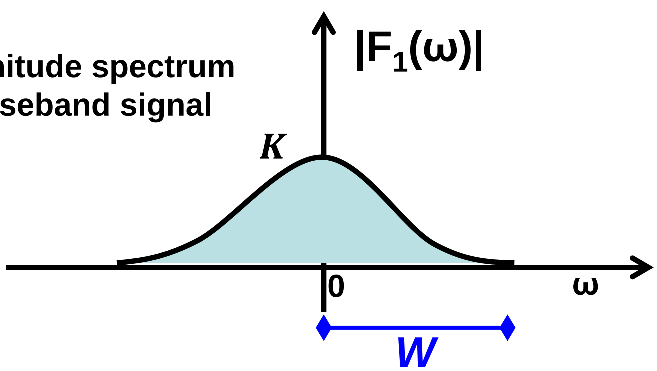
- **Baseband:** **Signal band of original signal**
  - **Generally clustered around DC (ω = 0)**
  - **E.g., natural voice band ≈300 Hz to 3000 Hz**



⇕

Note: Real-valued $f_1(t)$
➔ Symmetric $F_1(\omega)$

Magnitude spectrum of baseband signal

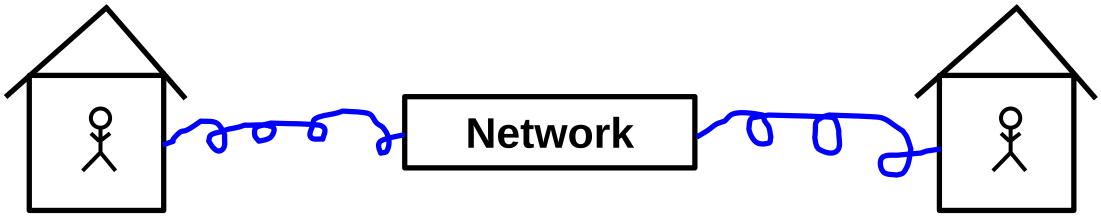


- **$f_1(t)$ can be anything ➔ Not fit a functional form**
  ➔ Model in frequency domain via its bandwidth

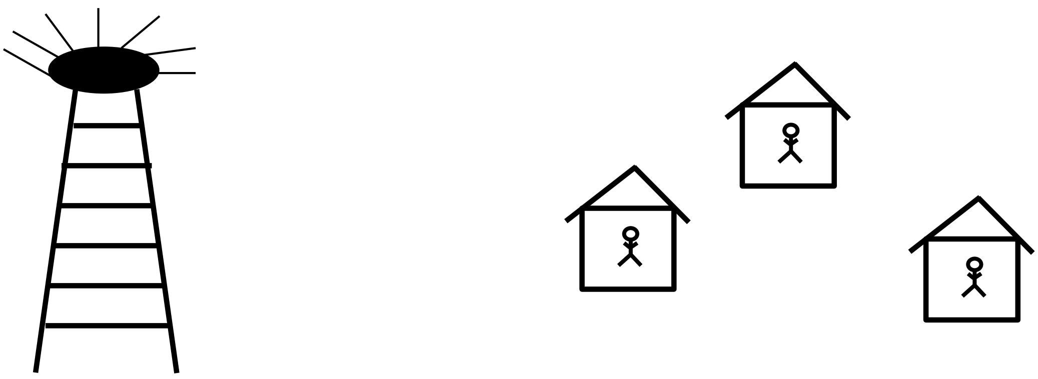
Signal Transmission

- **Goal: Transmit signals from place-to-place**

  – **One-to-one:**
    - **E.g., Telephone (landline or cell)**



  – **One-to-many:**
    - **E.g.**



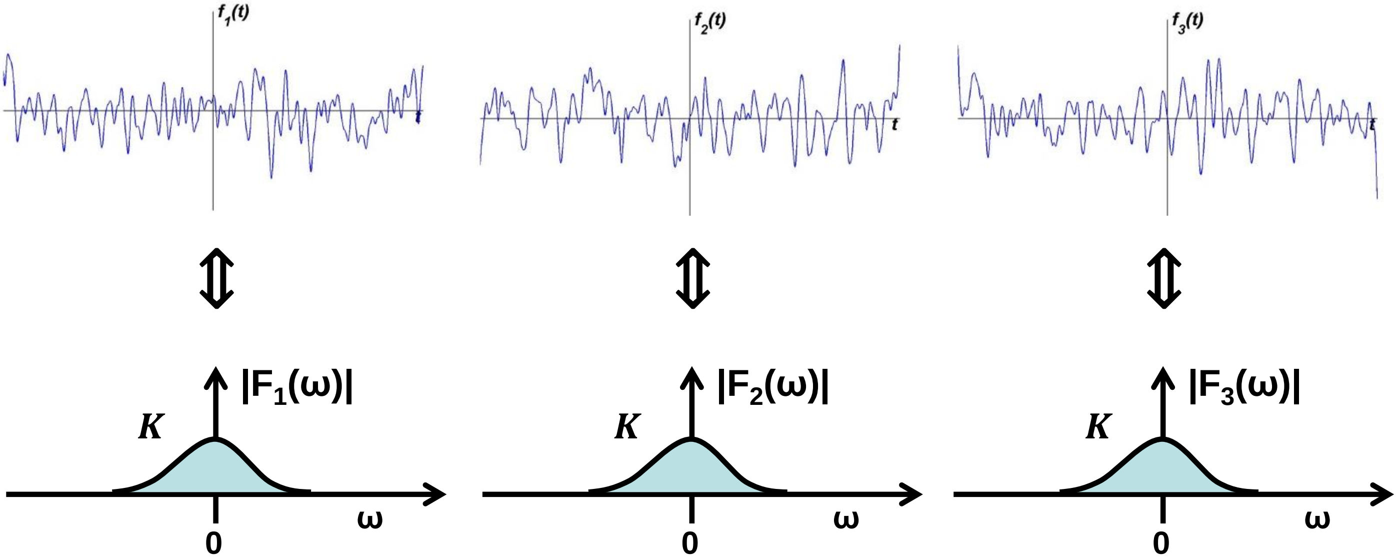
  – **Many-to-many:** **E.g., Conference call**

- **Transmission via wire and/or air ➔ channel**

- **Issue:** **Must transmit multiple signals over same channel**

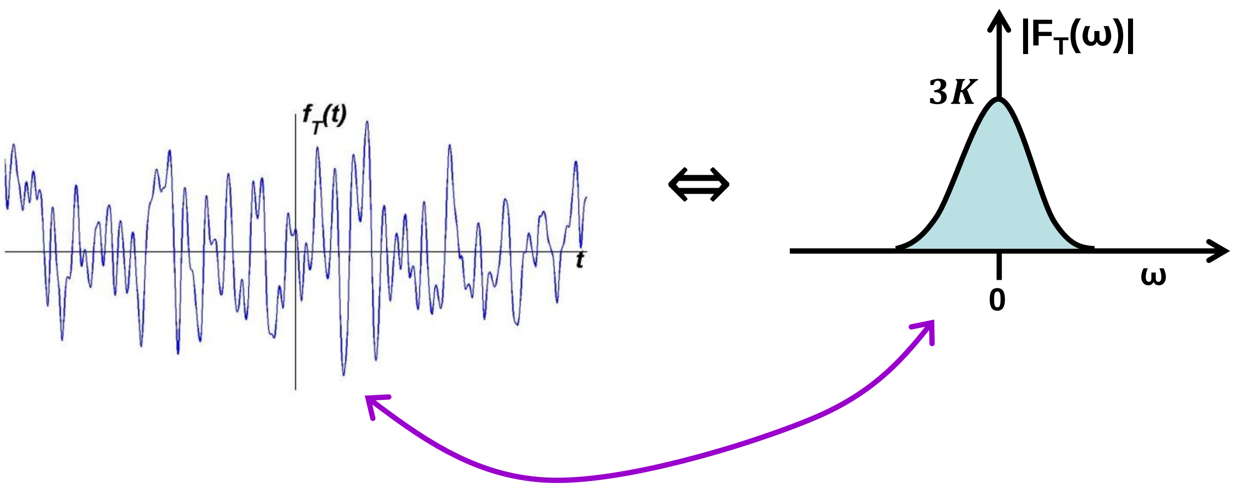
Baseband Signals Interfere

- Consider three baseband voice signals:



- Signals overlap in time <u>and</u> frequency

- No obvious way for multiple baseband signals to share channel
  - E.g.: Consider $f_T(t) = f_1(t) + f_2(t) + f_3(t)$



Cannot resolve $f_i(t)$

Antennae for Baseband Signals

- Wireless transmission (e.g., radio, TV, cell phone, Bluetooth, …)
  - **Requires** antenna $length \geq \frac{signal\, wavelength}{10}$,

    where $v = f\,\lambda$

    Velocity: ~ v = 3x10^8 m/s

    Signal frequency (Hz)

    Wavelength (m)

  - Thus,

$$Antenna\ Length\!: \ L \geq \frac{v}{10 \cdot f}$$

- Consider voice: 300 Hz → 3000 Hz

$$L_1 = \frac{3 \cdot 10^8\ m/s}{10 \cdot 300/s} = 100\ km$$

$$L_2 = \frac{3 \cdot 10^8\ m/s}{10 \cdot 3000/s} = 10\ km$$

Too long !!!

- Must find other way to share channel
- Wireless systems usually not able to transmit at baseband

- MANY POSSIBLE SOLUTIONS

Signal Modulation — 1

- **Baseband signal *m(t)***
  - **Characterize with bandwidth over *[–W, W]***

- **Form:**

$$s(t) = m(t) \cdot A_c \cos(\omega_c t)$$

$$\updownarrow$$

$$S(\omega) = \frac{1}{2\pi} \cdot M(\omega) * A_c \cdot \pi \left[\delta(\omega - \omega_c) + \delta(\omega + \omega_c)\right]$$

Carrier gain

Carrier frequency

- **Assume** $\omega_c \gg W$

$$S(\omega) = \frac{A_c}{2}\left[M(\omega - \omega_c) + M(\omega + \omega_c)\right]$$

Recall:

$$x_1(t) \cdot x_2(t) \Leftrightarrow \frac{1}{2\pi} X_1(\omega) * X_2(\omega)$$

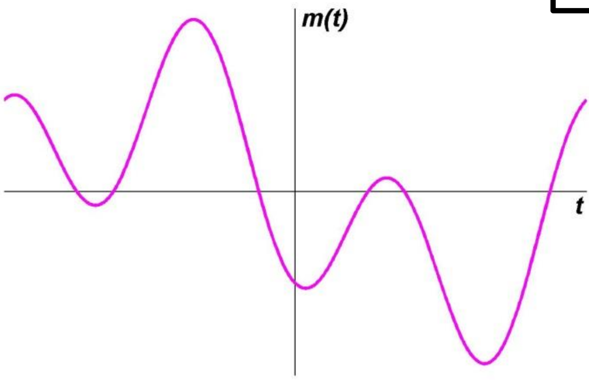
Continued

Signal Modulation — 2

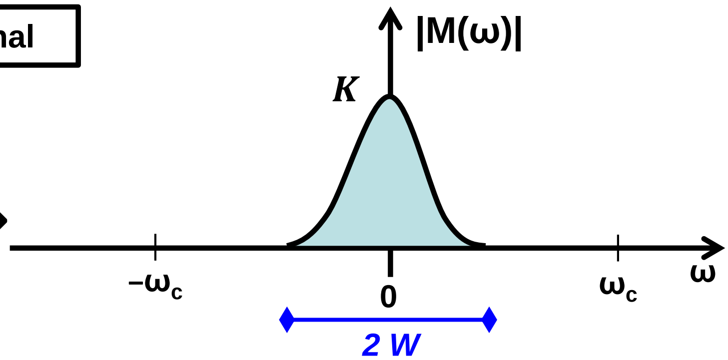
Time Domain:

Frequency Domain:

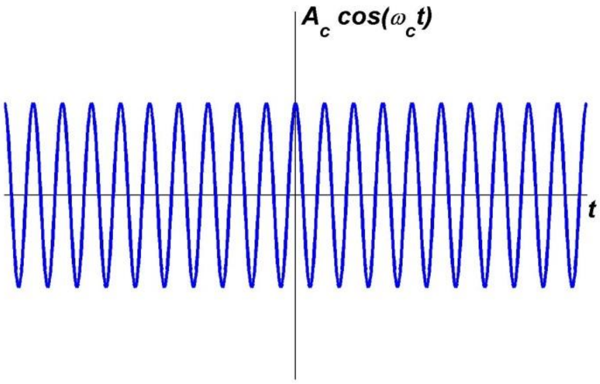
Signal



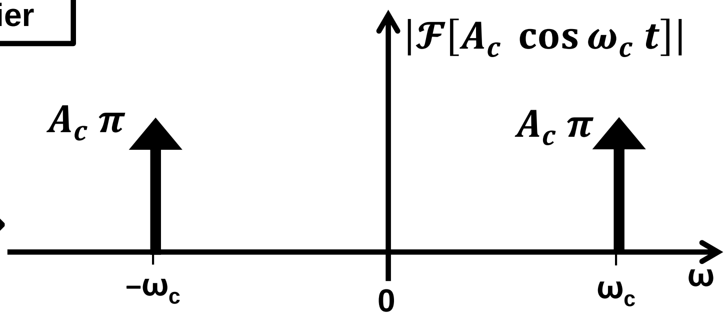
⇔



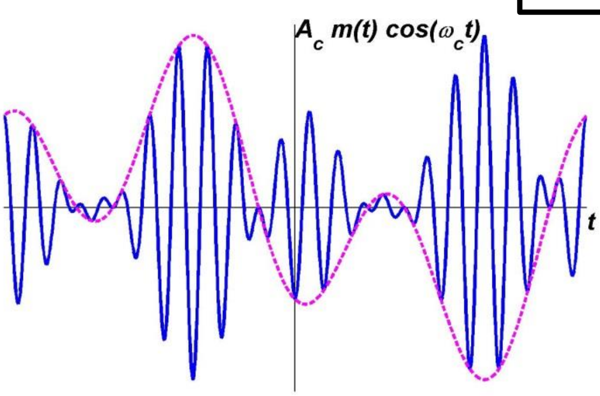
Carrier



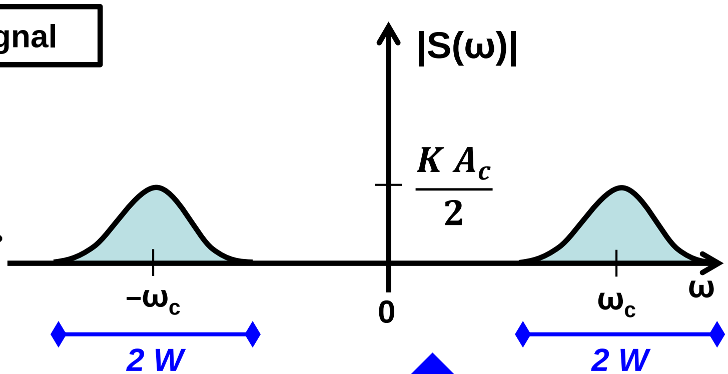
⇔



AM Signal



⇔



NOTES:

1. Amplitude halved
2. Bandwidth doubled

Frequency-Division Multiplexing

- **Example:** Two signals $m_1(t)$, $m_2(t)$

- **To share a channel:**

  - **Modulate $m_1(t)$ at carrier frequency $\omega_{c,1}$**

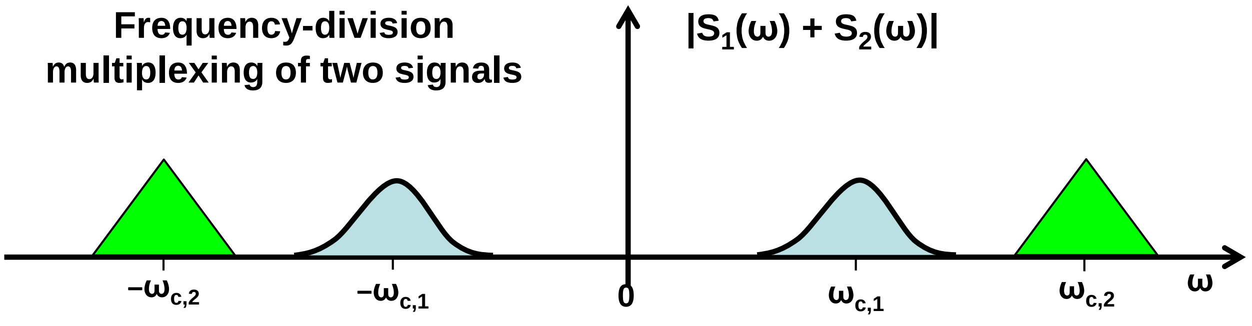
$$s_1(t) = m_1(t) \cdot A_{c,1} \cos(\omega_{c,1}\, t)$$
$$\updownarrow$$
$$S_1(\omega) = \frac{1}{2\,\pi} \cdot M_1(\omega) * A_{c,1} \cdot \pi \cdot \left[\delta(\omega - \omega_{c,1}) + \delta(\omega + \omega_{c,1})\right]$$

$$= \frac{1}{2} \cdot M_1(\omega) * A_{c,1} \cdot \left[\delta(\omega - \omega_{c,1}) + \delta(\omega + \omega_{c,1})\right]$$

  - **Modulate $m_2(t)$ at $\omega_{c,2} \neq \omega_{c,1}$**

  - **Transmit:**

$$s_T(t) = s_1(t) + s_2(t)$$
$$\updownarrow$$
$$S_T(\omega) = S_1(\omega) + S_2(\omega)$$

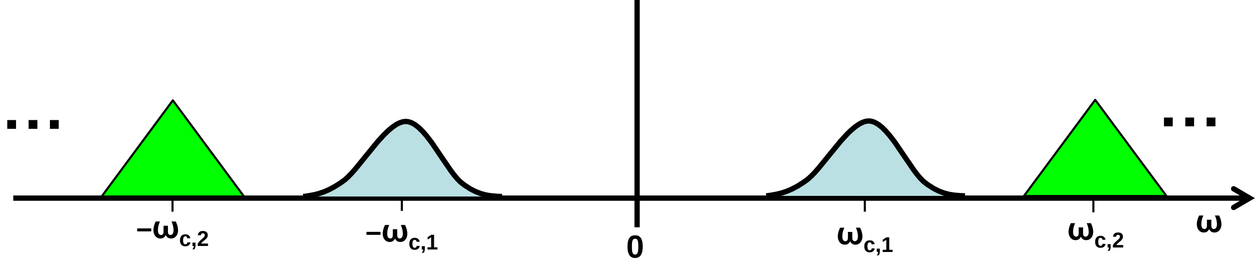


  - **Select $\omega_{c,1}$, $\omega_{c,2}$ to avoid spectral overlap**
  - **Extend to many signals (E.g., radio/TV stations)**

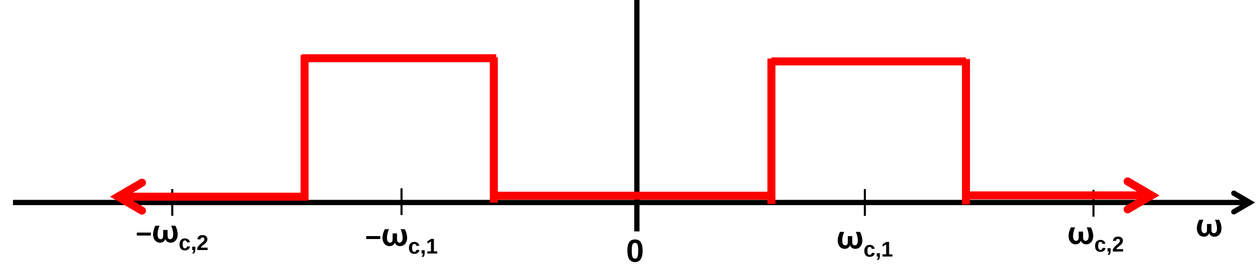
Receiver Bandpass Filter

• Common receiver step ➔ Bandpass select desired frequency band

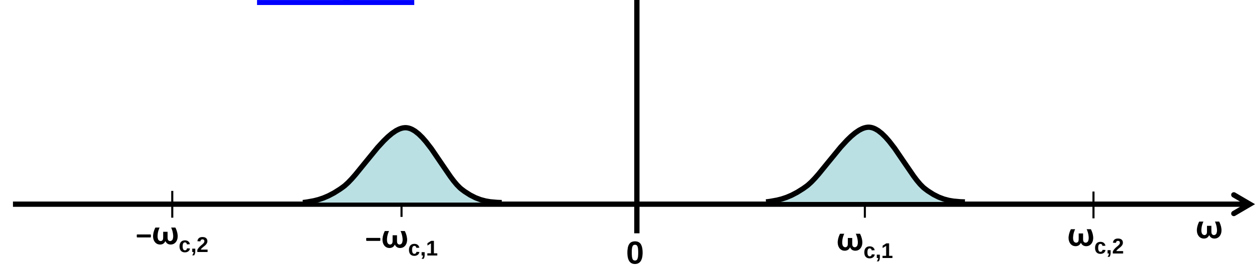
Receive

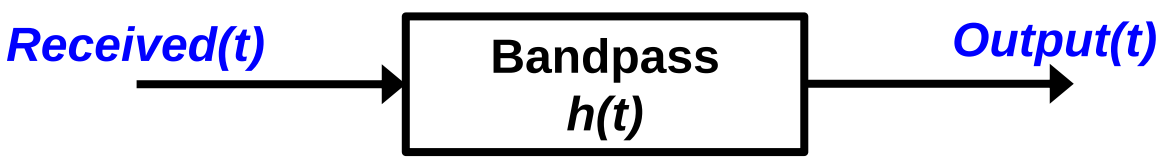


Bandpass Filter



Output

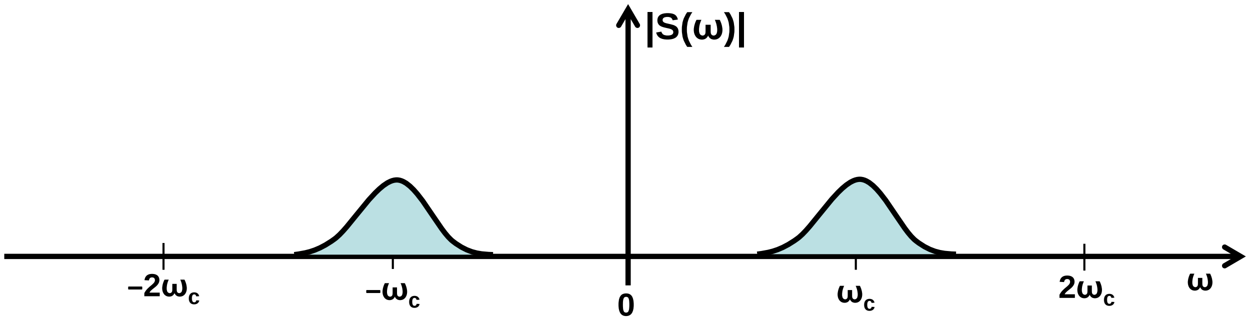




Coherent Detection (Product Detection) — 1

- After bandpass filter, have received:

$$s(t) = m(t) \cdot A_c \cos(\omega_c\, t)$$
$$\updownarrow$$
$$S(\omega) = \frac{A_c}{2}\,[M(\omega - \omega_c) + M(\omega + \omega_c)]$$



- Multiply again by: $A_r \cos(\omega_c\, t)$, giving:

$$r(t) = s(t) \cdot A_r \cos(\omega_c\, t) = m(t) \cdot A_c \cos(\omega_c\, t) \cdot A_r \cos(\omega_c\, t)$$

- Time Domain:

$$r(t) = A_c\, A_r \cdot m(t) \cdot \cos^2(\omega_c\, t)$$

$$= A_c\, A_r \cdot m(t) \cdot \left[\frac{1 + \cos(2\,\omega_c\, t)}{2}\right]$$

$$= \underbrace{\frac{A_c\, A_r}{2} \cdot m(t)}_{\text{Desired (scaled) baseband signal}} + \underbrace{\frac{A_c\, A_r}{2} \cdot cos(2\,\omega_c\, t) \cdot m(t)}_{\text{Undesired signal at frequency } 2\,\omega_c}$$

Continued

Coherent Detection (Product Detection) — 2

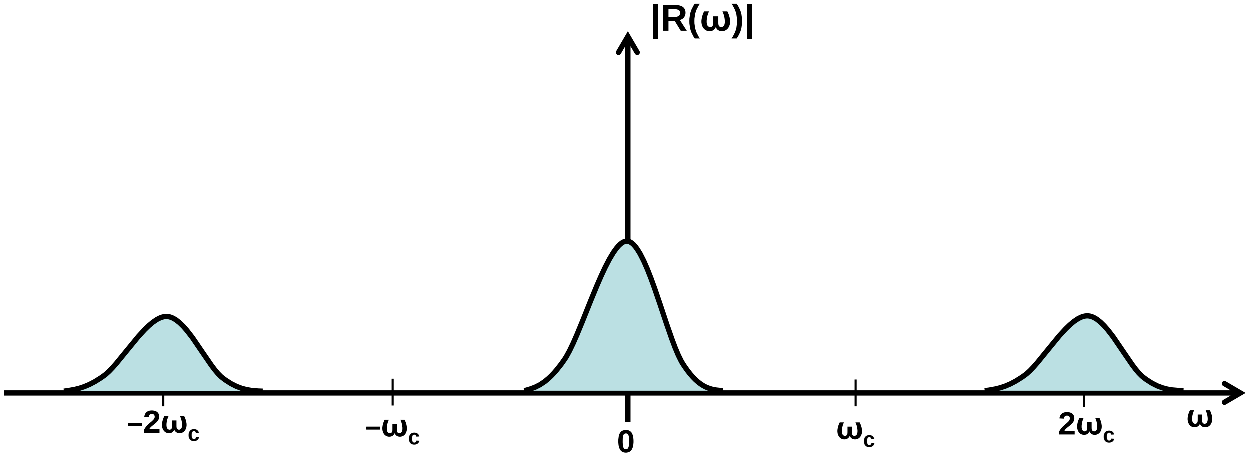
• **Frequency Domain:**

$$R(\omega) = \frac{1}{2\,\pi} S(\omega) * A_r \cdot \pi \left[\delta(\omega - \omega_c) + \delta(\omega + \omega_c)\right]$$

$$= \frac{A_r}{2} \cdot \left[S(\omega - \omega_c) + S(\omega + \omega_c)\right]$$

$$= \frac{A_c \cdot A_r}{4} \cdot \{[M(\omega - 2\omega_c) + M(\omega)] + [M(\omega) + M(\omega + 2\omega_c)]\}$$

$$= \underbrace{\frac{A_c \cdot A_r}{2} \cdot M(\omega)}_{\text{Desired (scaled) baseband signal}} + \underbrace{\frac{A_c \cdot A_r}{4} \cdot [M(\omega - 2\omega_c) + M(\omega + 2\omega_c)]}_{\text{Undesired signal at frequencies } \pm 2\,\omega_c}$$



Magnitude spectra of AM signal after second multiplication with carrier.

• **Thus, LOWPASS FILTER to recover** $M(\omega) \Leftrightarrow m(t)$

ISSUE: Receiver and transmitter cosine must have same frequency & phase !!!

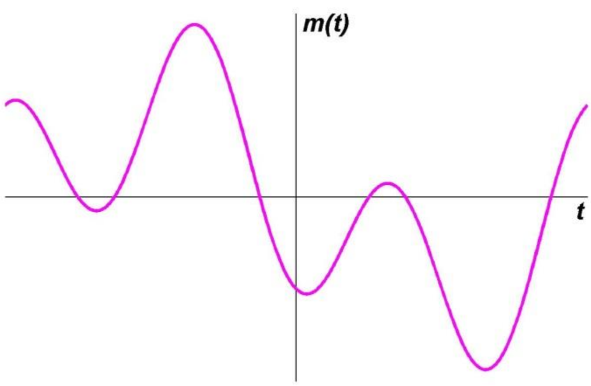
Non-Coherent (Envelope) Detection

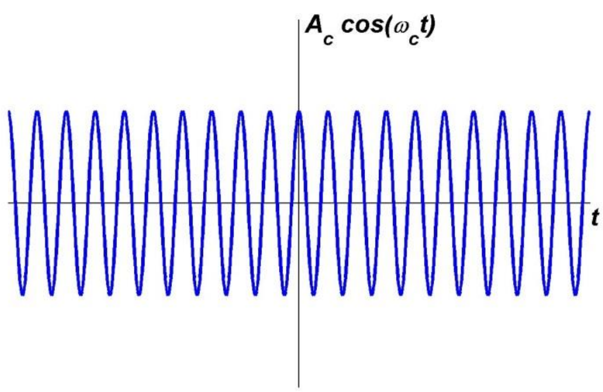
- **Coherent detection**
  - Harder (phase- and frequency-matched mixer)
  - More expensive
  - But, recovers full signal
  - ➔Some schemes transmit carrier frequency, phase in real time

- **Non-coherent detection**
  - Simple, cheap
  - But, only recover $|m(t)|$

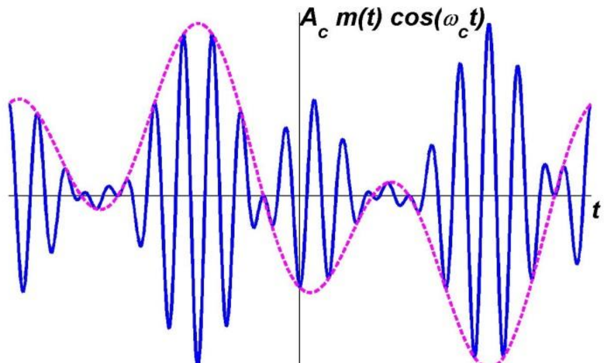
Recall: Amplitude modulation in time domain:

Signal x Carrier





= Transmitted Signal

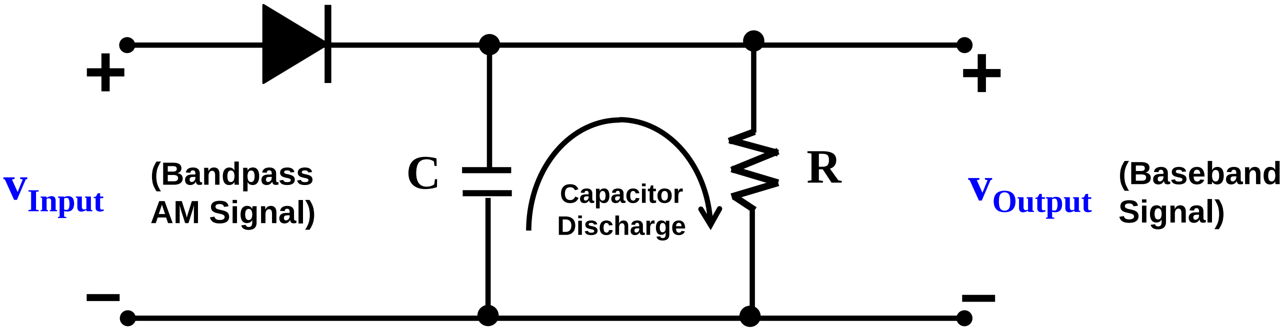


Signal envelope gives $|m(t)|$

Sinusoidal Amplitude Modulation

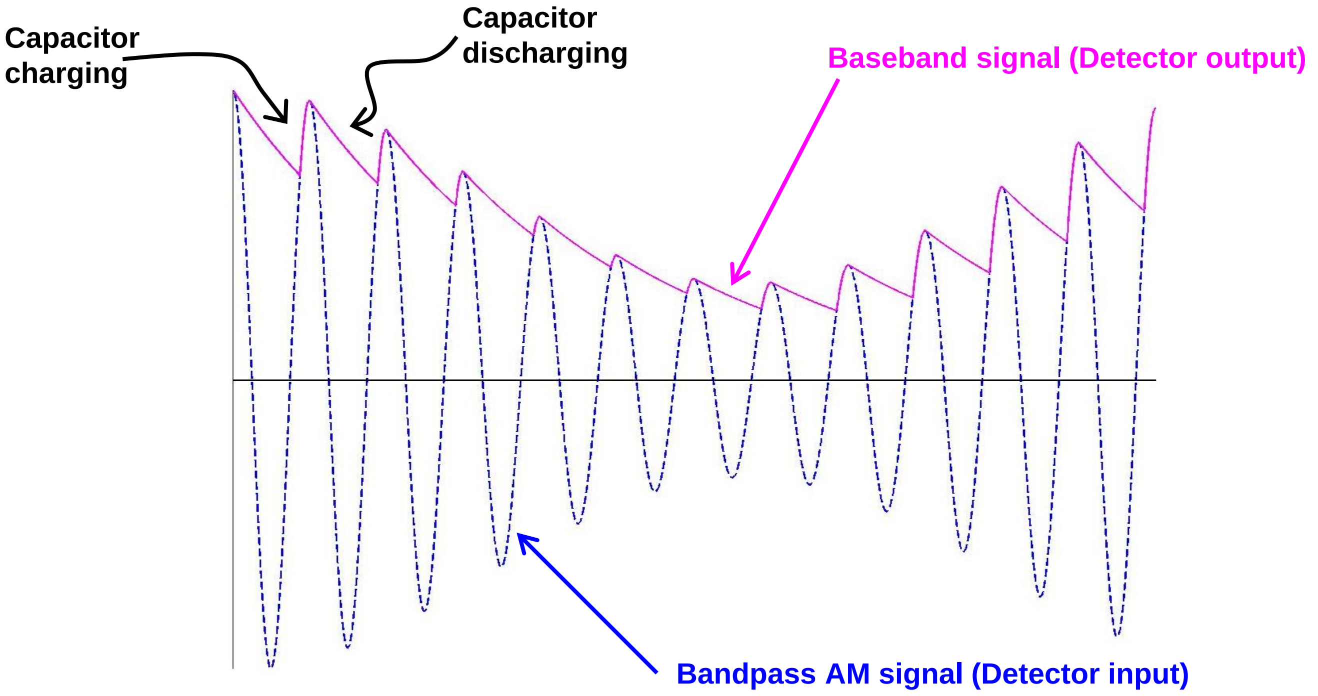
Envelope Detection Circuit

- Simple electronic circuit to detect envelope
  - Found in simple AM radio receivers:



Envelope detection circuit

- Sample Input-Output signals: AM signal & envelope-detected signal



- Detected signal approximates $|m(t)|$
  - So, add offset to m(t) to make signal non-negative
- Some ripple at carrier frequency
  - Can lowpass filter detector output to reduce ripple

Amplitude Modulation—Epilogue

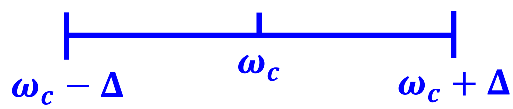
- **Many amplitude modulation schemes available**

- **Commodity applications fostered by standards**

- **Another modulation genre: Frequency modulation**
  - **Signal:** $m(t)$
  - **Carrier frequency:** ω_c
  - **Transmit:**

$$s(t) = A_c \cos(\,\omega_c\, t + f[m(t)]\,)$$

- Signal varies its transmit frequency over a band



  - **For example:**
    - **Limit** $-1 \le m(t) \le 1$
    - **Let**

$$s(t) = A_c \cos[\omega_c\, t + k \cdot m(t)]$$

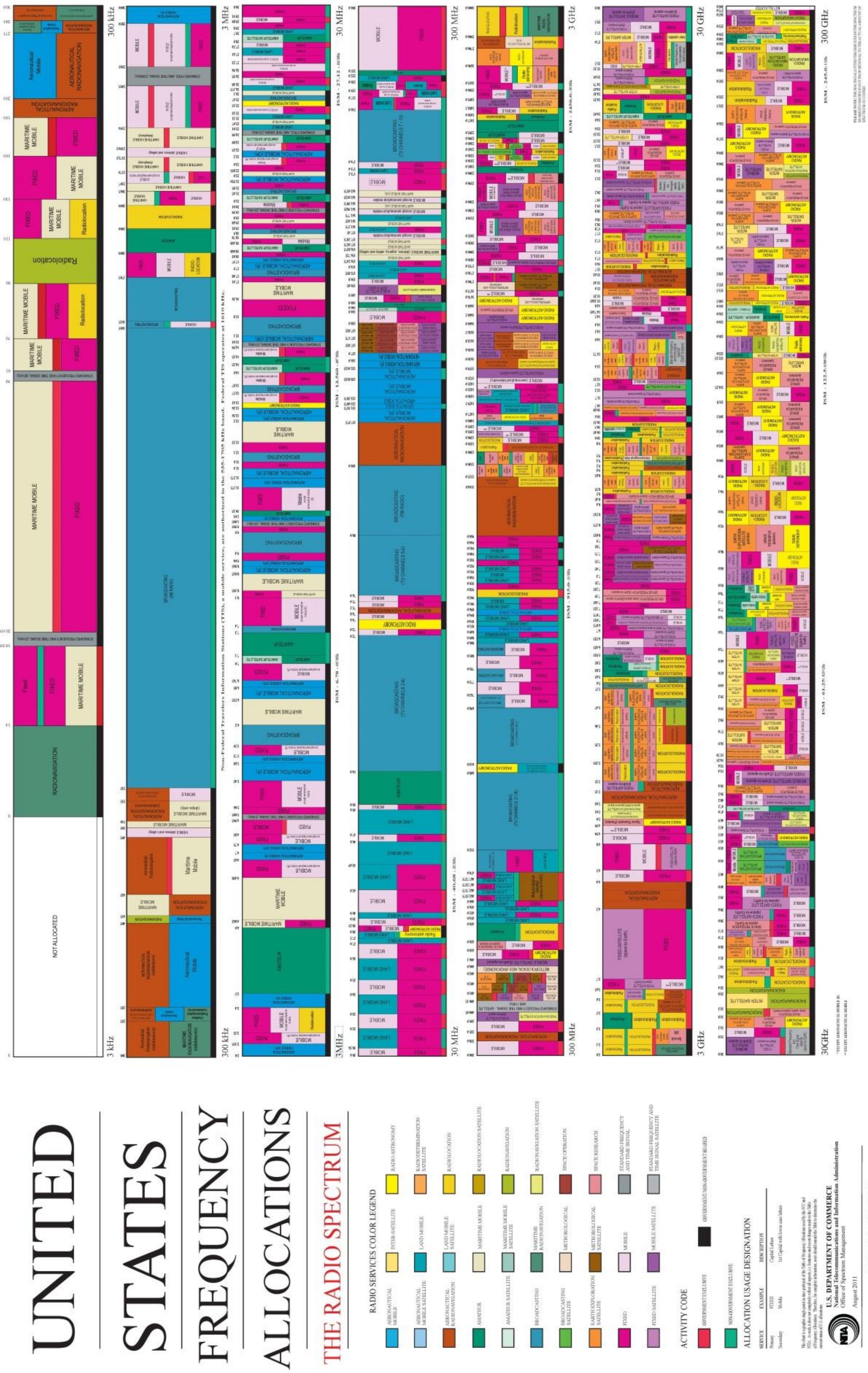
k: Scalar

Instantaneous frequency ≡ Time derivative of cosine argument

Here, $\frac{d}{dt}\left(\omega_c\, t + k \cdot m(t)\right) = \omega_c + k \cdot \frac{d}{dt} m(t)$

– **Instantaneous frequency can be outside of [ω_c–k, ω_c+k]**

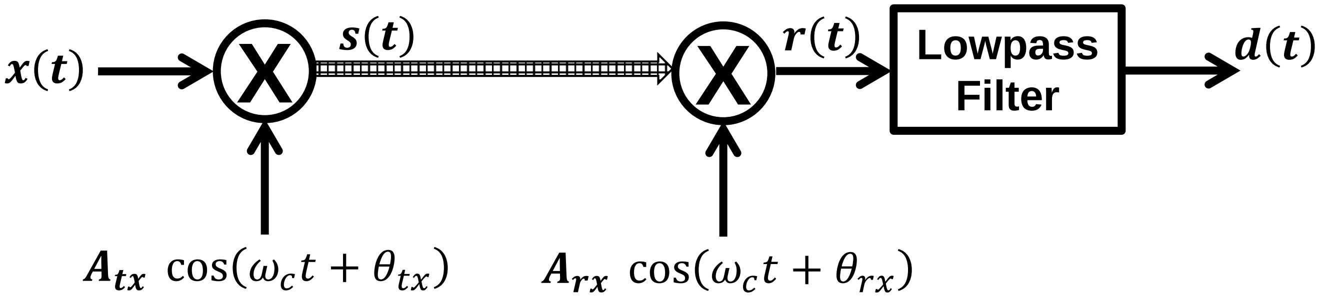
U.S. Frequency Allocations Chart



Course Evaluations

- **As the schedule permits, students should be asked to complete Course Evaluation at this time.**

Sine Amp Mod—Phase Matching



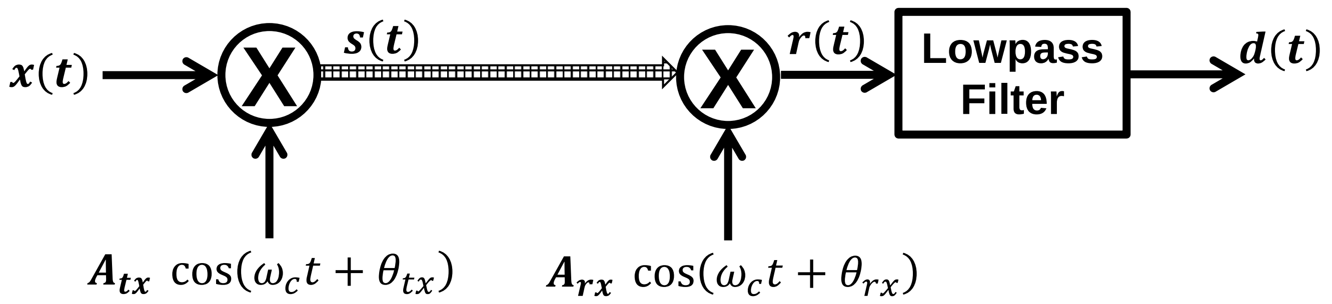
Phases may not match!!

- Original signal to transmit: $x(t)$
- Transmit (TX) modulation signal: $A_{tx}\ \cos(\omega_c t + \theta_{tx})$
  - ➔ Transmitted signal: $s(t) = A_{tx}\, x(t)\ \cos(\omega_c t + \theta_{tx})$
- Receive (RX) modulation signal: $A_{rx}\ \cos(\omega_c t + \theta_{rx})$
  - ➔ Received signal: $r(t) = A_{tx}\, A_{rx}\ x(t)\ \cos(\omega_c t +$

Sum frequency ➔ Remove via lowpass filter

Phase mismatch applies gain between –1 and 1

Sine Amp Mod—Phase Matching Demonstration



From above: $d(t) = \frac{A_{tx}A_{rx}}{2}\cos(\theta_{tx} - \theta_{rx})\,x(t)$

- **MATLAB (audio) demonstration:** **handel_demo_phase.m**
  - **Launch MATLAB, navigate to directory with demo**
  - \>> handel_demo_phase

    - **Function will play 8–9 s duration of Handel's Messiah**
    - **Vary phase from 0 to 90 degrees (via screen prompt)**
      - **Function plays "Messiah" with phase mismatch**
      - **Full gain at 0 degrees**
      - **Gain = 0 at 90 degrees**

- **Mismatched phase lowers demodulated signal strength**
  - **Noise becomes more problematic (lower SNR)**
  - **Might need to boost transmit power**

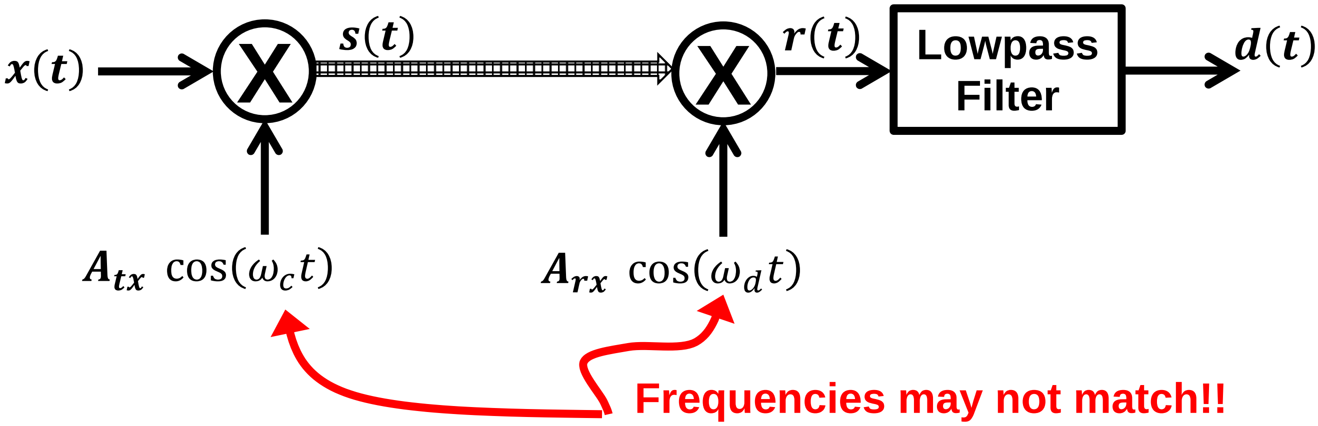
  - **One method to lock phase of receiver to transmitter ➔ Phase locked loop**

Sine Amp Mod—Phase Matching Demonstration Code

- **Code for MATLAB demo from previous slide:**

```
function handel_demo_phase

Fs = []; y = []; load handel.mat; % Matlab demo of Handel's Messiah (~9 s).
% To play: sound(y,Fs)  % Fs = 8192 Hz.
display('Listen to short clip of Handel''s Messiah');
sound(y, Fs); pause(8);

while 1==1
  % Request phase error (Degrees) in detector.
  Theta = input('Input detector phase error (in degrees): ');
  % Compute revised received Handel clip.
  y2 = cos(Theta*pi/180) * y;
  % Play revised received signal.
  sound(y2, Fs); pause(8);
end

return
```

Sine Amp Mod—Frequency Matching



- **Original signal to transmit:** $x(t)$
- **Transmit (TX) modulation signal:** $A_{tx}\ \cos(\omega_c t)$
  - ➔ **Transmitted signal:** $s(t) = A_{tx}\, x(t)\ \cos(\omega_c t)$
- **Receive (RX) modulation signal:** $A_{rx}\ \cos(\omega_d t)$
  - ➔ **Received signal:**

$$r(t) = A_{tx}\, A_{rx}\ x(t)\ \cos(\omega_c t) \cos(\omega_d t)$$

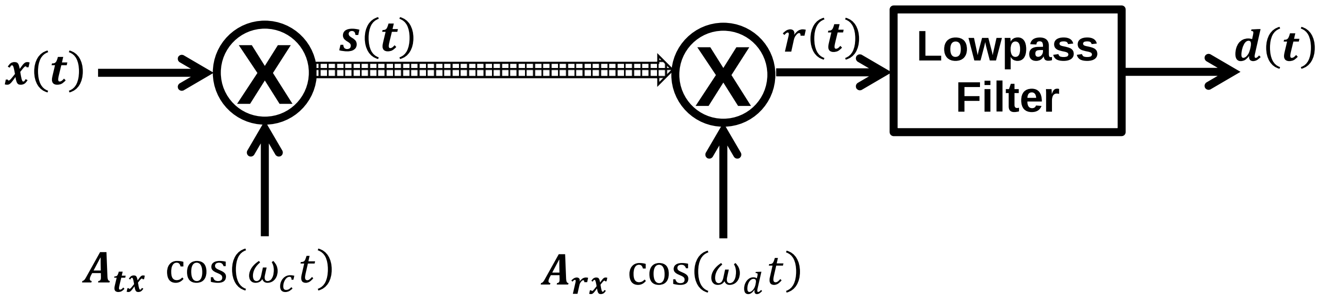
- **Recall:** $\cos(\alpha) \cdot \cos(\beta) = \frac{1}{2}[\cos(\alpha - \beta) + \cos(\alpha + \beta)]$, **giving:**

$$r(t) = \frac{A_{tx} A_{rx}}{2} x(t) \cos(\omega_c t - \omega_d\, t) + \underbrace{\cos(\omega_c t + \omega_d\, t)}_{\text{Sum frequency ➔ Remove via lowpass filter}}$$

$$d(t) = \frac{A_{tx} A_{rx}}{2}\ \cos\big((\omega_c - \omega_d)\ t\big)\ x(t)$$

Frequency mismatch applies <u>TIME-VARYING</u> gain between –1 and 1

Sine Amp Mod—Frequency Matching Demonstration



From above: $$d(t) = \frac{A_{tx}A_{rx}}{2}\ cos\big((\omega_c - \omega_d)\ t\big)\ x(t)$$

- **MATLAB (audio) demonstration:** **handel_demo_freq.m**
  - **Launch MATLAB, navigate to directory with demo**
  - >> handel_demo_freq

    - **Function will play 8–9 s duration of Handel's Messiah**
    - **Vary frequency mismatch (via screen prompt)**
      - **Function plays "Messiah" with frequency mismatch**
      - **No mismatch at 0 Hz TX-RX difference**
      - **Audible differences at 1, 2, 3, 10, 100, 1000 Hz**

- **Mismatched frequency distorts signal**
  - **Not simply a problem of low gain.**
  - **Time-varying distortion**

  - **One method to lock frequency of receiver to transmitter ➔ Phase locked loop**

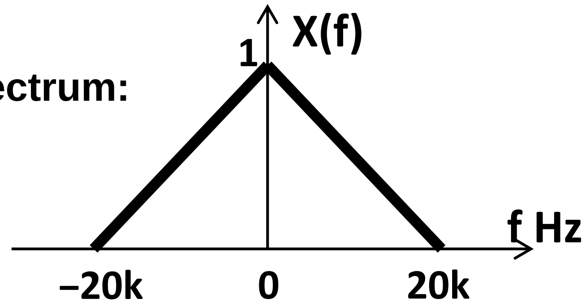
Sine Amp Mod—Frequency Matching Demonstration Code

- **Code for MATLAB demo from previous slide:**

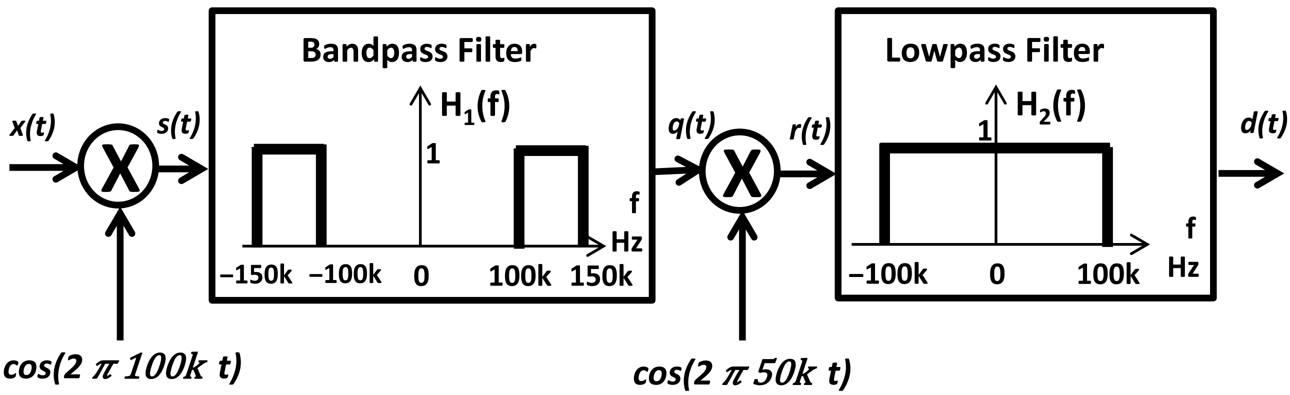
```
function handel_demo_freq

Fs = []; y = []; load handel.mat; % Matlab demo of Handel's Messiah (~9 s).
% To play: sound(y,Fs)   % Fs = 8192 Hz.
display('Listen to short clip of Handel''s Messiah');
sound(y, Fs); pause(8);

% Creat time vector.
t = ( 0:(length(y)-1) ) / 8192;

while 1==1
  % Request receive mixer frequencies in Hz.
  DeltaF = input('Input TX-RX frequency DIFFERENCE (Hz): ');
  % Compute revised received Handel clip.
  y2 = cos(2*pi*DeltaF*t) .* y';
  % Play revised received signal.
  sound(y2, Fs); pause(8);
end

return
```

Sinusoidal Amplitude Modulation: Example 1

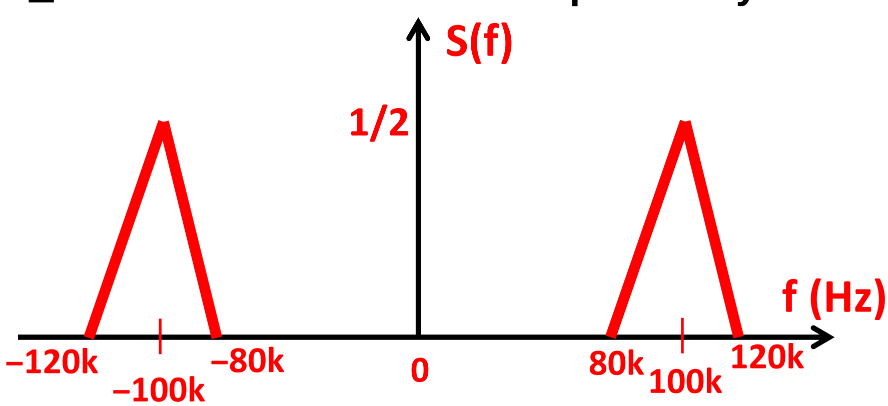
- Consider the signal *x(t)* with spectrum:



- **Let** *x(t)* be processed through the system shown below to produce output signal *d(t)*.
  - **Draw** the spectrum of signals *s(t)*, *q(t)*, *r(t)* and *d(t)*



- **Solution:**
  1. First mixer creates two copies of original spectrum, shifted ±100k Hz and reduced in amplitude by half:

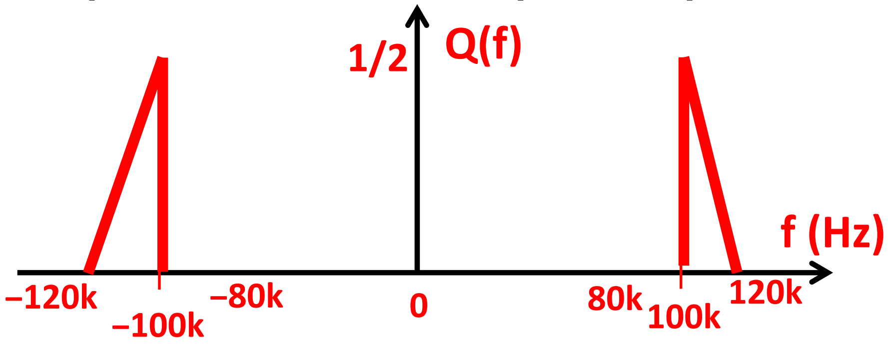


Continued

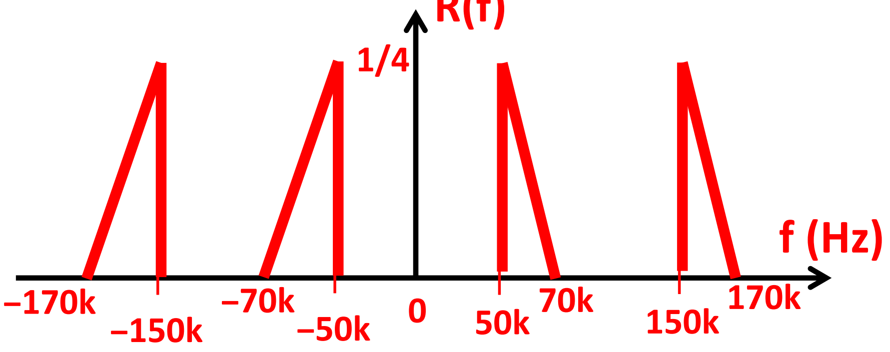
Sinusoidal Amplitude Modulation: Example 1 (Continued)

- **Solution (Continued):**

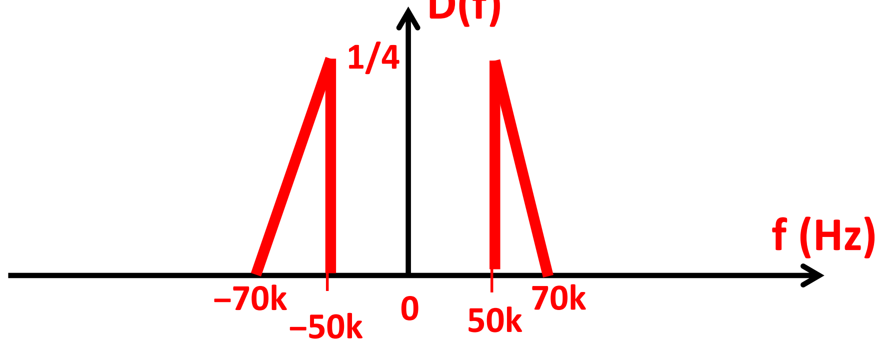
2. **Bandpass filter eliminates spectrum portion below 100k Hz:**



3. **Second mixer shifts BOTH spectral portions by ±50k Hz and reduces amplitude in half:**

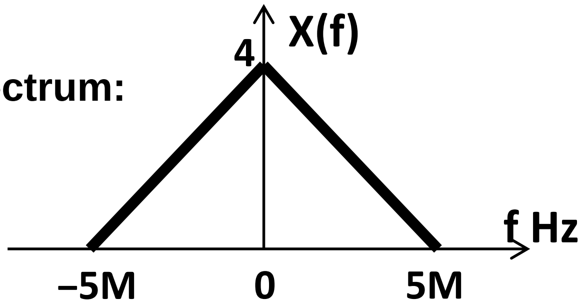


4. **Lowpass filter eliminates the higher frequencies portions:**

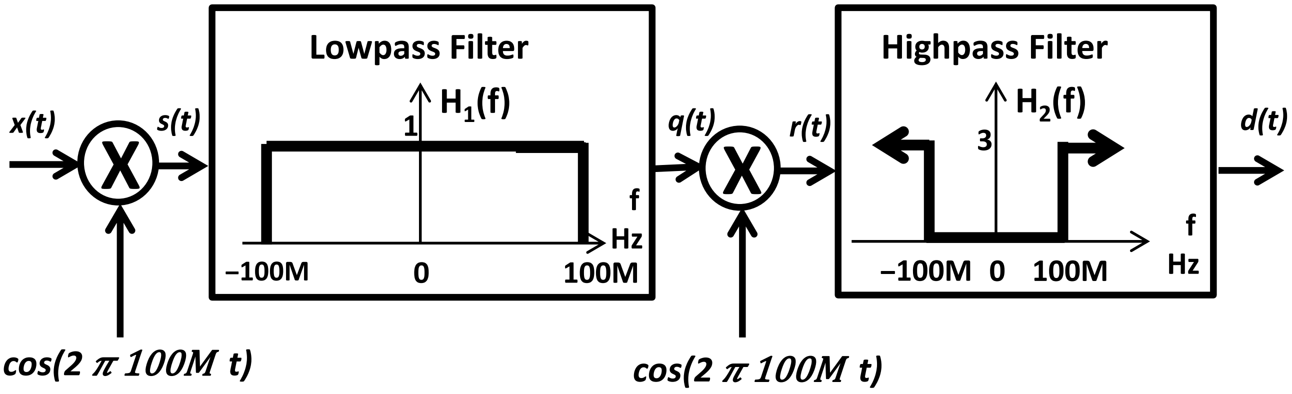


Sinusoidal Amplitude Modulation: Example 2

- Consider the signal *x(t)* with spectrum:

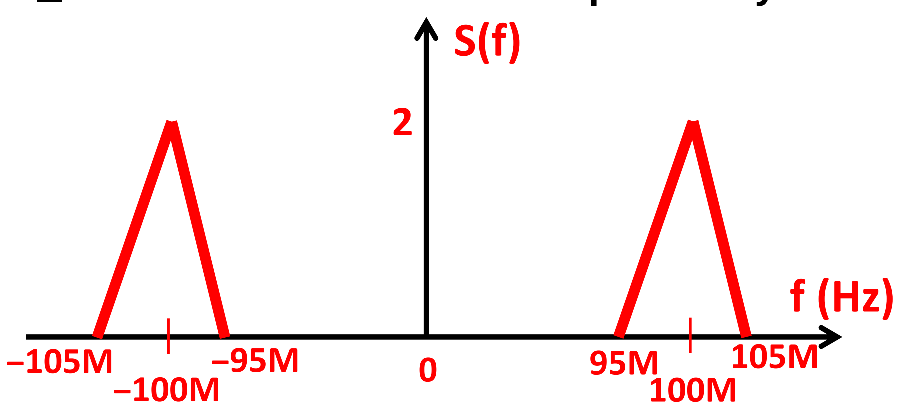


- Let *x(t)* be processed through the system shown below to produce output signal *d(t)*.
  - Draw the spectrum of signals *s(t)*, *q(t)*, *r(t)* and *d(t)*



- **Solution:**

1. First mixer creates two copies of original spectrum, shifted ±100M Hz and reduced in amplitude by half:

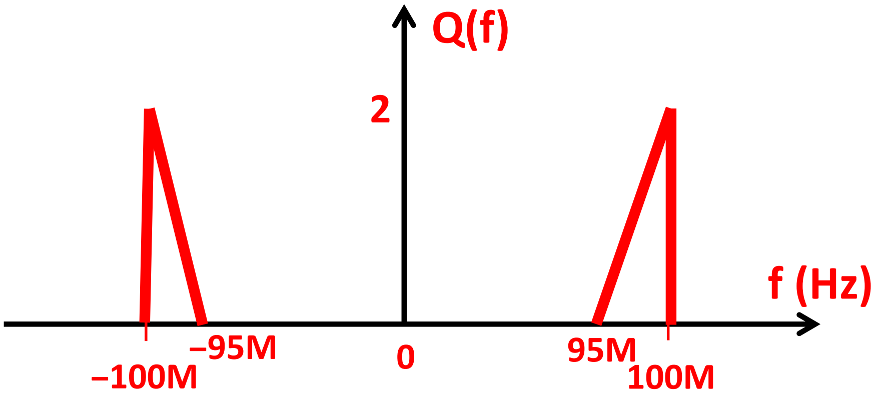


Continued

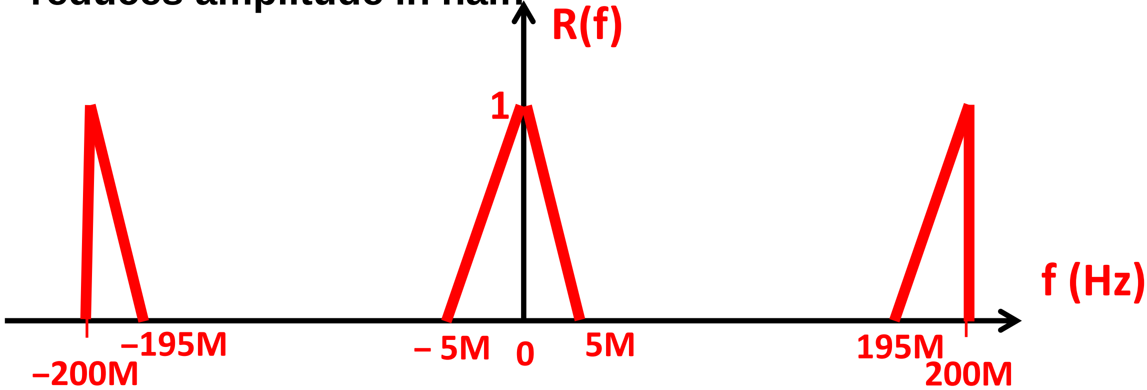
Sinusoidal Amplitude Modulation: Example 2 (Continued)

• **Solution (Continued):**

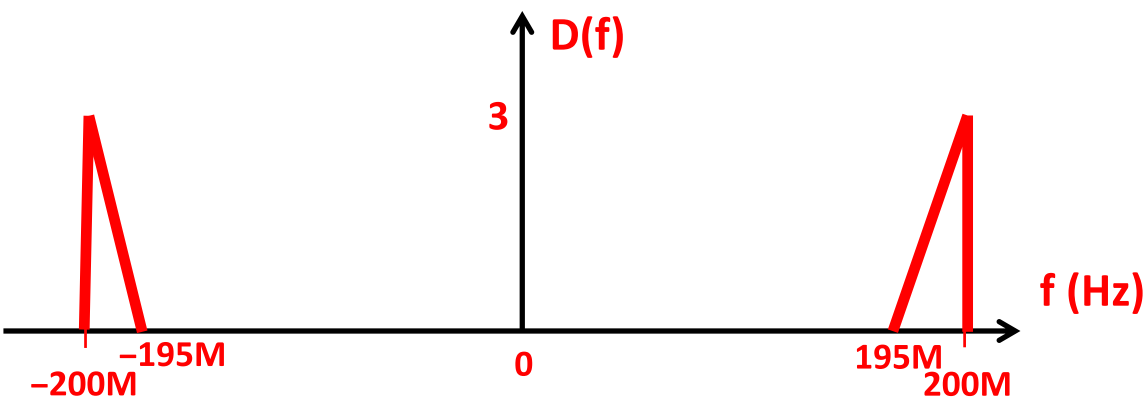
2. Lowpass filter eliminates spectrum portion above 100M Hz:



3. Second mixer shifts BOTH spectral portions by $\pm$100M Hz and reduces amplitude in half:

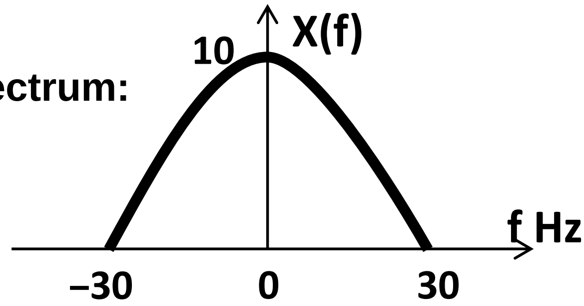


4. Highpass filter retains higher frequencies portions (gain of 3):

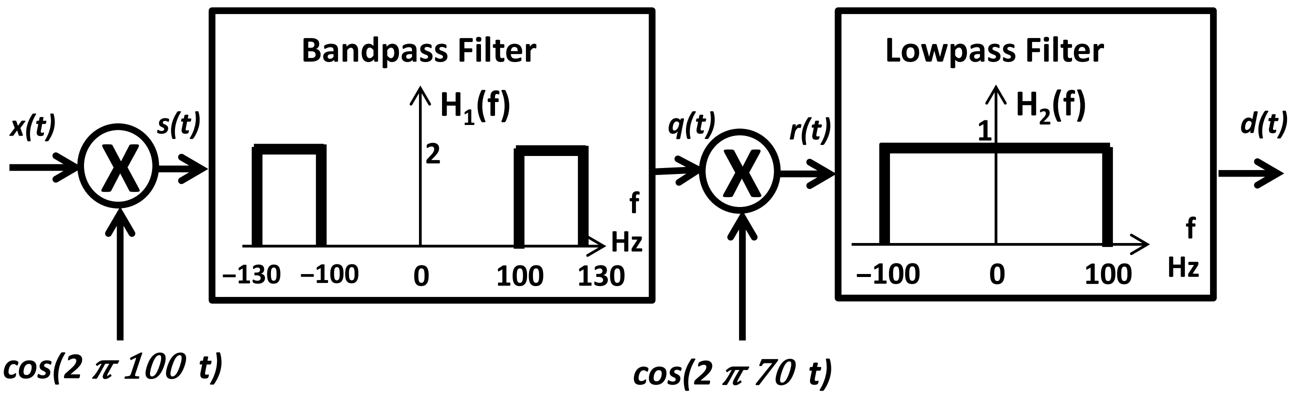


Sinusoidal Amplitude Modulation: Example 3

- Consider the signal *x(t)* with spectrum:

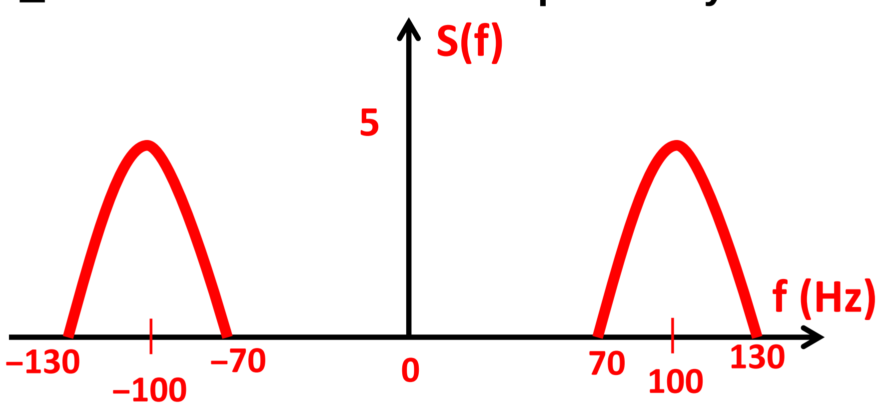


- Let *x(t)* be processed through the system shown below to produce output signal *d(t)*.
  - Draw the spectrum of signals *s(t)*, *q(t)*, *r(t)* and *d(t)*



- **Solution:**

1. First mixer creates two copies of original spectrum, shifted $\pm$100 Hz and reduced in amplitude by half:

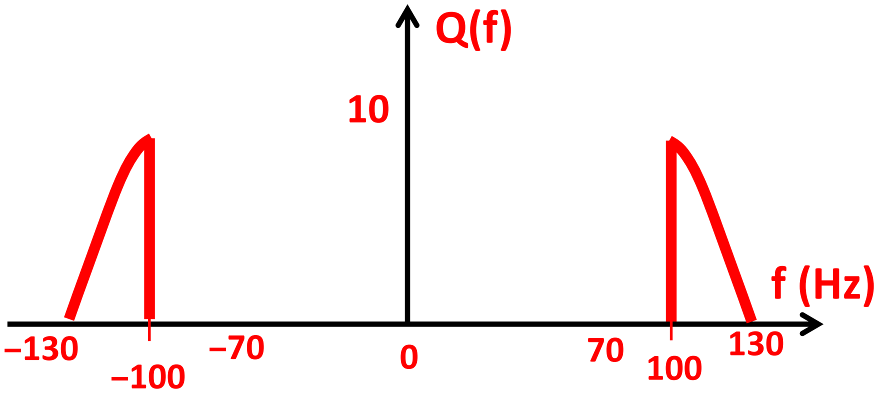


Continued

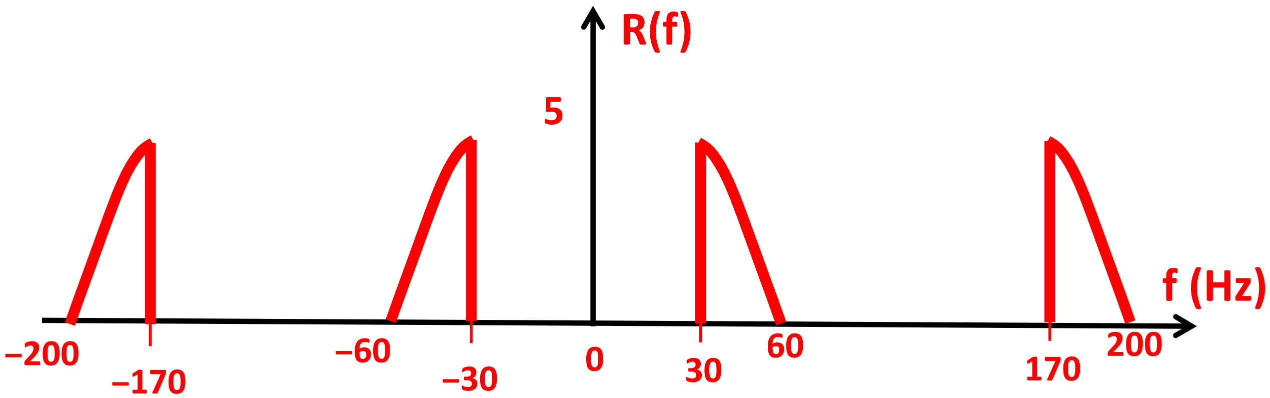
Sinusoidal Amplitude Modulation: Example 3 (Continued)

- **Solution (Continued):**

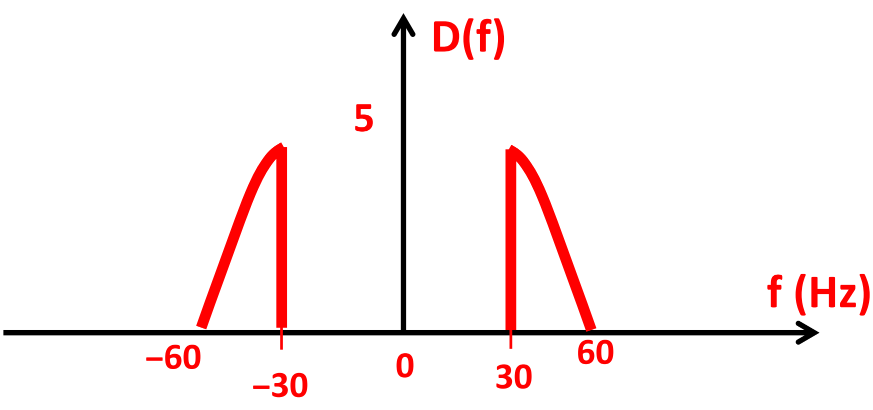
2. **Bandpass filter eliminates spectrum <100 Hz (gain of 2):**



3. **Second mixer shifts BOTH spectral portions by $\pm$70 Hz and reduces amplitude in half:**



4. **Lowpass filter retains lower frequencies:**

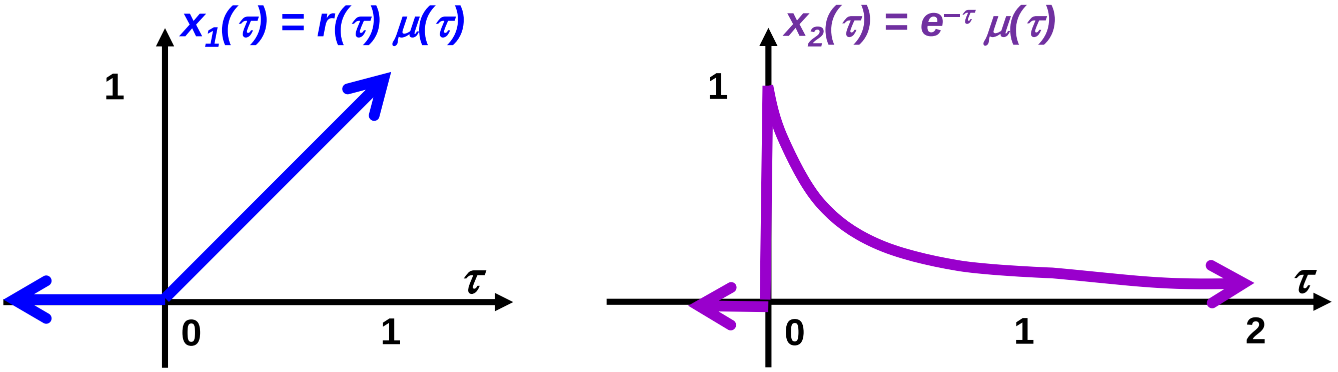


Convolution Example: Ramp ∗ Exponential

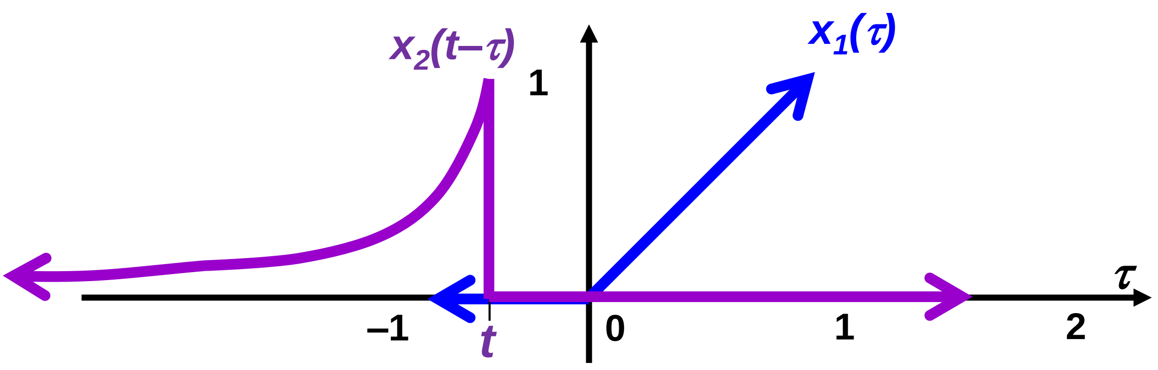
- **Use convolution integral to convolve $x_1(t) * x_2(t)$, where:**

$$x_1(t) = r(t)\,\mu(t), \qquad x_2(t) = e^{-t}\,\mu(t)$$

- **Solution:**



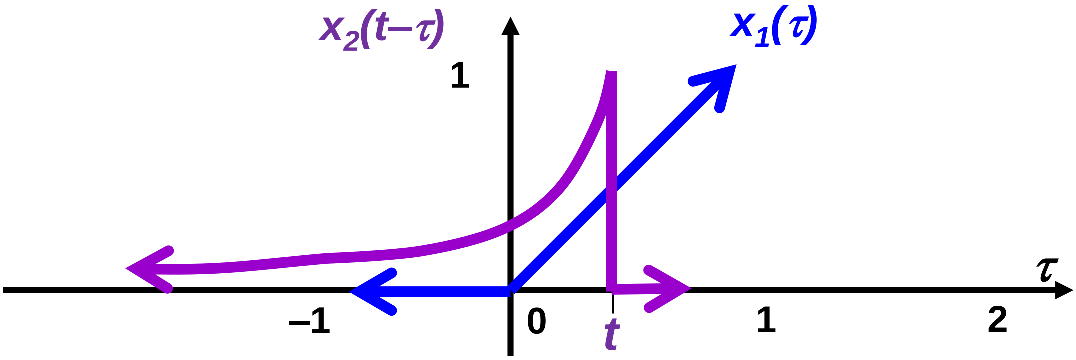
- **t < 0: No overlap. Convolution equals zero in this range.**



Continued

Convolution Example: Ramp ∗ Exponential (2)

- **t ≥ 0:** Partial overlap from left.



$$= \int_{\tau=0}^{t} \tau \cdot e^{-(t-\tau)}\, d\tau = e^{-t} \int_{\tau=0}^{t} \tau \cdot e^{\tau}\, d\tau = e^{-t}\left[e^{\tau}(\tau-1)\right]\Big|_{\tau=0}^{t}$$

$$= e^{-t}\left[e^{t}(t-1)\right] - e^{-t}\left[e^{0}(0-1)\right] = t - 1 + e^{-t}$$

- **Total Solution:**

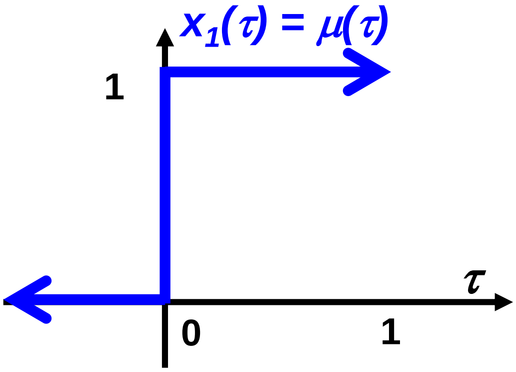
$$x_1(t) * x_2(t) = \begin{cases} 0, & t < 0 \\ t - 1 + e^{-t}, & t \geq 0 \end{cases} = \left(t - 1 + e^{-t}\right)\, \mu(t)$$

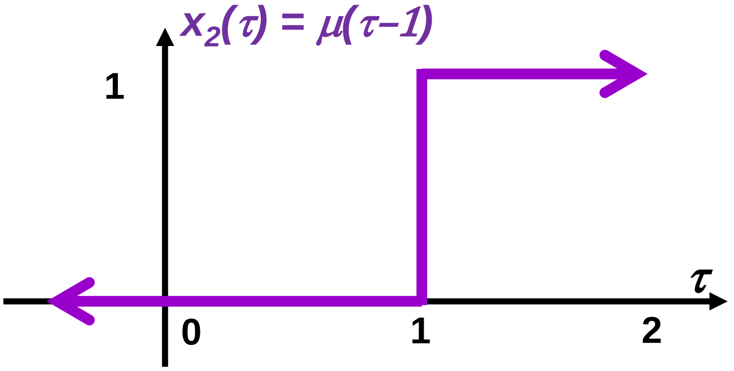
Convolution Example: Step ∗ Step

- Use convolution integral to convolve $x_1(t) * x_2(t)$, where:

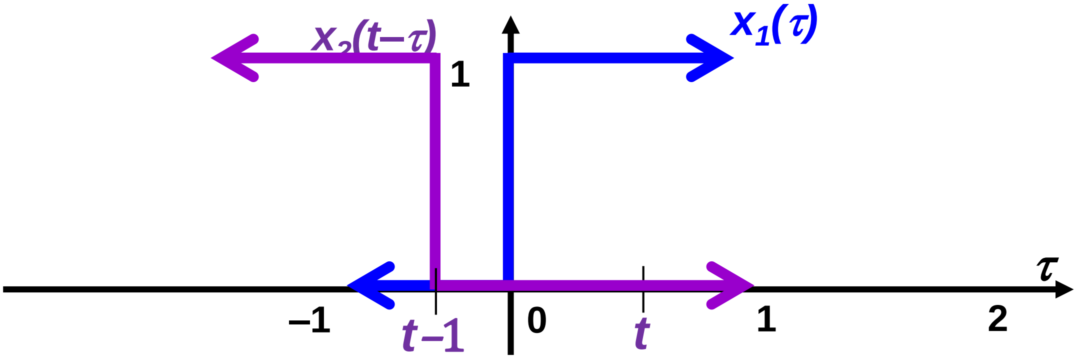
$$x_1(t) = \mu(t), \qquad x_2(t) = \mu(t-1)$$

- **Solution:**





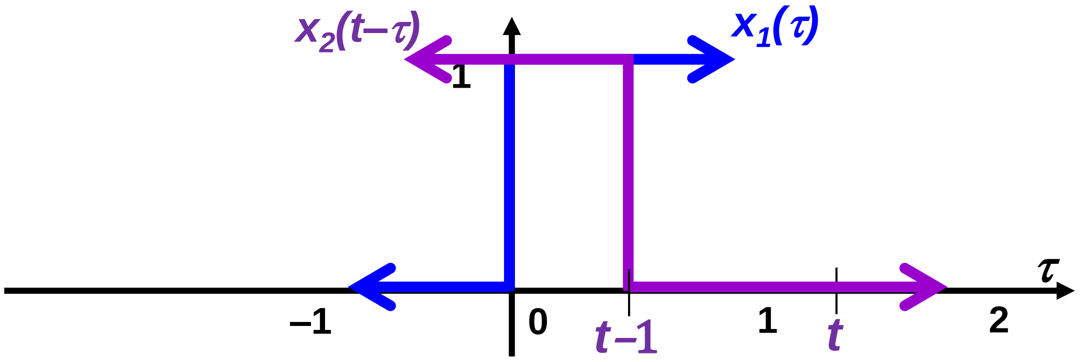
- **t < 1:** No overlap. Convolution equals zero in this range.



Continued

Convolution Example: Step ∗ Step (2)

- **t ≥ 1: Partial overlap from left.**



$$= \int_{\tau=0}^{t-1} 1 \cdot 1 \, d\tau = \tau \Big|_{\tau=0}^{t-1} = t - 1 - 0 = t - 1$$

- **Total Solution:**

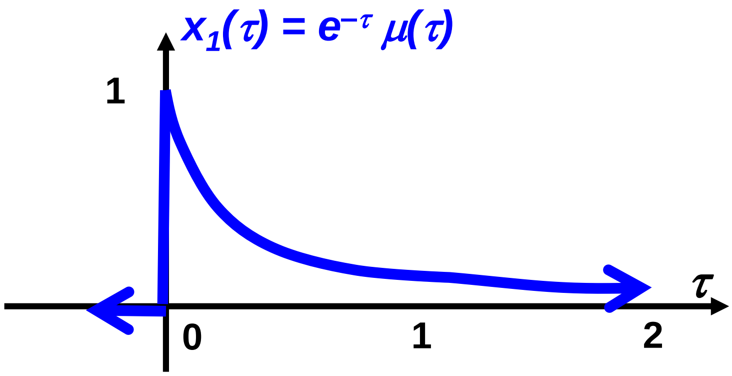
$$x_1(t) * x_2(t) = \begin{cases} 0, & t < 1 \\ t-1, & t \geq 1 \end{cases} = (t-1)\ \mu(t-1)$$

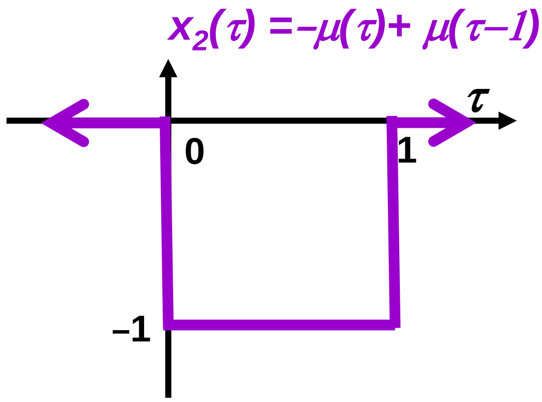
Convolution Example: Exponential ∗ Negative Gate

- **Use convolution integral to convolve $x_1(t) * x_2(t)$, where:**

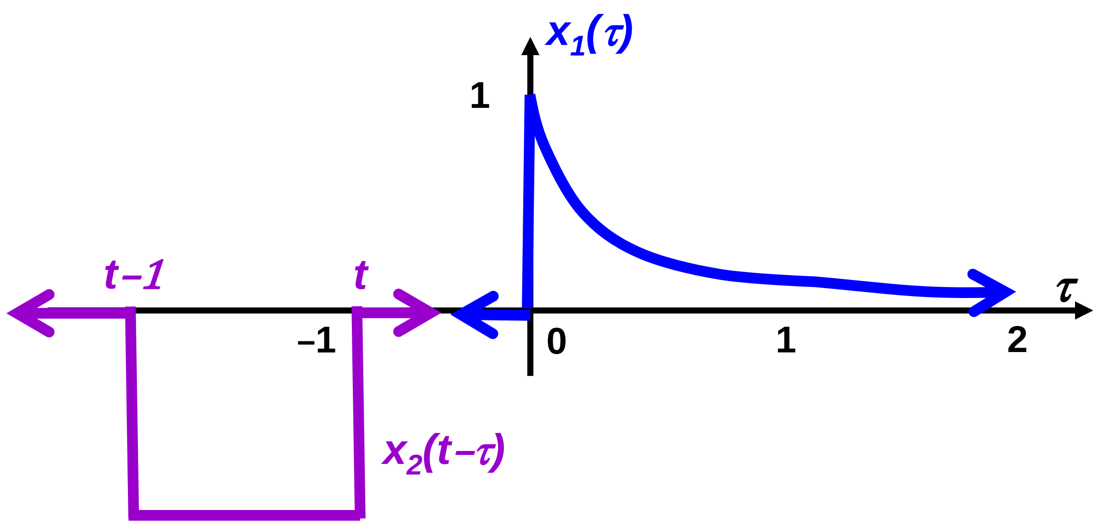
$$x_1(t) = e^{-t}\,\mu(t), \qquad x_2(t) = -\mu(t) + \mu(t-1)$$

- **Solution:**





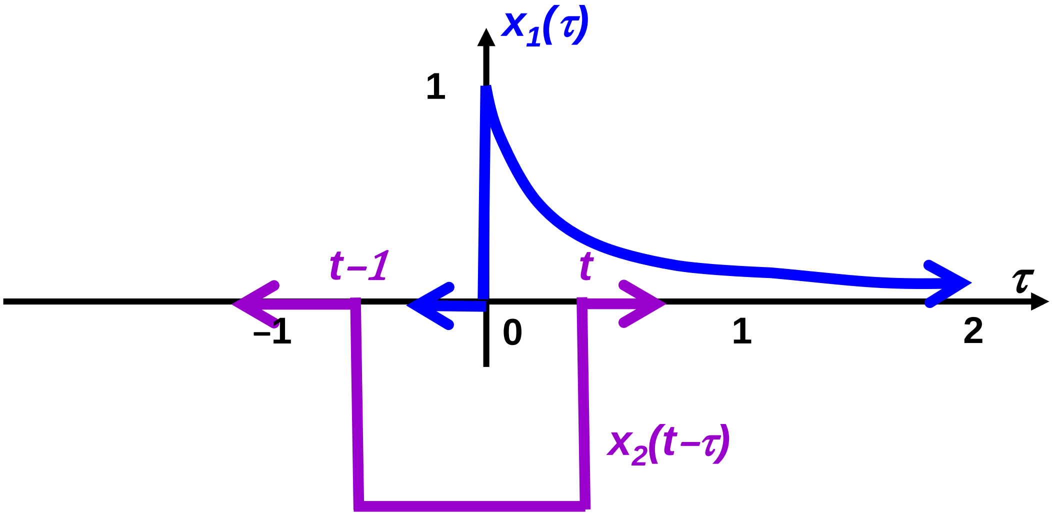
- **t < 0: No overlap. Convolution equals zero in this range.**



Continued

Convolution Example: Exponential ∗ Negative Gate (2)

• **0 ≤ t ≤ 1:** Partial overlap from left.

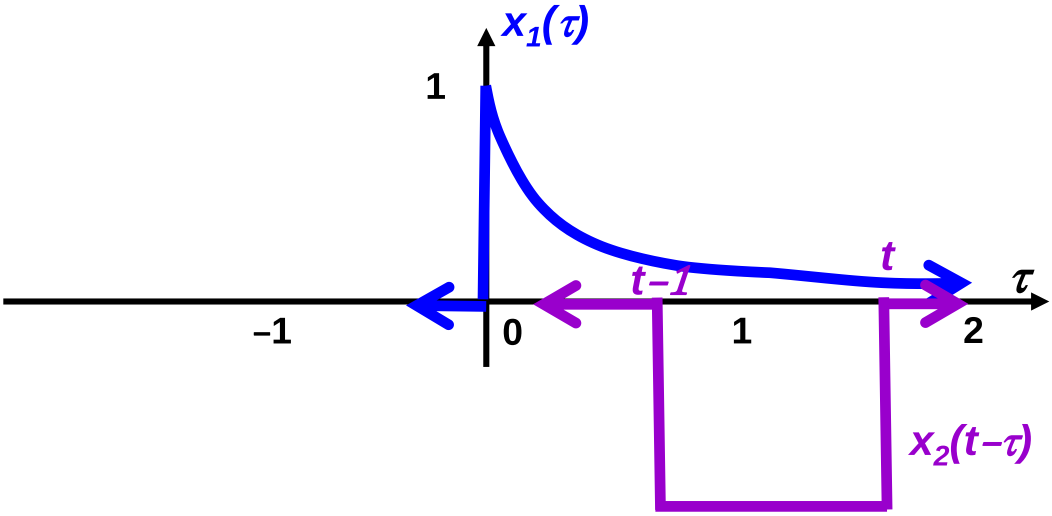


$$= \int_{\tau=0}^{t} e^{-\tau} \cdot (-1)\, d\tau = \left.\frac{-e^{-\tau}}{-1}\right|_{\tau=0}^{t} = e^{-t} - e^{-0} = e^{-t} - 1$$

Continued

Convolution Example: Exponential ∗ Negative Gate (3)

- **t > 1: Full overlap.**



$$= \int_{\tau=t-1}^{t} e^{-\tau} \cdot (-1)\, d\tau = \left.\frac{-e^{-\tau}}{-1}\right|_{\tau=t-1}^{t} = e^{-t} - e^{-(t-1)}$$

$$= e^{-t}(1-e)$$

- **Total Solution:**

$$x_1(t) * x_2(t) = \begin{cases} 0, & t < 0 \\ e^{-t} - 1\,, & 0 \le t \le 1 \\ e^{-t}(1-e), & t > 1 \end{cases}$$